the
UNIVERSITY
of
GREENWICH

An investigation into establishing a generalised approach for defining similarity metrics between 3D shapes for-Case-Based Reasoning (CBR)

Soran Saeed

A thesis submitted in partial fulfilment of the requirements of the University of Greenwich for Degree of Doctor of Philosophy

December 2006

University Of Greenwich
London

Abstract

This thesis investigates the feasibility of establishing a generalised approach for defining similarity metrics between 3D shapes for the casting design problem in Case-Based Reasoning (CBR).

This research investigates a new approach for improving the quality of casting design advice achieved from a CBR system using casting design knowledge associated with past cases. The new approach uses enhanced similarity metrics to those used in previous research in this area to achieve improvements in the advice given. The new similarity metrics proposed here are based on the decomposition of casting shape cases into a set of components. The research into metrics defines and uses the Component Type Similarity Metric (CTM) and Maximum Common Subgraph (MCS) metric between graph representations of the case shapes and are focused on the definition of partial similarity between the components of the same type that take into account the geometrical features and proportions of each single shape component. Additionally, the investigation extends the scope of the research to 3D shapes by defining and evaluating a new metric for the overall similarity between 3D shapes. Additionally, this research investigates a methodology for the integration of the CBR cycle and automation of the feature extraction from target and source case shapes.

The ShapeCBR system has been developed to demonstrate the feasibility of integrating the CBR approach for retrieving and reusing casting design advice. The ShapeCBR system automates the decomposition process, the classification process and the shape matching process and is used to evaluate the new similarity metrics proposed in this research and the extension of the approach to 3D shapes.

Evaluation of the new similarity metrics show that the efficiency of the system is enhanced using the new similarity metrics and that the new approach provides useful casting design information for 3D casting shapes. Additionally, ShapeCBR shows that it is possible to automate the decomposition and classification of components that allow a case shape to be represented in graph form and thus provide the basis for automating the overall CBR cycle.

The thesis concludes with new research questions that emerge from this research and an agenda for further work to be pursued in further research in the area.

Research Keyword

Case-Based Reasoning, Shape Recognition, Shape Decomposition, Shape Classification, Similarity Metrics, AutoCAD, Knowledge Management, Visual Reasoning.

Dedication

I dedicate this thesis to my two daughters Lana Saeed, Soma Saeed, my son Raheel Saeed, my wife Sakar, all Saeed family and all friends who supported me.

Acknowledgement

First and foremost, I thank my first supervisor, Dr. Miltos Petridis for his advice, friendship, and continuous support. Many of the ideas presented here and probably the better ones were actually his. I would like to thank Professor Brian Knight for his most useful advice and help and again many thank for Dr. Miltos for serving on my guidance committee. The many long "corridor conversations" we had throughout my years at Greenwich University, have helped shape my understanding of Case-Based Reasoning (CBR). Many of the underlying philosophical ideas expressed in this work, I have undoubtedly "inherited" from him. He has opened a window for me into the world of human vision. His theory of Recognition by components is an important foundation of this work. His enthusiasm towards his own work, and towards mine, was always a source of encouragement. They both gave invaluable suggestions and were always there for advice.

I thank the University of Greenwich for their help with all my administrative concerns throughout these years. Special thanks go to Academicals staff and Technical Support. I am also indebted to my fellow students in the research group, past and present. I will always cherish the great experience of working and socialising with such a diverse set of people from different backgrounds and with different cultures. I especially thank to Mr. Amin Mina with his family in UK for his full support. A great salute to Mr. Aziz M. and his team with over 30 years experience in metal casting design from city Slemani (KGR).

Content

List of Figures

List of Tables

List of Equations

Chapter 1

This chapter gives an overview and the objectives of the research for the readers. The research questions have been as far as possible answered in brief and it is explained how they were addressed. The knowledge contribution of this research follows, along with an overview of the following chapters.

1.0 Introduction

The research is concerned with establishing a generalised approach for casting metal designs to define similarity metrics between 3D shapes using graphical representations for shape matching in Case-Based Reasoning (CBR).

In order to evaluate the approach, an application system has been developed at the University of Greenwich, called "ShapeCBR system". This system has evolved through addressing additional research objectives such as the decomposition process, the classification, and matching for shapes together with two new metrics have been created the first one, called "Component Type Metric" (CTM) is improving the efficiency of similarity measurement for shape matching and the second one "Overall Similarity Metric" (OMS) is to calculate overall similarity metrics between complex 3D shapes. Finally an algorithm has been developed to evaluate the ShapeCBR system and CBR itself, to improve the efficiency and performance of the system. ShapeCBR has the potential for being integrated within CAD packages in the current market.

1.1 Rationale for current research

The research is mainly concerned with similarity metrics for the shape retrieval problem. It is also concerned with automating the processes of decomposition and classification for 3D geometrical shapes using a graphical representation to allow for the efficient retrieval of similar shapes and thus reuse of relevant casting design knowledge.

The background of the problem is metal casting designs. It is useful to define what "metal casting" means. A casting may be defined as a "metal object obtained by pouring molten metal into mould and allowing it to solidify." The liquid metal is poured into the mould cavity where it is shaped.

Of all the methods of processing components such as forging, machining, casting is the cheapest for mass production. The problem with casting is one of quality of the final product. This is very dependent on the know-how relating to design of the mould. There is now considerable body of knowledge which has been acquired from the work carried out by industries, government bodies and universities relating to casting products soundly within cost constraints. Although the value of design knowledge is widely recognised throughout the industry, the management of design knowledge is often unplanned in some respects. Design histories are often lost, or banished to paper files that are difficult to search. Also, design engineers retire [Pegler C.J.1993], or move away leaving inadequate design records. There are many problems faced by a casting design engineer, centring on the physical freezing processes. Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness [Campbell, 1991]. Other practical problems arise during pattern making and subsequent machining of the cast part.

Jolly [Jolly, M. 1996] found in his survey that the foundry industry is looking for software applications that can not only predict problems that occur during metal solidification (such as shrinkage porosity) but also, having predicted these problems, propose intelligent solutions for the problems found. Current commercial casting software can be classified into two broad areas:

1. Intelligent knowledge-based systems (IKBS), [Hennessy, D. Hinkle, D 1992],
2. and numerical simulations based on physical process models [Corbett, C.F. 1989].

The advantages of a CBR system are that it is to possible store the valuable know-how and to distribute the expertise.

Intelligent knowledge-based systems (IKBS) attempt to support an earlier stage in the design process. Numerous software tools such as those discussed in [Knight, B; et al, 1995] have clearly demonstrated the effectiveness of knowledge-based and other advanced (new) heuristic-based programs for designing castings.

There are some commercial software packages available in the market that can calculate the position of feeders, e.g. NOVACAST [NovaCast: Sillen, R.1991]), analysis geometric properties and gives suggestions further improvement the design e.f. AutoCast [Ravi, B.1999].

Although many prototype tools have demonstrated the efficiently of CBR in the domain of engineering and design [Marir, F; Watson 1994 -15], there is an insufficiency of research for its use in the foundry industry. CBR can play an important role in intelligent casting software. One commercial CBR system [Price, C.J. et al. 1997] called Wayland is used for the setting of parameters in pressure die-casting. This research has demonstrated that CBR has an exciting future in casting software.

The main problem for a CBR system is how to retrieve cases efficiently, where the retrieval based on the shape. Although there are other possible search indices, for example the type of casting alloy, weight and general description of part (wheel, sea-gland, valve, engine bearing cap. etc.), these descriptions are too general for accurate retrieval. General classifications of shape components have been proposed; for example, Biederman's *geons* [Biederman et al 1992]. However, during this research, it became apparent during knowledge elicitation, that a decomposition of shapes specific to the casting industry already existed in practice [Knight B. et al, 1995, and Wlodawer, R: 1967].

1.2 Research questions

1.2.1 The main question is:

[Q] Is it possible to retrieve useful casting information efficiently and automatically from a "similar" existing three-dimensional casting design for a given target shape? (The similarity problem in casting design).

To fully understand the main question of similarity metrics between three dimensional shapes for shape retrieval problems it is necessary first to answer the following additional questions:

1.2.2 The Componentisation questions

This section presents the primary objective along with the additional questions of this research, attempts to prove the feasibility of shape componentisations automatically into connected generic components, which would help a casting designer to store the products of decomposing and classification into case base knowledge. Regarding automatic retrieval driven by a given target design, the first componentization question is:

[CQ] Can the CBR process for the shape retrieval of casting designs be automated?

CBR process: is retrieve and re-uses experience for problem-solving tasks. CBR process is effectively applies past solutions to new situations. From a case base, which stores and organises past situations, the CBR process chooses situations similar to the problem at hand and adapts their solutions.

This question raises three sub questions:

[CQ-a] Can 3D shapes be automatically decomposed into a set of substantially different 2D shapes (views) that can be used to retrieve useful casting knowledge?

[CQ-b] Can 2D shapes be automatically decomposed to a set of connected generic components?

[CQ-c] Can useful casting knowledge about casting shapes be retrieved automatically from a CBR system that stores the componentised views of the shapes?

It is shown in this thesis that it is possible to decompose a shape automatically to a set of connected generic components.

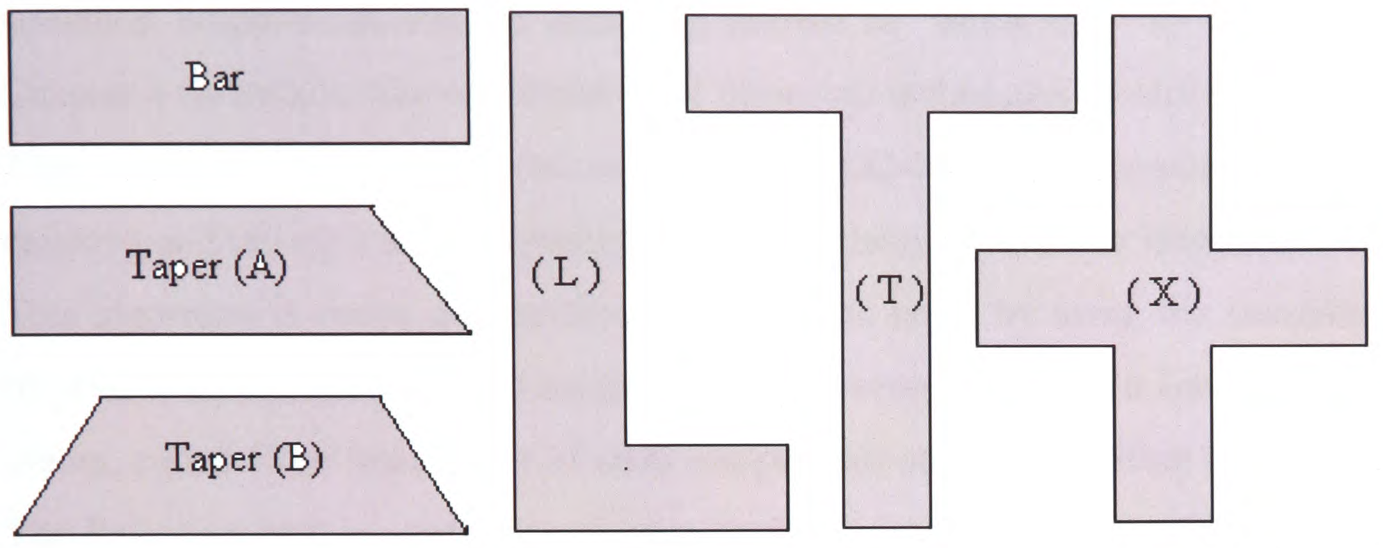


Fig.1.0 shows the proposed primitive components (L, T and X), Bar and Tapers elements [Mileman: 2000].

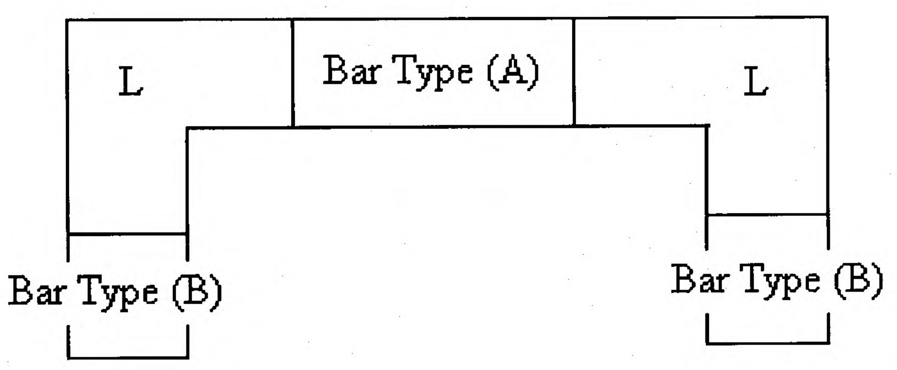


Fig. 1.1 shows bars type (A) and type (B).

This thesis attempts to answer this question by devising and testing a novel algorithm for decomposing 3D shapes into a set of substantially different 2D shapes or (views).

This algorithm is based on the identification of hotspots. A hotspot is an important point for the decomposing process which is made up from two connected original lines, (the original lines concerning the boundary of the shape). *This point is only concerned with internal geometrical information for the shape and only from this point can penetrate (go through) the shape (A shape could be made up from one or more components and these components can be classified into different component types and these types could be identified by their* *internal geometrical information for the shape).* Once hotspots have been found, projecting techniques are used for decomposition of the shape into a set of rectangles and triangles can be used to define the taper types.

It is shown in this thesis that it is possible to classify the generic components automatically and efficiently into identifiable elements, components and regions. Components are classified as Ls, Ts, and Xs (Fig.1.0) etc. by using an algorithm known as "Full-scan" to identify the structural components and an algorithm known as "Semi-scan" to identify elements. See Chapter 4 for details. The combination of these two techniques identifies the regions.
This thesis attempts to answer the sub question **[CQ-b]** of the componentisation question by devising and testing a novel algorithm for shape classification into identifiable components.
This algorithm is based on identifying component types by using the *searching method* for the *first hotspot*. Once the first hotspot has been recognised, then *a rectangular shape* will be drawn, called Core-Spot (heart of each component) see the algorithm in Chapter 4 in details. The Full-scan task is continuing to scan cyclic overall the shape, point by point, to find the hotspots and identify components by their number of hotspots. This method is called "Full-scan".. (See on Fig.1.0).

This thesis attempts to answer the sub question **[CQ-c]** of the componentisation question by devising and testing a novel algorithm for shape matching.
Once sub-question **[CQ-a]** and **[CQ-b]** are resolved and stored in case-based knowledge, then it is possible to retrieve useful products from 3D shapes automatically and efficiently for a given target shape. Section 1.3.3 introduces the second main question of similarity metrics.

1.2.3 The Similarity metrics questions

This section is focusing on similarity metrics between 3D shapes. An algorithm has been designed that could produce a competent and efficient way to retrieve useful casting design, automatically, from case base knowledge, within the ShapeCBR System.

The similarity question poses different sub-questions which are as follows:

The first sub question is:

[SQ-a] Is it possible for the similarity metrics devised in previous research to be improved to produce more efficient retrieval of useful casting advice from the ShapeCBR System?

The problem concerned:

Mileman [Mileman: 2000] proved that useful casting advice from the ShapeCBR system could be retrieved from similar designs. However, Mileman assumed that components of the same type are identical. For example any types of bar whether thick or thin, are deemed similar, so basically the size of the component types have not been considered.[Mileman: 2000]In his approach one of the similarity metrics were *component numbers with their types*, which was an inefficient and cumbersome process that could hamper the practical use of a commercial system. Additionally, only the type and not the actual geometrical dimensions of a component were stored. This prevented us from increasing the sensitivity of the similarity criteria to take account of a similarity measure between components of the same type. For example, it makes sense that the aspect ratio of a Bar component would affect its similarity to another bar component for purposes of casting. The positioning of feeders and chills can be affected, so that the knowledge associated with a shape may be contingent not only on the types, but on actual geometrical features of the constituent components.
The leaf metric is defined by the nodes of a graph which are connecting components (for example: bar, taper) and that have one 'free' interface connection.

Mileman, assumed that bar and taper component types count as leaves But this research investigated the assumptions that they are different elements. For example a bar is rectangular-shaped and cannot be divided into more elements. But the taper component type can be divided into two or more elements. Fig.1.0 shows that the taper type (A) made up from one rectangular-shaped and one triangle-shaped so it means there are two elements and for taper type (B) made up from two triangles and one rectangular-shaped so it means there are three elements). Therefore we believe that the differences between these two component types play an important role and they are affecting the degree of competent similarity measurement for shape retrieving.

However the only new additional metrics that have been applied for current research is combining numbers of component types with types of component plus their size. And we believe that this new metric is useful for the efficient shape retrieval process.

The research deals with 3D ***axisymmetric*** shapes (shape has one view cross section); and also 3D ***arbitrary*** shapes (shape has a finite number of different cross-section views) within the current thesis.

The second sub-question is:

[SQ-b] Can a competent similarity metric between 3D casting shapes be defined to allow for retrieval of useful methoding advice associated with 3D shapes?

This Similarity question deals with 3D arbitrary shapes.

The above question was considered by Mileman in his thesis [Mileman: 2000], but was left as future work.

Typically, casting shapes are stored as files produced by CAD packages such as AutoCAD. These files contain all geometrical information and most CAD packages provide facilities for providing 2D sections through the 3D shape. The case base in the first system contained only one 2D cross-section (3D **axisymmetric** shapes) through each shape assumed by Mileman. However, in many cases two or more substantially dissimilar 2D sections could provide a more accurate description of a 3D shape. These would need to be taken into account for a more efficient retrieval of 3D shapes. The selection of dissimilar 2D sections can be achieved with the use of a similarity threshold to define substantially dissimilar sections.

Arbitrary 3D shapes can be treated as two or more cross-sections or views and these can provide valuable identifiers to enable accurate retrieval. In this case, the overall measure of similarity between two 3D shapes needs to be considered. For example take multiple views of a target shape and compare with multiple views of retrieval shapes.

1.3 Research Methodology

In order to answer the research questions defined in this section, a conclusive literature review was conducted. The literature review investigated the body of knowledge covering the application area in casting design and looked into previous attempts to automate decision support using empirical, numerical and knowledge based/AI techniques. Additionally, there was a comprehensive review of CBR techniques and applications to similar application areas. Based mainly on previous successful work in the area [Mileman :2000], it was decided to extend the CBR techniques used there with a view to automating the CBR process, improving the efficiency of knowledge retrieval and extending to 3D shapes.

The new approach and techniques were implemented into a system called ShapeCBR and a qualitative evaluation was first conducted. Following this, further evaluation was conducted using the case base used in Mileman research as a benchmark to evaluate the efficiency of the new approach. A further evaluation of the applicability of the approach to 3-D shapes used elicited from a casting domain expert and advice retrieved was compared to the human expert advice. The results of the evaluation showed that the CBR technique can be automated for 2D shapes and that the proposed enhanced similarity metrics brings about efficiency gains in terms of the quality of the advice gained and that the approach can be extended to 3D casting shapes.

1.4 Achievements

This section describes briefly the main achievements of this research while investigating the questions posed.

This research contains a number of contributions both in the specific field of similarity metric between 3D shapes for CBR, and to CBR itself in general. These are:

A new algorithm has been designed and tested to automate shape processing in a competent and efficient way for decomposing shapes into a set of connected generic components and to classify decomposed products into generic components of identifiable types (Fig.1.0) [Mileman, Thesis: 2000] such:

- Bar
- L-component
- T -component
- X-component
- Taper

The similarity metrics between components on the same type have been extended using methods that take into account the geometrical features of each single shape component. The improved similarity metrics have been shown to give better results by matching and retrieving better expert casting advice. (See chapter 5 on similarity metrics).

Finally, an efficient equation has been created for overall similarity metrics for 3D rotational symmetric shapes using graphical representations to matching the shapes. Overall similarity metrics between arbitrary 3D shapes can be defined and used to retrieve relevant casting advice. The 3D shapes can be treated from one view to a number of cross-sections or views. Often these shapes can provide valuable identifiers to enable accurate retrieval. Chapter 6 on evaluation discusses this in detail.

1.5 Thesis Summary

The thesis is divided into seven chapters, which are dealing with particular processes dependent on each other.

Briefly, the scope of each chapter can be described as follows:

The first chapter covers the Introduction of the thesis and provide a background to the problem casting design, by investigating the main research questions, its sub-questions, and possible ways of finding solutions, to overcoming the problem. It also describes the methodology of the current research and followed by achievements and thesis summary.

The second chapter presents the relevant literature review, as well as extracts of the undertaken research. The research was quite widespread, since various issues had to be investigated and the nature of the casting designs (shapes) and their method of engineering had to be established. The information technology that was used, more specifically CAD application [AutoCAD] and their suggestions were examined. The availability of artificial intelligence (AI) approaches to CAD, case-based reasoning (CBR) techniques, visual design, and knowledge management were studied.

The third chapter covers the decomposition algorithm to automate the decomposition process of shapes. This is analysed in depth along with the 3D models approach. The decomposition algorithm deals with numbers of 2D cross sections or (views).and these views represents 3D shapes. The outcome stored into the case-base knowledge is ready for next step of 'classification'.

Chapter four covers the design of a second algorithm to automate shape classification, which is analysed in depth by recalling decomposition that results from the case-based database. In this chapter, new techniques have been analysed and designed by developing two algorithms to automate shape classification. The first method is "Full-scan" to identify the components and regions (L, T, X and tapers) and the second method is "Semi-scan" to identify elements (type of Bar (A) and Bar (B)) see on Fig.1.1. The results of this classification are stored into the case-based knowledge, ready for retrieval.

Chapter five deals with the similarity approach. A (CBR) technique has been used for similarity metric and has been analysed in detail. A number of algorithms have been created to implement an application for shape matching. Several equations and formulas have been created to achieve both individual similarity and population similarity (in this case looking for overall similarity between two 3D shapes) and also to describe the actual problem with their final solutions for the shape retrieval process.

Chapter six introduces an evaluation for current research based on experimental results, by testing over 100 3D shapes and 20 additional new shapes from 3D arbitrary shape types. In this chapter evaluation is based firstly on previous research results from Mileman [Mileman: 2000] and secondly took previous data and replacing with new record from current research and evaluate by the system ShapeCBR to see the progresses and successful of through current methods have been used and the third test was on overall similarity metrics between arbitrary 3D shapes. These shapes can be treated from one view to the number of cross-sections (views). Often these shapes can provide valuable identifiers to enable accurate shape retrieval. And the final results for the three testes have been judged a human expert in related areas.

Chapter 7 presents the conclusion and further work of the thesis. It starts with a brief summary of the main achievements, and then discusses future enhancements, present a numbers of contributions, followed by a number of Appendices.

Chapter 2

Literature Review

An overview and objectives of this current research have been introduced in chapter one and a brief explanation of each question has been presented. The aim of this chapter is to establish the domain of this project, namely casting design, CAD and CBR. The various well-established techniques, knowledge management, computer Aid design (CAD), knowledge based design, Case-Based Reasoning, and more are examined in an attempt to discover if improvements can be made to the early phases of casting designs.

In addressing expertise in the comprehension of casting design and CAD, this study draws upon literature from many sources including cognitive science, psychology, and architecture. Although it is beyond the scope of this thesis to present an extensive review of this literature (Gobert, 1998 for a thorough review), some literature will be presented to provide the readers with a suitable background for the research that took place. It should also be noted here that it is beyond the purposes of this study to provide a detailed discussion of the numerous different design theories.

2.0 Introduction

Experiential knowledge plays a significant role in the human reasoning process as previous experiences help in understanding new situations and in finding solutions to new problems. Experience is used when performing different tasks, both those of a routine character and those that require special skills. This is also the case when designing where over 50% of the work on a day-to-day basis is routine design that consists of modifying past solutions [Moore: 1993]. This means that most of the design problems have been solved before, in many cases over and over again. Despite this, the computer support used by designers still lacks the ability to use experiential knowledge in a rational way. In recent years, researchers in Artificial Intelligence (AI) have studied if cases (knowledge about specific problem-solving experiences) could be used as a new case of experiential knowledge. Cases are valid in a specific situation in contrast to generalised knowledge, e.g. base rules.

Making use of past experience in the form of cases is commonly known as Case-Based Reasoning (CBR) [Kolodner: 1993]. The application of CBR in design, known as Case-Based Reasoning Design, is still in its infancy even though several CBR systems focusing on various domains have been developed [Maher, et al 1997, Rivard and Fenves, 2000]. Although many of these applications are useful in solving the specific problem that they are aimed at CBR systems are seldom used in practice the reasons is that the information of the case used is system-specific to creating such representations provides the system developer with an opportunity to investigate new ways to represent design information and much knowledge has in this way been gained on the other hand, this limits the information available for the CBR to information either created by the CBR system or information translated to the system-specific representation. Because these representations are rather complicated and different from those used by the ordinary designer when documenting design information, it is difficult to achieve an automatic translation. For this reason, most CBR systems only contain cases that are produced using the respective system or information translated by hand to the system-specific representation.

Case-Based Reasoning (CBR) originates from the cognitive observation that humans often rely on past experience to solve new problems. Using this observation, [Schank, 1982] created the theory of dynamic memory, which describes a concept of memory organization that could be used as a guideline for computer representation [Schank, 1999]. The premise of dynamic memory is that remembering; understanding, experiencing and learning cannot be separated from each other [Kolodner, 1993]. We understand by remembering old similar situations and use these to create expectations about the new situation. If these expectations turn out to be right, we feel that we understand; if the expectations fail we try to explain why by remembering old situations with similar failures. These explanations are then used to change the memory (Learning) so that the new situation can be understood.
In order to make this possible, the same knowledge has to be used for remembering, understanding, experiencing, and learning.

The analogy between dynamic memory and a system facilitating CBR is rather near at hand. The main aim for CBR in such a system is to find, i.e. recall, old experience that can be helpful in the present design situation. This experience is used when designing for understanding the problem and for finding a solution. The design activity creates another experience that can be stored in the design system for the purpose of reuse. As stated in the theory of dynamic memory, this can only be possible if the CBR activity and the design activity share common properties and structures.

The theory of dynamic memory also implies that understanding is the main aim and remembering supports this activity. Using analogy again, it can be stated that designing is the main aim for a design system while CBR aids this activity. Concerning the choice of presentation, this ought to yield that the representation and the information used for designing should also be used for CBR. It should also be pointed out that unless the CBR process becomes more or less automatic, the designers would be reluctant to add potentially useful cases to the case-base [Flemming and Woodbury 1995] or to try to reuse old cases. The only way to avoid extra work for the sake of CBR is by enabling the CBR system to use the information created by the designer during the design process. This research proposes an approach for capturing shape decomposition (Chapter 3), and classification process (Chapter 4). Chapter 5 briefly presents how this information can be retrieved. Having this approach, Chapter 6 describes CBR-Shape System, a prototype implemented to test the proposed approach and Chapter seven concludes the thesis.

This chapter deals with current research, reviewing the case-based reasoning (CBR) literature as the main method to tackle the problem, based on the reuse of past cases and the use of a computer-aided design (CAD) tool to design the components.

CBR is a part of Artificial Intelligence (AI) which was discovered in the mid 90's, It can be used to review CAD documentation to produce plans and all types of engineering drawings (which can mean producing all documents with the computer) In addition to drawings, different bills of quantities are directly attached to all types of engineering drawings (architects, mechanic, electric, electronic works etc), and Visualization (Visual reasoning, e.g. thinking in shapes, forms and images) is a fundamental attribute of casting design, and therefore combining it with CBR may provide significant results both for the field of design thinking as well as for the field of Computer-aided Design (CAD). All these three elements combining together will have a high level benefit for the research and the knowledge above is of immediate interest in answering the research questions of this project. Also the information in this section can be of great importance for the usability of the produced software, and the options that need to be implemented.

2.1 Overview of the research problem

The research focuses on the problem in metal casting designs. Jolly [Jolly, M. 1996] found in his survey that the foundry industry is looking for software that can not only predict problems that occur during metal solidification (such as shrinkage porosity) but also, having predicted these problems, propose intelligent solutions for the problems found.

2.2 Background of the problem

There are many problems faced by a casting design engineer, centring on the physical freezing processes. Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness [Campbell, 1991]. Other practical problems arise during pattern making and subsequent machining of the cast part. Many software tools have been developed to assist the designer to solve this problem.

2.2.1 Metal casting designs problem

Solid Shrinkage; is one of the problems in metal casting designs (often called patternmaker's shrink) occurs after the metal has completely solidified and is cooling to ambient temperature. Solid shrinkage changes the dimension of the casting from those in the mould to those dictated by the rate of solid shrinkage for the shapes see on Fig.2.0 (Aziz: 2004).
Pouring; is another problem in metal castings are produced in moulds that must withstand the extremely high temperature of liquid metals. Interestingly, there really are not many choices of refractors to do the job. As a result, high molten metal temperatures are very important to casting geometry as well as what casting process should be used (Online).

The problem with casting is one of quality, which depends on the existence of casting design knowledge. The advantages of a CBR system have been introduced in this research as a help to solve casting designs problem cases in a CBR case-base contain detailed information on the design process for products. This is an advantage allowing CBR systems to realise casting know-how as a valuable asset [Mileman: 2000].

Fig.2.0 shows the rate of solid shrinkage problem for the shape.

2.3 Survey of Computer Aided Methods to Assist Casting Design.

Expert systems is one of Artificial Intelligent (AI) techniques, expert system is a system in which knowledge is represented as it is, possibly in the same form that it was extracted from an expert. In an expert system the represented knowledge should endeavour to solve problems in the same way as the expert knowledge source solved them. Computational Flow Dynamic (CFD) [Cleary et al.: 2003], is another tool using numerical methods representing the fundamental physical processes. Many researchers used this method for casting design problem and many applications have been developed for this problem. This method needs an expert with a high mathematical background to run and use the program. As such, this approach comes with a cost and great difficulty for the user, and the recent one, case-based reasoning (CBR) discovered early 90's and was raised by researches for the first time to solve this type of problem [Kolodner et al: 1994].

CBR is the cheap and easy way to run. More than half of the daily work done by designers is routine design that consists of modifying past work [Moore 1993]. It should be, therefore, of great use to create a case base in order to reuse old cases in similar future projects. Nevertheless, the methodology of case-based reasoning (CBR) in design is rarely used, probably due to the problems with structuring the database and finding easy ways for saving and reusing the information, i.e. the issue of standards for information exchange.

Case-based reasoning: (CBR) has been pointed out as a promising aid to help this situation. In order to be of practical use, however, a case-based reasoning design system has to be able to use the information that the designer creates during the design process [Kolodner et al: 1994].

CBR can be used both when a domain is well and not so well understood. In the latter case it assumes the role of a generalised model. Provides for efficient solution generation and evaluation is based on the best cases available. Needs a means of evaluating its solutions, guiding its adaptation and knowing when two cases are similar. Next section is discussing on case-based reasoning in details.

2.4 Case-Based Reasoning

This research used case-based reasoning to aid casting design. (CBR) provides case stories to help designers solve design problems. It provides designers with stories about previous successes and failures during the early phases of problem solving. It comprises case studies, shapes, images, text documents and photographs, and also includes a library containing design principles, previously encountered problems and resolutions to help designers anticipate and avoid conflicts among the service systems. Next section is on using case-based reasoning system.

2.4.1 Using Case-Based Reasoning

In this chapter review, we propose a method for the computer to presume interactively the design support method, in order to provide useful information for design based on the framework of CBR. CBR is decision-making, learning, and problem solving. Case-based reasoning methods generally have the following aims: to avoid preparing a priori fixed, detailed rules and knowledge sources: to provide flexible and various information through the modification; to add Case-based Reasoning Support Method Recognition and to extend knowledge sources step by step.

2.4.2 Case-Based Support in Casting Design

This section presents a number of CBR tools related to this area such as:

- CYCLOPS [Navinchandra 1991] which supports landscape layout.
- JANUS ([Fischer and Nakakoji 1991] is supporting kitchen design.
- FABEL [Consortium 1993] is supporting construction component.
- SEED [Flemming and Woodbury, 1995] is a system environment which aims at providing computational support for the early phases in building design. The next section introduces some models of CBR.

2.4.3 Models of CBR

Oxman [Oxman: 1994] identifies four cognitive approaches for modelling design case knowledge:

Generic models: Knowledge is used to define classes of designs called generic designs. It is often convenient to make the generic nature of knowledge explicit. Rather than using grammatical rules, a design space may be defined in terms of a class description called a generic model.

Associative models: The associative mechanism is another key principle of cognition, which is present in design thinking. In associative reasoning concepts are linked on the basis of conceptual relations to form a structure of concepts. Historical styles such as Doric and Gothic also provide associative models. Doric gives democratic and Gothic religious, associations.

Exemplar models: In this approach it is attempted to re-use past knowledge rather than to generate new designs. The previous solution is adapted to the current situation. Previous knowledge is associated with specific design cases in which the knowledge is highly explicit. Casting designs are example-based and detailing is often based on the re-use of specific examples, which are exemplars, or examples that equate to models in the knowledge domain. Three broad classes of domain knowledge can be identified:

1. Procedural knowledge is a process or algorithm for design. The design of a staircase is an example where the calculations are based on floor to floor height, length of the stair run, and the tread riser relationships.
2. Causal knowledge is a detailed procedure for calculation. An example is the calculation and design of partitions for thermal or acoustic properties.
3. Behavioural knowledge is the understanding of the performance achieved by particular materials or by a particular configuration of elements in a building. This characterises much of the knowledge of building detailing.

The design precedent: the selection process of relevant ideas from prior designs in current-design situations has been termed precedent-based design. During the course of exploration of design ideas within precedents, designers are able to browse freely and associatively between multiple precedents in order to make relevant connections. This makes the discovery of unanticipated concepts possible in precedents. In precedent-based systems the ability to encode, search and extract design knowledge relevant to the problem at hand is significant.

For the problem domain of this research, design knowledge based on the domain of practical knowledge (exemplar model) seems to be the required solution. The next section presents a brief definition of CBR techniques and how they work.

2.4.4 Case-Based Reasoning techniques

Case-based reasoning (CBR) systems expertise is embodied in a case-based knowledge of past cases, rather than being encoded in classical rules. Each case typically contains a description of the problem, plus a solution and/or the outcome. The knowledge and reasoning process used by an expert to solve the problem is not recorded, but is implicit in the solution. To solve a current problem CBR techniques have been suggested and the lines below describe CBR methods.

All case-based reasoning methods have in common the following techniques:

To retrieve the most similar case (or cases), they compare the case to the case-based knowledge which they have stored in the past, by reusing the retrieved case to try to solve the current problem. Then they revise and adapt the proposed solution if necessary and what they retain will be the final solution as part of a new case.

There are a variety of different methods for organising, retrieving, utilising and indexing the knowledge retained in past cases.

Retrieving a case starts with a (possibly partial) problem description and ends when a best matching case has been found. The subtasks involve:

In identifying a set of relevant problem descriptors, matching the case and returning a set of sufficiently similar cases (given a similarity threshold of some kind); and selecting the best case from the set of cases returned.

Some systems retrieve cases based largely on superficial syntactic similarities among problem descriptors, while advanced systems use semantic similarities.

Reusing the retrieved case solution in the context of the new case focuses on: identifying the differences between the retrieved and the current case; and identifying the part of a retrieved case which can be transferred to the new case. Generally the solution of the retrieved case is transferred to the new case directly as its solution case.

Revising the case solution generated by the reuse process is necessary when the solution proves incorrect. This provides an opportunity to learn from failure.

Retaining the case is the process of incorporating whatever is useful from the new case into the case base. This involves deciding what information to retain and in what form to retain it; how to index the case for future retrieval; and integrating the new case into the case-based knowledge. The concept of the CBR system used in this research more details on chapter 6.

2.4.5 Related Applications for casting design in general

Several software tools may be used to assist the methoding process. For the initial stages of methoding these tools need to be fast and easy to use: simple models based on the cooling modulus principle, or fast empirical mould-filling models. Amongst these are: SOLSTAR [SOL], which support the initial design stages, and slower, more detailed numerical models such as SIMULOR [SIM], which support the simulation stages. CRUSADER give numerical support on such aspects as feeder sizes and feeder-feeder distances, but do not attempt to give experiential advice on such elements as re-design for casting, or mould orientation. More advanced numerical software (SPH), using computational fluid dynamics techniques [Cleary et al.: 2003], which support the simulation stages of die filling predictions is very high and the last locations to fill correlate well with porosity void age observations made by manufactures of these components.

All people use CBR in one way or the other, in much of their on a daily basis reasoning. It's the natural way people solve any kind of problem in their life by remembering solved problem and reused when it needs. CBR is easy to understand, does not require a lot of knowledge and is easy to use.

2.5 An Overview for Computer Aided Design (CAD)

The first articles concerning computer-aided design were published in 1961 and 1962. They referred to programs intended to produce plans and all type of engineering drawings. The intention was to describe a part of any tools as one graphical object. Despite the development from 1965 to 1975 programs were very difficult and awkward to use. That created a kind of "bad reputation" that followed CAD for years to come. CAD is designing where traditional tools are replaced with one system. CAD is a wide concept containing almost all features of information technology in design [Kiviniemi and Penttilä 1995]. Without effective utilization, investments are useless and working shrinks to computer aided drafting. Insufficient capabilities shift attention from design to equipment and programs and the work itself suffers [Heikkonen 1995]. Also the wrong basis for CAD investments has led to poor results and caused a negative attitude towards information technology on a wider scale [Naaranoja 1997]. CAD can mean producing all documents with the computer. In addition to drawings, different bills of quantities are directly attached into all types of engineering drawings (architectural, mechanical, electrical, electronic works etc). In reasonable CAD these bills of quantities can be produced straight from the database. Building specifications and other text documents are, however, produced with separate computer applications, at least so far [Kiviniemi & Penttilä 1995]. The ideal situation from the design point of view would be the possibility to process in three-dimensional models, which almost exactly match the forthcoming shapes. Managing the model, especially geometrical its information is difficult and the size of the file will easily become too big to handle. In present applications there are two main solutions to treat the three dimensional information, namely, *vector graphics objects and oriented objects.*

The central concept in the object approach is that of the object. An object associates data and processes in a single entity, leaving only the interface visible from the outside. The object approach is characterised by the structuring of problems into object classes. But the domains where this approach is used require complex software. Lately it seems more and more applications are using vector-based graphics instead of objects-based, although in many cases there is a combination of solutions. Applications with vector graphics (e.g. AutoCAD) are based on graphical elements, vectors and lines and they generally use drawing programs. Most of the general application programs, like AutoCAD, have sub-applications, which utilise the environment in the main application (e.g. PomARK and ARKsystems, are special applications developed in architectural design and use the framework of AutoCAD). Almost all present applications are based on working on 2D levels, but in some programs the three-dimensional model evolves in the background and can be seen from another window. 3D-based models can be divided into wire-frame, solid, surface, space and rendered models. The majority of 3D applications can produce wire-frame or surface models, but no applications can make the space model. The wire frame consists of lines in the edges of the object, and the surface model is the surfaces of the object represented with visible lines. The space model describes the real object [Davies *et al.* 1991, Holvio 1993, and Medland 1988]. Rendering means producing coloured and shaded pictures. Colour, brightness, material and transparent features, lights and shadows are added into space models [Kiviniemi and Penttilä 1995].

So after examining the currently available CAD technologies, the 2D-based approach using vector-based graphics seems to be the one recommended. The next section discusses the knowledge management.

2.6 Knowledge Management

The reason for having this section in this research is that it depends to a large extent on the availability of sound knowledge. And this knowledge was of immediate interest in answering the research questions of this project. Also the information in this section can be of great importance for the usability of the software produced and the options which need to be implemented.

If this information is not present then the designer cannot proceed. Whatever solution is finally proposed its success will depend to a large extent on the access to this kind of information. Implementation of this project's software solution would not be possible if certain shapes had not been studied for example the properties of shape.
Nonaka [Nonaka: 1998] states that in an economy where the only certainty is uncertainty, the one source of lasting economical advantage is knowledge. Knowledge management (KM), as defined by the Gartner Group (www.gartnergroup.com), is a discipline with new processes and technologies that differentiate it from information management. New technologies are required to capture knowledge that was previously unspoken. And unspoken knowledge is

embodied in the minds and expertise of individuals. Once captured, knowledge must be shared to add weight to its value and so that it can be reused in similar situations and contexts.

Knowledge is reasoning about information and data to actively enable performance problem solving, and decision-making, learning and teaching [Beckman: 1999]. Knowledge Management (KM) is the formulisation of and access to experience, knowledge, and expertise that create new capabilities enables superior performance, encourages innovation and enhances customer value. KM has emerged as an integrated, multi disciplinary and multi-lingual discipline providing methodologies and tools for identifying, eliciting, validating, structuring and deploying knowledge within the enterprise. From a management perspective, two major strands have developed within the discipline [Vergison, 2001]. The next section discusses the modelling approach in design.

2.6.2 Knowledge-Based Design

In this section the various Knowledge-Based Design approaches are investigated in an attempt to see which one would be more appropriate to use in the final solution proposed.

The first generation of Knowledge-Based Design Systems (KBDS) was characterised by the dominance of logic models and Rule-Based Systems then prevailing within expert systems technology. The paradigm of Knowledge Engineering (KE) appeared to be promising and relevant to design. (KE) turned out to be far more applicable to Knowledge Management (KM) than it is likely to form the holistic operational framework for globally enabled design and project environments. (KE) has limited use for the range and complexity of design tasks. Debenham [Debenham 1998:1] states that a unified KE methodology treats data, information and knowledge in a standardised mode.

However, with a few exceptions, models of expert knowledge appeared to have limited utility for the range and complexity of design tasks [Oxman: 1994]. An expert system is a system in which knowledge is represented as it is, possibly in the same form that it was extracted from an expert. In an expert system the represented knowledge should endeavour to solve problems in the same way as the expert knowledge source solved them. Debenham

[Debenham 1998] defines a Knowledge-Based System as a system that represents an application containing a significant amount of real knowledge and has been designed, implemented and possibly maintained with due regard for the structure of the data, information and knowledge.

According to Debenham data is the set of fundamental, indivisible things information is the set of implicit associations between data things and Knowledge is the set of explicit associations between the information things and/or the data things [Debenham 1998:20].

[Debenham 1998:23] identifies differences between Knowledge-Based Systems and expert systems where as Expert Systems perform in the way of a particular trained expert. A knowledge-based system is not constrained in this way. In a knowledge-Based System the represented knowledge should be "modular" in the sense that it can easily be placed alongside knowledge extracted from another source.
Furthermore, Expert Systems do not necessarily interact with databases. In general, knowledge-based systems belong on the corporate system platform and should be integrated with all principal, corporate resources.

CAD/ Engineering researchers have been focusing their attention on the Knowing aspects of the design-case process since approximately 1990. They have been constructing models of design knowledge and reasoning that have not proved themselves for design applications of substance.

Due to the complexity of design, systems for design have often defined the task with artificial narrowness [Hinrichs 1991:3]. In AI, as in Fuzzy Set theory, limiting the universe of discourse or even closing it in an attempt to simplify the enormously complex design problems made progress in the past. To make the systems tractable the following typical four approaches were used [Hinrichs 1991:3]:

1. **Selection.** Select components to instantiate a skeletal design.
2. **Configuration.** Arrange a given set of components.
3. **Parametric.** Fix numeric parameters.
4. **Constructive.** Build up designs from components.

Hinrichs observes the fact that if design problems are viewed as instances of the above-mentioned types; they can often be solved using efficient algorithms and heuristics. However, rigid classifications do not capture the flexibility that real designer's exhibit. In addition to the different types of design approaches, research has explored different approaches to the process of design. Hinrichs summarises some of these approaches as:

Pure synthesis: construct designs from the bottom up.
Hierarchical refinement: refine skeletal designs from the top down.
Transformational approach: mapping from equation to structure.
Case-Based Design: the case-based and analogical approaches assume that the problem being solved is probably similar to one that was seen before.

Currently the most promising solution is the use of design cases (CBR). This has empirically validated successful and failed solutions to design problems from the past. If structured design methodologies are to be used, then any design knowledge generated should be stored in such as way as to expedite future designs. Then next section is discussion on Artificial Intelligence and design.

2.6.3 Artificial Intelligence and Design

Although in the late 1950s Allen Newell and Herbert Simon proved that computers could do more than calculate, and it was said that within a generation the problem of creating Artificial Intelligence would be substantially solved, the field of AI ran into unexpected difficulties. The trouble started with the failure of attempts to program an understanding of children's stories. The program lacked the common understanding sense of a four year old and no one knew how to give the program the background knowledge necessary for understanding even the simplest stories.

AI is based on the Cartesian idea that all understanding consists in forming and using appropriate symbolic representations. For Descartes, these representations were complex descriptions built up out of primitive ideas or elements. Dreyfus [Dreyfus- 1993: *xi*] states, *"Common-sense understanding had to be represented as a huge data structure comprised of facts plus rules for relating and applying those facts."*

AI struggles to cope with essentially three main problems, [Dreyfus 1993: xviii]:
How knowledge can be organised so that inferences can be made. Then how skills can be represented as knowing-that; and how relevant knowledge can be brought to bear in particular situations.

(Dreyfus 1993: *xxxix*) *"The point is that a manager's expertise and expertise in general, consists in being able to respond to the relevant facts. A computer can help by supplying more facts than the manager could possibly remember, but only experience enables the manager to see the current state of affairs as a specific situation and so see what is relevant."*

CAD researchers became interested in AI due to the frustrations with the unintelligent nature of commercial CAD systems. Even today CAD is contributing very little to the initial and most demanding stages of design. AI is generally concerned with tasks whose execution appears to involve some intelligence if done by humans. Design falls into this category.

AI research can be divided into two broad approaches.

1. ***Understanding of the human brain:*** computer models in this tradition represent a model or simulate human cognition and succeed to the degree to which they emulate human performance.
2. ***Intelligent systems:*** *these are systems* that perform intelligent tasks effectively without concerns for how faithfully the model simulates human performance or cognition acceptance.

Computers that work exactly like people are unlikely to do better than people. CAD tools, whether AI based or not, should always be seen as a complement to human designers, assisting them in tasks where they perform less well, but do not compete in areas that the human brain performs well.

Programs that assist in design are most useful in the following areas: they suggest possibilities to designers they have not thought of, and remind them of things they might have forgotten.
The author will attempt to prove that, in addition to these two possibilities, a third option exists. This is where intelligent components are used to facilitate the manipulation of

complex design information in a convenient environment to facilitate concept selection and design experimentation during the early phases of design. During this phase the designer is often confronted with incomplete information and designs could very easily change. At the same time decisions taken during this phase will significantly influence operational characteristics. Next section is discussion on Visual reasoning.

2.7 Visual Reasoning

Visual reasoning (e.g. thinking in shapes, forms and images) is a fundamental attribute of shape designs, and therefore combining it with CBR may provide significant results both for the field of design thinking as well as for the field of Computer-Aided Design (CAD). All these three elements combining together will provide a high level of benefit for the research. When visual mental shapes are formed, the reasoning processes access the stored representation of the structure of an object in associative memory [Kosslyn and Osheron 1995]. The ability to access the underlying structure, a concept, a schema, a drawing is significant for our ability to reformulate images/design. The reformulation of these visual images is one of the cognitive foundations of emergence in design [Oxman: 2001]. So from this, a system that would allow the images of these previous designs always to be there and to be looked up when needed would benefit a designer. Essentially, a design comes into being through the manipulation of non-verbal information: the visual is the way in which the designer knows, thinks and works. The centrality and power of visual reasoning as a cognitive mechanism makes design, in general, an ideal field for CBR.

Furthermore, it suggests interesting possibilities with respect to the incorporation of visual material in computerised design case libraries, and the potential to interact with and exploit visual case data in the process of computationally supported design. The concept of case bases for casting design needing visual and/or diagramming is supported by this consideration:

Firstly, most designers (painters and engineers) prefer to sketch than write down early design ideas. They *sketch diagrams* to explore possible adaptations of old cases to current design tasks. **Secondly,** design tasks that deal with layout configuration such as arranging components and region for shape often benefit from previous cases of success or failure.

2.8 Conclusions of CBR and Related Issues

This chapter discusses why Case-Based Reasoning was reasonably used in this research. CBR should use the information created by the designer during the design process. This thesis suggests the Case-Based Reasoning approach. It has been shown that this allows us automatically to gain a new case, and match the similarity of cases. It also makes it possible to adapt old components to new cases using derivational replay. It is stated, in this thesis, that a conceptual framework is defined using a number of 2D cross sections or views. A number of views represented 3D shapes by cutting 3D shapes into dissimilarity views (CAD application) for target and it is shown how this can be captured. Although promising, issues regarding the shape matching of such a framework are indicated as the main approach. Kolodner [Kolodner, J.:1993] suggested CBR depends on the method of parameter adjustment for interpolating values in a new solution based on those from an old shape. In parameter adjustment, changes in parameters in an old solution are made in response to differences between problem specifications in an old and a new case.

2.9 Summary of literature review

Other approaches to assist casting design have been used. The literature shows the advantages and limitations of such approaches. CBR provides an alternative way to solve this problem. The research in CBR shows that although not a lot of applications have been pursued in the area of casting, the approach provides some advantages in managing casting design knowledge. Mileman [Mileman: thesis 2000] research has demonstrated the feasibility of a CBR system to assist the casting design process, but the work in that research did not show how the process can be automated and also had some limitations such as efficiency of retrieval and not dealing with 3D shapes. The research in this thesis aims to further the understanding on how CBR systems can be efficiently be used to retrieve and reuse useful and applicable information to assist real casting design processes.

The aim of this chapter was to establish the domain of this research by provide a concise overview about the research keywords in brief and in more details about the main research keyword namely; CBR and of the four main tasks involved in the CBR cycle, namely retrieval, reuse, revision, and retention. Rather than presenting a comprehensive survey, we have focused on a representative selection of work from the CBR literature in the past few decades. We have tried to strike a balance between research that can be seen as laying the foundations of CBR and more recent contributions. The fact that many of the cited papers were published in the last few years is also evidence of a significant amount of ongoing research activity. It should be clear from our discussion that much of the recent research has been motivated by an increased awareness of the limitations of traditional approaches to retrieval, reuse, and retention.

Chapter 3

Shape Decomposition

In previous chapter the domain of this research was established that is the use of the Case-Based Reasoning (CBR) methodology to assist with casting design. The various well-established techniques such as Knowledge Management, Computer Aided Design (CAD), Knowledge-Based Design, Case-Based Reasoning, and more have been examined in an attempt to discover if improvements can be made to the early phases of casting design decision process.

The aim of this chapter is to discuss the design of an efficient algorithm automatically decomposing a number of 2D cross sections or 'views' of a 3D shape into generic connected components.

3.0 Overview of This Chapter

The body of this chapter is organised into eight sections and sub sections. They are explained briefly as follows:

Section 3.1 provides an introduction of the shape decomposition problem. Section 3.2 poses the background problem. In Section 3.3, there is an overview of shape description, which plays an important role in shape decomposition. Section 3.4 presents the key contribution to this chapter that is how shape partition algorithms can automate the casting design process. Section 3.5 gives an overview of related work and its relation to the current project. Section 3.6 gives the details of an algorithm for shape decomposition. Sections 3.7, discusses the implementation and evaluation of the algorithm using a number of experiments. Section 3.8 the conclusion of this chapter.

3.1 Introduction

The aim of this chapter is to present a shape decomposition process that will aid the case-based reasoning (CBR) system. An algorithm has been designed to decompose a number of 2D cross sections or views that represent 3D shapes into generically connected components. This method can assess the approximate shape, by breaking down the shape into subsets of disjointed components. This is done by projecting horizontal and vertical lines from vertex points. The algorithm (described in section 3.6) was tested on a large set of 2D cross section views that has been generated from real 3D shapes using a CAD application. Experimental results are presented in section 3.7 of this chapter. The experimental results demonstrate that the proposed shape decomposition algorithm can segment complicated shapes automatically and efficiently into meaningful connected components.

This research uses a number of 2D cross sections or views to represent 3D shapes. It would be more realistic to consider three-dimensional objects as these are the objects we encounter in our design environment. However, the study of 3D objects is much more difficult than that of 2D shapes. One reason for this is the ambiguity that results from the projection of the 3D object onto the 2D shape. Although easier than the 3D cases, the analysis of 2D shapes is still a very challenging and interesting problem. In addition, the ideas and methodology developed from analysing the 2D case could help in addressing the more general 3D case.

In fact, from investigatory work in the area [Knight, et al 1995] the casting engineers work with and reason using 2D shapes [Aziz casting design engineer private communication 2003]. A vast number of researchers (Mileman: 2000, Kotschi and Plutshack 1981) have simplified the evaluation of 3D shapes by using a slicing technique to simulate 3D shapes as 2D slices. Mileman assumed that only one 2D view represents a 3D shape, but in this research 3D shapes can be represented by one 2D view or a number of 2D views. (See on Fig.3.4).

For this research, two groups of 3D shapes are considered for the decomposition process:

(1) Axisymmetric Shapes and
(2) Arbitrary 3D Shapes (see Fig.3.4).

The process proposed in this chapter starts with a shape decomposition of 3D models into a number of 2D cross-sections or views through CAD modelling. Next these 2D cross sections are automatically decomposed into the six generic components. The primitive components proposed by [Mileman: 2000], (Fig.3.0) the six generic components:

- L shaped (L)
- T shaped (T)
- X shaped (X), Tapers and two types of elements such as Bars (Type (A) and type (B)) and regions this part will be discussing in details in chapter 4 and chapter 5.

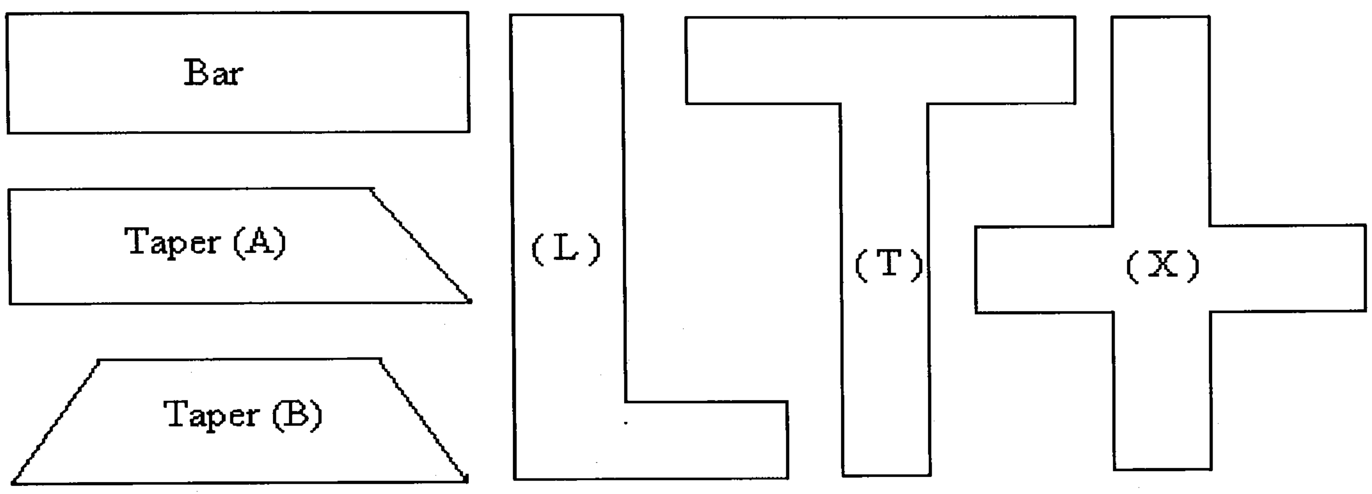


Fig.3.0.The six generic components: shows the proposed primitive components (L, T and X) and Bar and Tapers elements [Mileman, Thesis: 2000].

Over 100 3D shapes went through slicing process using CAD application, mapped into a number of 2D cross sections, which represented the 3D shapes. Those views were tested over the CBR-Shape system, developed at Greenwich University. Experiments were conducted on a large number of both artificial shapes and other 3D models provided by previous research. Experimental results demonstrated the performance and efficiency of the decomposing algorithm and the details for testing are given in the chapter evaluation.

3.2 The Background

The aim of this chapter is to discuss the analysis and design of an algorithm for the shape decomposition problem. It is one of the research questions: Can the decomposition process break down a given number of cross-sections or views into generic component types such as L-component, T-Component, X-component, Taper and bar element. [Mileman: 2000].

Decomposition is achieved by automatically splitting the given shape into subsets of disjoint components until a primitive (Fig.3.0) is recognized, as illustrated in Figure 3.1 and see chapter 4 classifications. The shapes are constructed from geometric Rectangles and Triangles. They can be simple or complex, so their attributes be stored in a database for later retrieval for other uses.

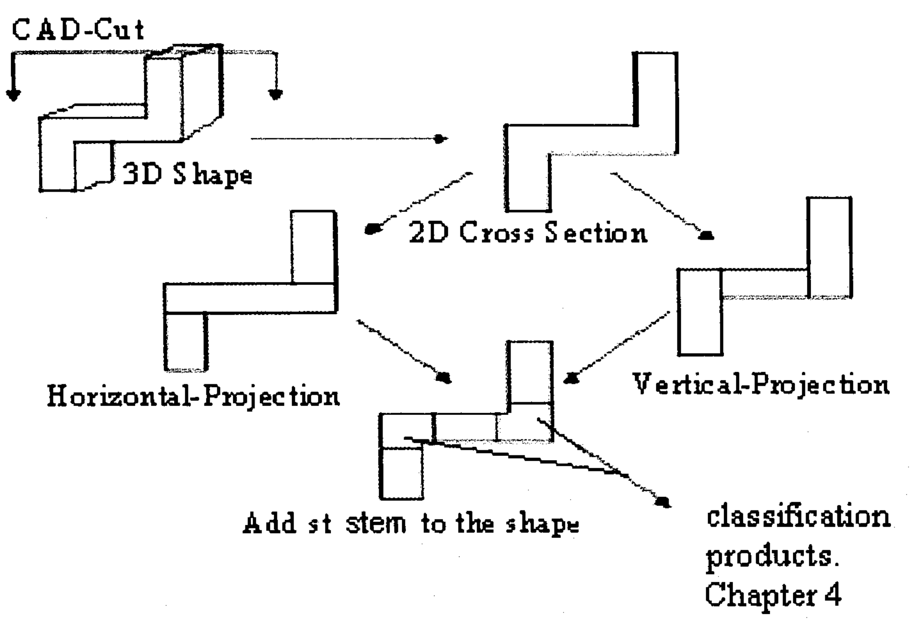


Fig.3.1 Illustrates the hierarchal of 3D-2D for possible shape decomposition.

3.3 Shape Representation

Before designing the algorithm for the shape decomposition process, shapes need to be described. This section gives an overview for shape description in brief. Shape description plays an important role and it's a fundamental problem in shape recognition and engineering design perspective.

Various methods for shape description have been suggested through the years of research in engineering design and human perception, but none provide a complete and natural solution to the problem. Furthermore, this problem seems to be one of the most challenging problems, and is perhaps equivalent to the vision problem itself. The shape in this work refers to the outer form of the objects, or more specifically, to the geometry of an object in three dimensions, or to the bounding geometry of an object. In many two dimensional cross-section views, it should be capable of describing partially parts.

Figure 3.1 describes a shape in detail using geometrical information such as properties and structure of the shape as it is made up of vertices and edges. Vertices are used to determine the types of components (L_component, T_component, X_component and Taper component

[Mileman: 2000]. Edges display the connections between the components, which give different types of elements such as Bars (Type (A) and Type (B)). The connection works as a bridge between the components and it means that the connectivity between the components plays an important role in shape recognition. This will be further discussing in details in Chapter 4, Figures 4.4 and 4.12.

Table-3.0 can be eliminated by letting the shape table reference the vertices (V) directly, rather than drawing up twice, because it is not realised that the same set of points has been visited before and that the edges, being connections, share two regions in between. We could go further and eliminate the (V's) table by listing the entire coordinate explicitly in the shape table, but this wastes space because the same points appear in the shape table several times.

Using all three tables also allows for certain kinds of error checking. We can confirm that each shape is closed, that each point in the V table is used in the edge table and each edge is used in the shape table.

A table also allows us to store additional information in the future like components that are sub classes for the shape. Each entry in the edge table could have a pointer back to the shapes that make use of it. This would allow for a quick look up of those edges (see steps in Fig.3.1).

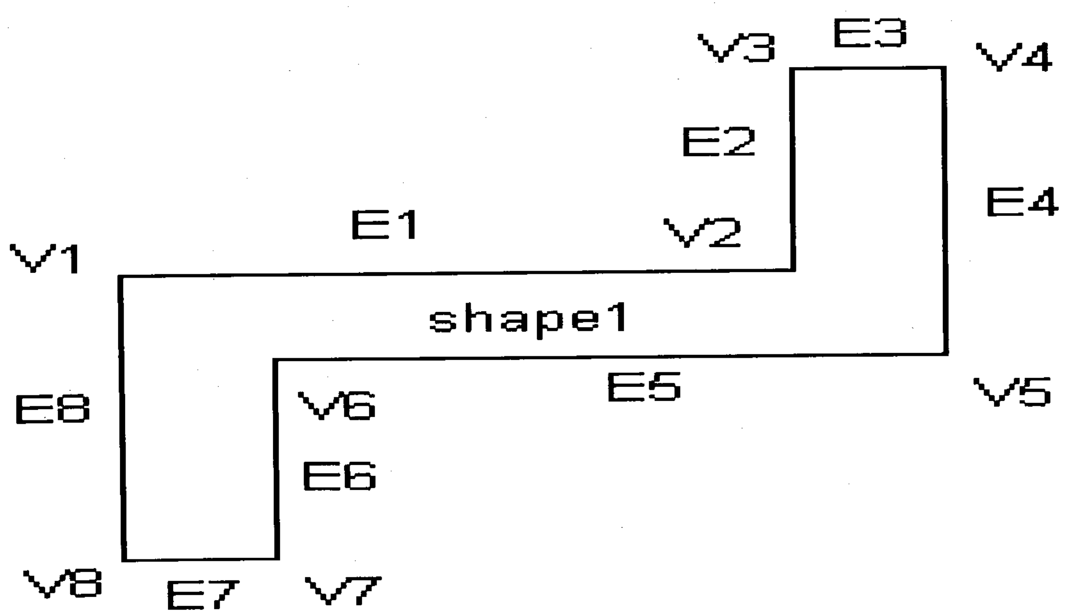


Fig.3.2 Shows description of 2D cross-section shape1.0.

Vertices (V) table	Edges (E) table	Shape tabe
V1= x1, y1	E1=v1, v2	Shape1 = E1, E2, E3, E4, E5, E6, E7 and E8
V2= x2, y2	E2=v2, v3	Component table = L-shaped or L-region
V3= x3, y3	E3=v3, v4	
V4= x4, y4	E4=v4, v5	
V5= x5, y5	E5=v5, v6	
V61= x6, y6	E6=v6, v7	
V7= x7, y7	E7=v7, v8	
V8= x8, y8	E8=v8, v1	

Table-3.0 shows current research shape representation (analysis).

3.4 The Shape Decomposition Process

Many tasks in computer vision, computer graphics, and reverse engineering are performed on objects or models. Those object become more complex when the treated object geometry is complicated, for example, when it contains multiple components. Therefore, shape decomposition is attractive since it simplifies the problem with multi-part, complex objects into several sub problems dealing with their constituent single, much simpler shapes. In application areas of object recognition, shape description representation and object manipulation, shape decomposition is a crucial pre-processing step, and can further reduce the efforts involved with the original multi-part objects [Hoffman D. and Richards W.:1984 and Pentland A.:1981]. While a significant amount of research for shape segmentation or decomposition of 2D shapes has been conducted over the last two decades [Hoffman et al 2000], little effort has been made on shape segmentation of 3D models [Rom H. and Medioi G.:1994] Rom proposed "a framework consisting of decomposing 3D objects into single components and then describing those parts by higher-level primitives, such as generalised cylinders". Additionally, this work is able to handle 3D shapes by using CAD packages to draw the shapes and the CAD slicing the shape into numbers of 2D cross-sections or views, even though 3Ds are the most dominating representation elements in the 3D computer graphics world. Wu [Wu K.:1997] presented a physics-based part segmentation approach.

The novelty of this method projecting (H) horizontal lines and (V) vertical lines; is that the shape's properties will be identified and all hidden geometrical information about the shape will appear and these new attributes determine the types of components by using this idea of projecting (H) and (V) from each vertex (Hotspot).

Hotspot is an important point for decomposition which is made up from two connected original lines and only from this point the projecting lines (H) and (V) can go through. The original lines only concreted the boundary of the shape. (See Fig.3.7).

The advantage of shape segmentation is low computational cost in computing geometric shapes. Alternatively a surface segmentation method was proposed in [Mangan A. et al 1999], based on either planar surfaces or arbitrary shapes. The disadvantage of this method is the limited usefulness of surfaces compared to equation parts in high-level tasks such as object recognition [Wu K.:1997]. For example, a 3D triangulated model composed of a cube and a cylinder will be segmented into six planar surfaces and a cylinder by the surface segmentation algorithm, while a part decomposition algorithm can decompose the model into its constituent parts (the current case divided into elements, components. and regions, a cube and a cylinder. In summary, there is a lack of part decomposition algorithms for handling 3D symmetric shapes based on projection analysis. However methods for handling number of 2D cross-section view segment models into types of components are useful. Therefore, in this thesis, we present the first attempt to decompose numbers of 2D cross sections or views into types of element, component, and region by techniques of horizontal, vertical and diagonal projections. The proposed algorithm is easy to implement, and it is able to handle a large number of 2D cross sections or views (2D views represents 3D shape).

A lot of the work discussed above has been conducted for the purpose of decomposition of shapes into generic components for various application areas. For the purposes of this research the required shape decomposition context is that of decomposition casting shapes into generic components to allow for the re-use of useful casting design knowledge through CBR retrieval of shapes based on similarity metrics.[Mileman: 2000] demonstrated the feasibility of this approach, but he used a manual approach to do this decomposition process. However, in order to produce useful CBR based casting design tools, it will be important to automate the decomposition process. This will make the tool effective and efficient in the creation of new target (query) cases and in the maintenance of the case base by the addition of new knowledge encoded in new cases.

In previous research Mileman [Mileman 2000] assumed that one 2D cross-section or view representing 3D shapes is enough to represent 3D shape. Although this is true for axisymmetric shapes, it is not true for a complex arbitrary 3D shapes. However there are many 3D complex shapes that can be viewed by a finite number of distinct sections, typically two to three. But slicing the 3D shape into numbers of 2D cross-sections or views we can elicit more details about the shape, such as structure and properties which may be different for each view section.

Furthermore, when the number of these 2D cross sections go through the decomposition process, new geometrical information or hidden information will appear because a 3D shapes is complex and we never know the internal geometrical information of the shape unless take the 3D shape through CAD application to sliced into different views to see the geometrical details that the engineers it does. 2D cross-sections or views are easier to display than 3D shapes [Kotchi and Plutshack 1981]. To do a complex 3D casting design, needs to cast several views in order to design a 3D casting shape an casting design expert poses[Aziz: 2003]therefore our 3D shapes represents by one or number of 2D cross-sections or views. The next section is discussion on the measurement of geometric internal information for 3D shapes

3.4.1 Measuring Geometric Internal information for the Shape

In the shape decomposition discussed in Section 3, the decomposition technique is based on the horizontal, vertical and diagonal projection method to decompose numbers of 2D cross-sections or views into different proposal components. The algorithm generates new points and new lines such as projected lines and projected points. All these new attributes are internal geometrical information for the 2D shape; they are considered to be the key attributes for improving the quality of the task; the internal is an adjunct of information of the shape can be blocks for measuring complexity internal attributes, and they are crucial for evaluating the efficacy of software methods. The next section discussion is on the feature extraction slicing.

3.4.2 Feature Extraction Slicing

Preparations for the Shape Decomposition Process

This section gives some clarification about the 3D shapes. Ideally, 3D shapes are drawn using a CAD application that is one of the most advanced engineering applications [Manuel J.: 2004], and using algorithm to generate CAD modules automatically. Then, selecting a high enough number of views for the shape, the system would cut these slices from CAD components, and the suggested decomposition process will break the 2D cross sections into identifiable elementary components.

In fact, the manuals on engineering design use many cross-sectional slices to see the geometrical information of the 3D shape to view other details. For example: civil engineers have to draw a 2D diagram to show the wall, details, doors and other information of the house. Slices are fundamental keys in processing shape decomposition for casting design problem. As tested in the section on shape decomposition, in our approach each 2D slice is used as a basic descriptive unit for testing. This is justified under the assumption that they (see Fig. 3.4) are approximately symmetrical. Thus each 2D cross-section slice is potentially equivalent to other slices, which is also useful in identifying other slices. Also 2D cross section shapes are easier to display than 3D shapes [Kotchi and Plutshack 1981]. The first step in this method is to extract a set of components with easily computed features from each 2D slice. For more details see the chapter on similarity metrics. To justify the use of cross sections further, Fig. 3.5 "2D cross sections for 3D Mug and 3D shape drawing examples".
This research deals with two types of 3D shapes, Axisymmetric Shapes and Arbitrary Shapes. A number of research projects [Kotchi and Plutshack 1981] have already investigated that one objective; a way to define the geometrical complexity of a 3D shape is to ask how many different cross sections are required to describe the shape. In the case of LIGA (is a model of shape), only one cross section is required (see Figure 3.3), thus surface micromachining, the primary MEMS (is a model of shape) process and three to five cross-sections are sufficient (see on Figure 3.4). EFAB (is a model of shape) shapes can be produced with such a high degree of complexity that hundreds or even thousands of cross sections are needed to describe them. Fig. 3.4 shows an extremely-complex device that might be fabricated with EFAB, giving a sense of what 3D dimensionality really means. Arbitrary 3D objects are what one sees when one lifts the hood of one's car observe the engine. Imagine an engineer at

General Motors being asked to design car parts that need only a handful of cross sections to describe them! These are the kinds of constraints that developers of Microsystems have had to live with so far. The same situation holds true for complex casting designs.

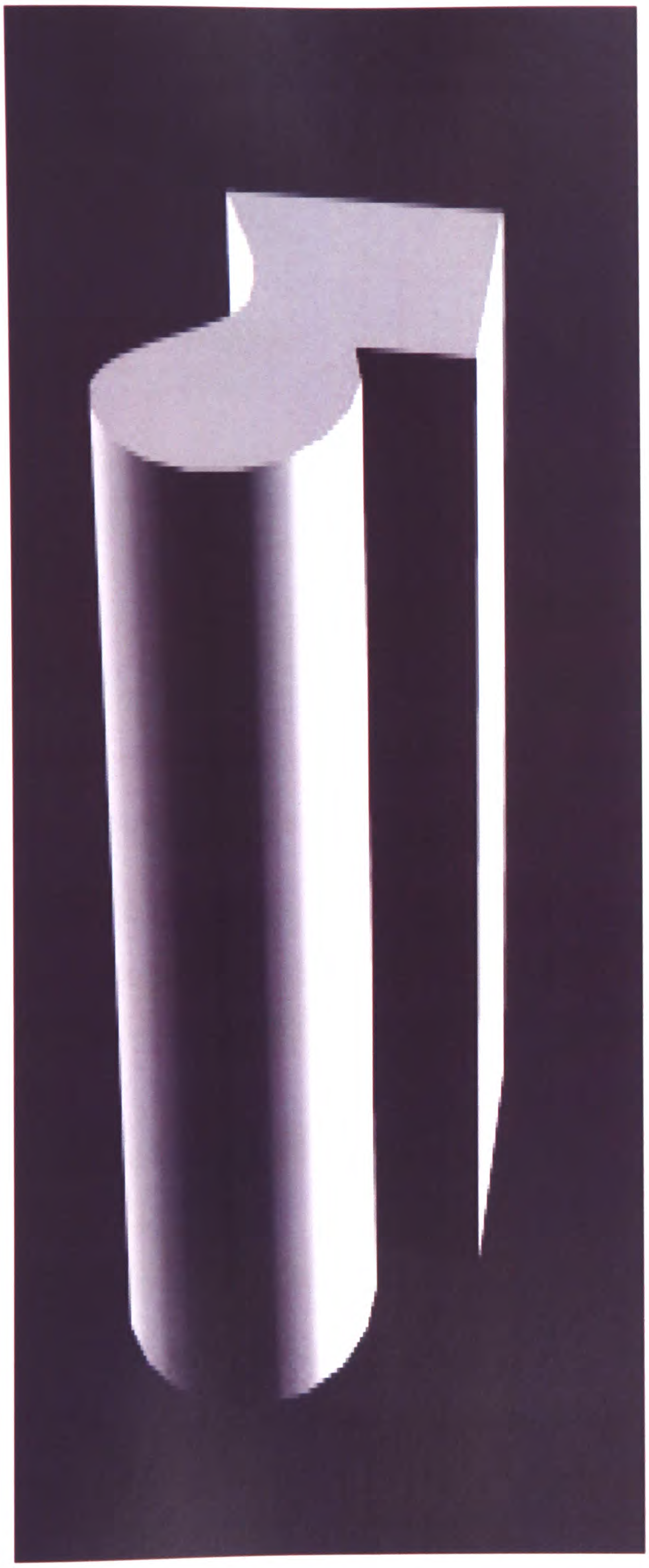


Fig.3.3. 3D shape that can be described by a single cross section view.

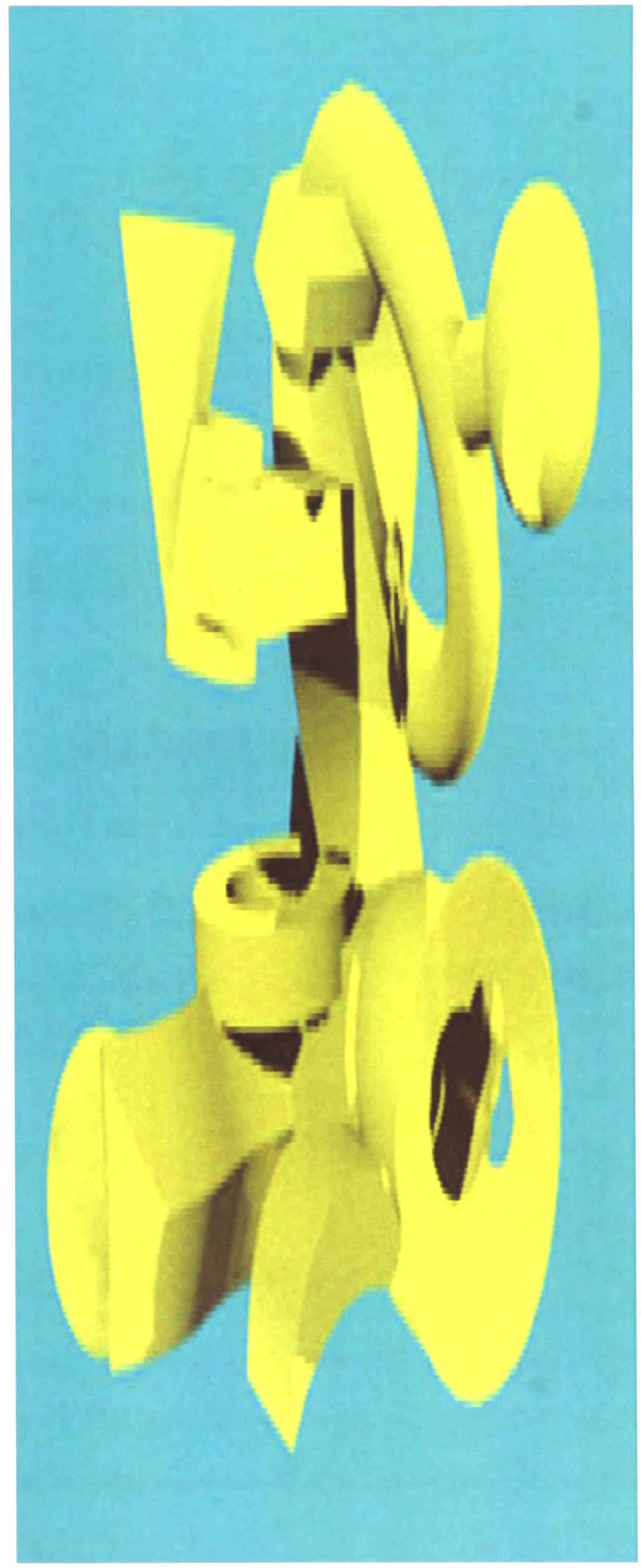


Fig.3.4. An arbitrary 3D that requires many unique cross sections to describe it.

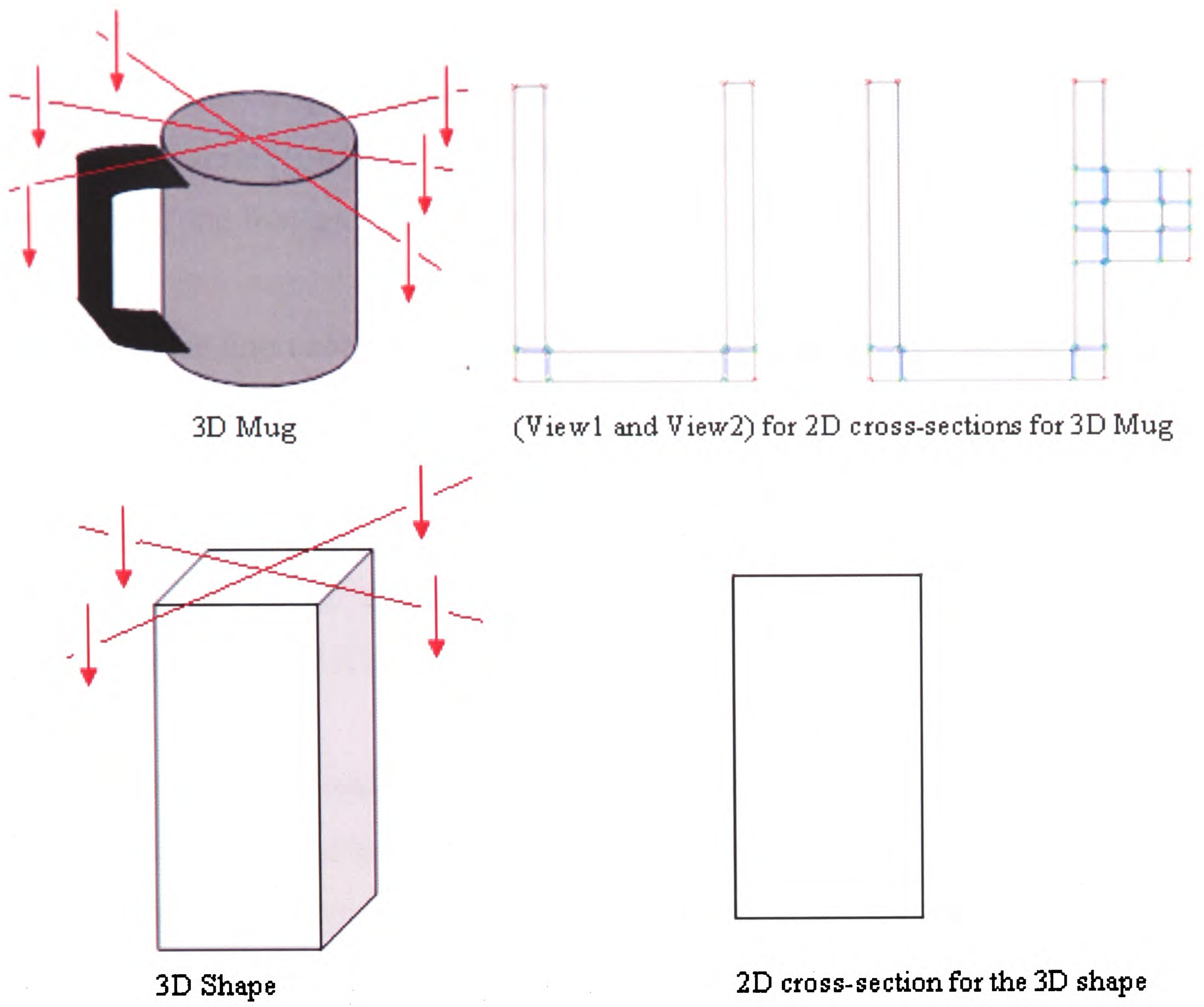


Fig.3.5 shows 2D cross sections for a 3D arbitrary shape for the mug.

Complex 3D shapes have one or more cross-sections or views. The other 3D axisymmetric shape has only one view. The next section presents an overview of the related work on shape decomposition which has been done in the past by other researchers.

3.5 Related Work

In this section work carried out by other researchers in this area is presented.

Decomposition of shapes is studied comprehensively in the computer vision community but there is still a lack of useful research, especially in the geometric shapes field. There are a number of problems that remain yet to be resolved, such as automating shape decomposition in an efficient way for casting design and finding useful methods for decomposition process Mileman [Mileman: 2000]developed a decomposition process by manually slicing the shape into connected and generic components and used projecting technique.

Xuetao Li and Tong Wing Woon [Xuetao Li and Tong Wing Woon 2001], developed an efficient framework to decompose polygon meshes into components that adopts the idea *of edge contraction a space sweeping* to decompose into objects automatically. The generalised cylinders method [Binford T.O. 1971], Geon's [Biederman I. 1987], Super-quadric [Hertel S. et al, 1984] and their extensions were used in 2D images and range data.

Though these approaches focus on acquiring components with identical features, as in our approach, there are no important extensions to work on 3D shapes.

Another work on volumetric objects was presented by Gayvani [Gayvani and Silver 2000]. Decomposition of 3D (volume) digital shapes is based on a hierarchical decomposition method developed by [G. Borgefors et al, 1999].

Lopes [Lopes A. M. and Metha P.M.: 1994] used a method that it is quite closely related to our current decomposition shapes, but only horizontal projection has been used to partition a polygon into rectangles and L-shapes and the decomposition process done manually.

On object decomposition process in general, Tan [Tan T. S. et al 1999] argued that he achieved good results in decomposing objects through the use of vertex-based simplifications. This approach works well for geometric and inorganic models such as, bottle necks, helicopters and a donkey skeleton. However, some of these methods do not support geometric and inorganic shapes and were found to be unsuitable as these models do not have any clear boundaries among their parts or components.

[Simmons M. and Sequin H. C.: 1998] developed an automatic system to generate a hierarchical 2D object representation especially for geometric tasks. Their approach is based on the axial generation module that could be replaced by an alternate construction, like that used in producing cores [Burbeck A.C. and Pizer M.S.: 1995].

Lopes [Mario A. Lopes and Dineshp P. Methat 1994], presented two practical algorithms for partitioning circuit components, represented by rectilinear polygons, so that they can be stored, by using the L-shaped corner stitching data structure. That is, the algorithms decompose a simple polygon into a set of non-overlapping L-shapes and rectangles by using horizontal cuts only [Nahar and Sahni 1988]. Nahar studied this problem as well and presented an object (kv) algorithm to decompose a polygon with n vertices and (kv) verticalinversions into rectangles using horizontal cuts only. In the extension to corner stitching, it was proposed by Blust and Mehta [Blust and Mehta 1993] that the data structure stores L-shaped tiles (hexagons) in addition to rectangular tiles. This L-shaped variant of corner stitching was motivated by a need for a data structure that could store rectilinear shapes more general than rectangles [Shanbhag et al. 1994; Mehta et al. 1995]. L-shaped objects, in particular, have been studied in the context of floor planning [Wang and Wong 1990; Yeap and Sarrafzadeh 1993] and routing [Dai et al. 1985; Cai and Wong 1993]. Once again, because circuit components can be rectilinear polygons that are not rectangles or L-shapes, these components needed to be partitioned in order to be stored in the L-shaped corner stitching data structure. Furthermore, using horizontal cuts for partitioning is desirable, because it simplifies the implementation of the operations (which are now more complex than for the rectangular corner stitching data structure).

We note that this problem is different from the problem of decomposition for a rectilinear polygon into a minimum number of rectangles using both horizontal and vertical cuts, which has been studied extensively in the literature.

This motivated the need for fast and practical algorithms for decomposition shapes into sub-set disjoint types of rectangle shapes, using only horizontal and vertical projecting methods this methods have been introduced for current decomposition problem. The gain of projection methods is the increase in the number of components through vertical, horizontal and diagonal projections, to optimise the similarity during the shape comparison between properties of the source shape and the target shape. More details are given in chapter 5. The next section gives an overview of the algorithms.

3.6 An Overview of the algorithm for decomposition

In this section, a broad framework for the shape decomposition algorithm for casting designs is described.

An algorithm was devised to provide for the automatic decomposition of shapes into the generic components used in this research (Fig. 3.0 see the shape decomposition diagram). This algorithm starts by projecting horizontal and vertical lines from each *hotspot (for hotspot definition see section 3.6.1(i) algorithm) to the nearest existing original lines. This provides a decomposition of the area of the shape into a set of rectangles and triangles. These are then reconciled and their sides merged defined by only internal points that connect them. A set of rules then identifies each element as one of the generic components needed for the componentisation of the shape.

Finally, the components are created by adding "stems" where appropriate (typically to joins, such as L, T and X). Figure 3.5 shows an example of such decomposition. Observe the top left L-component. In the middle figure, the algorithm has identified a rectangle there. The rule that identifies this as an L-component relies on the fact that this rectangle has two adjacent sides (right and bottom) that are internal lines. This identifies the L-component from the hotspot. A hotspot is an important point for decomposition process, is made up from two connecting original lines more detail can be seen on section 3.6.1 Geometric Algorithm.

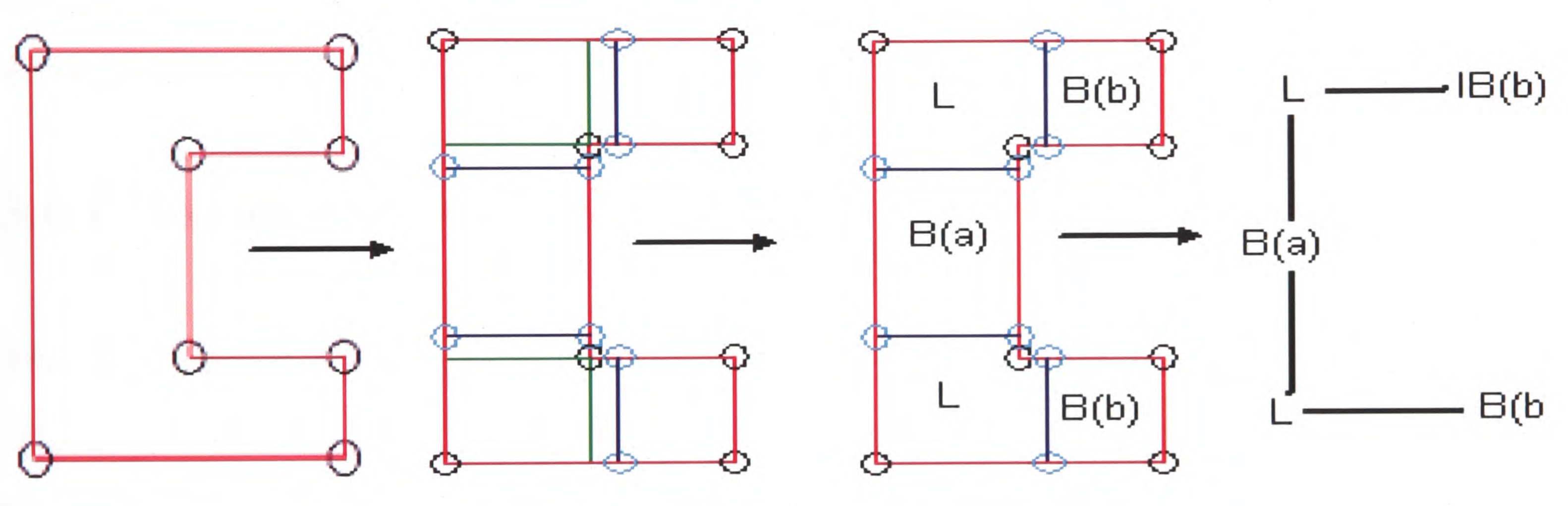


Fig.3. 6 shows an example of such decomposition produced by ShapeCBR.

An additional advantage of automating the decomposition into components is that the output of this process is not only the graph of connected components representing the structure of the shape, but also the association of each component with geometrical information describing the exact dimensions of the component (Fig 3.6 shows the type of connections between the components). This allows one to extend the definition of similarity between shapes, taking into consideration the actual geometry in addition to just the layout of the components in each shape.

Bar (A), is connector has two connections connected into two L-Components (see in Fig.3.6). This bar is not leaf it is a connector types of Bar.

Bar (B), is a connector has only one connection. The other face is free: we call it a leaf as well and it is a connector element. Table 3-1 shows the connection types by a number of nodes which play an important role in object recognition.

Component types	No_Of_connection	Connection type to
B (a)	1	L – down
L-down	2	B (b) & B (a)
B (b)	2	L-down & L-up
L-up	2	B (a) & B (b)

Table-3.1 describes the casting design engineering method in details for the decomposition process.

The table above describes the component types in the first column to the left of the table and the middle column shows the number of connections between the components. For example the (1) represents the connector bar from type B and has only one connection. The last column shows the position of the component types. The next section is discussion on, shape decomposition algorithm for casting designs.

3.6.1 Decomposition Algorithm

The first step of the decomposition algorithm is primarily based on the identification of "Hotspots" for shapes. A "Hotspot" is an important point for decomposition process, is made up from two connecting original lines. This point only concerns internal geometrical information for the shape. A Hotspot is one of the vertices of the original point of a shape, and its position is different from other vertices, because it is only from these points that penetration into the inside of the shape. Once the Hotspot is found, it maybe possible, using the projection (horizontal and vertical) method to decompose the shapes into rectangles and rectangle primitive elements. (See on diagram 3.7 shows hotspot position and the internal geometrical information such as projection lines.)

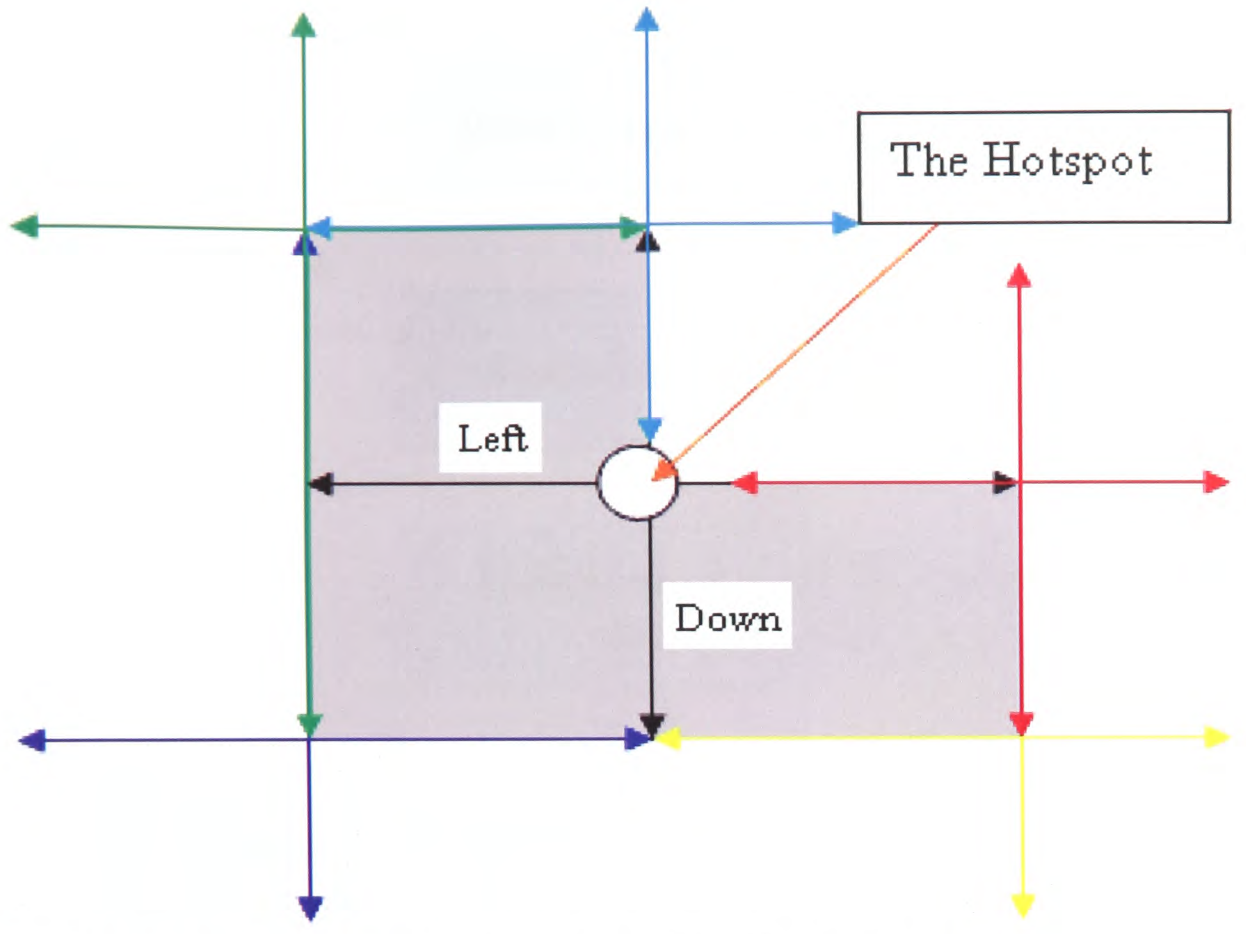


Fig.3.7 the diagram shows only internal projection identifies the hotspot.

The main issue is to how to identify the hotspot. (See the definition on section 3.6.1).
To recognise the hotspots, it is essential to initialise the direction of reading the data for the shape. Table 3-2 below, clarifies the entity of hotspot.

Direction (1) Clockwise	**Direction (2) anticlockwise**
Non hotspot the points to be skipped	Hotspot stores in hotspot collection
▪ Up >>>> Right	▪ _Left >>>>Down
▪ Right >>>down	▪ Up >>>>>Left
▪ Down >>>Left	▪ Right >>>>Up
▪ Left >>>>Up	▪ Down >>>Right

Table 3-2 shows the steps for skipped points by directions to identify the hotspots.

Table 3-2 shows the steps for identifying hotspots. The first column to the left represents the direction clockwise of reading data and the second to the right represents anticlockwise. The idea for this algorithm is to spot the hotspots. It's an important issue which will be dealing with the geometrical internal information of the shape. Figure 3.9 gives an overview of the shape decomposition algorithm and the hotspot. The next section explains the steps for the shape decomposition process and it illustrates the steps in a flowchart. Figure 3.8 shows the details for each step.

The algorithm steps for shape decomposition

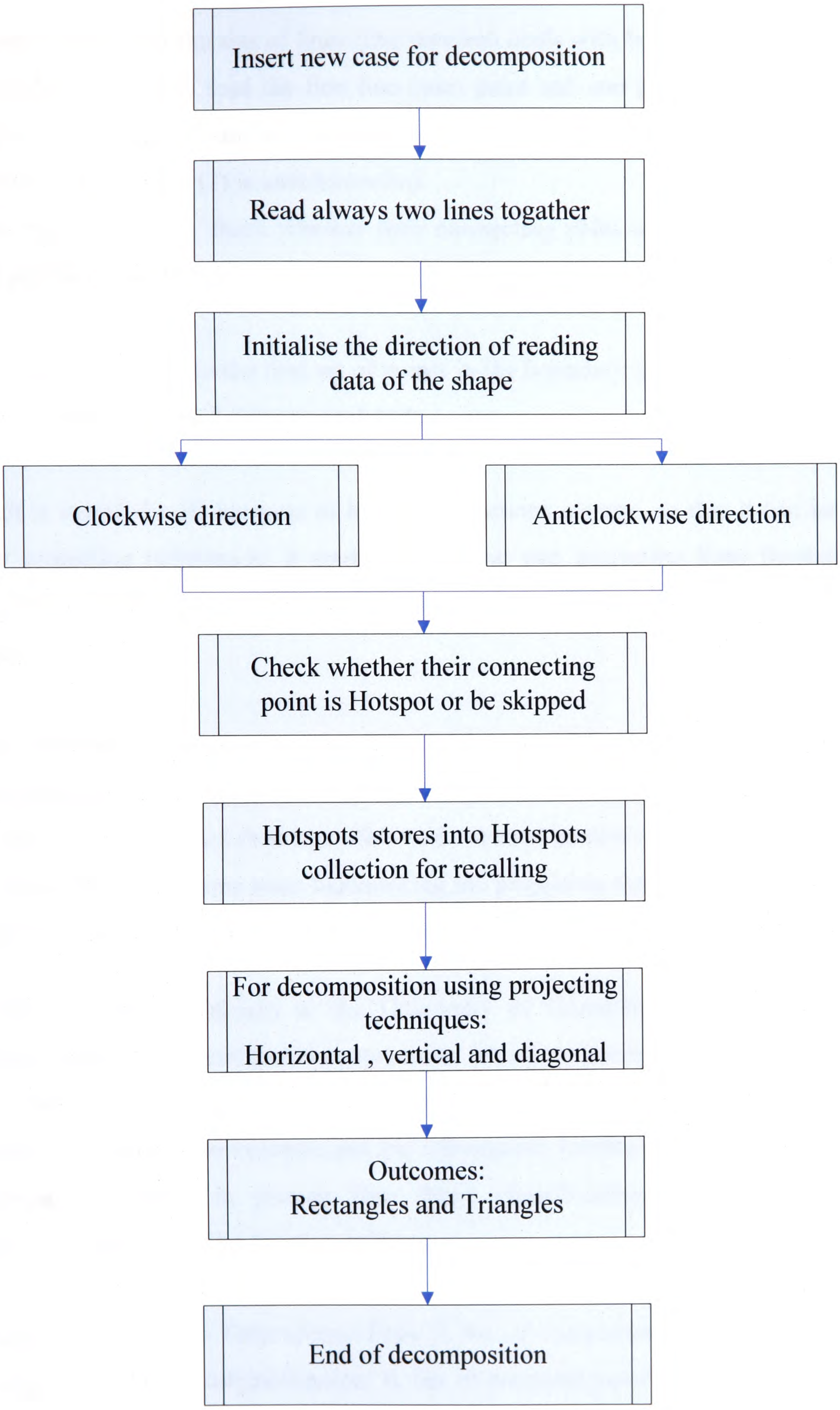


Fig.3.8 The Flow chart shows the Shape Decomposing Algorithm.

3.6.1.1 The discussion on the Algorithm

Read Lines: Firstly read the sets of lines (this research deals with both one set and two sets of shapes (cycle)); secondly read the first line (start point and end point) and thirdly read the second line. (start point and end point). Specify the direction of the reading data (direction (1) is clockwise and direction (2) is anticlockwise).
Take two success lines to check whether their connecting point is hotspot or to be skipped (skipped points are not hotspots and are simply concerned with the boundary of the shape).

This procedure is applied to the first set of points in the boundary of the shape. See Table 3-2 for skipped points. See Fig 3.7 for more details.

The result is stored for all hotspots to hotspot collections storage, so that it can be recalled later. By projecting techniques: It starts by drawing two projecting lines (horizontal and vertical), from each hotspot, to the nearest existing original lines according to the direction of projecting.

The above method provides the decomposition of the area of the shape into a set of rectangles and recto-triangles. These are then ready and merged if sides defined by internal points only connect them. A set of rules then identifies each element as one of the generic components. Fig 3.9 shows the three green lines representing the projecting technique from two hotspots. The output for this projection deals with internal geometrical information.

ShapeCBR has been developed at the University of Greenwich and can automatically decompose shapes into disconnected components. An example shape in Figure 3.8 shows this process. The task for shape decomposition is to generate new internal and external information about the shape example and this information involves component identification. More details are given in chapter four shape classifications. The automating shape decomposition product can be listed as follows:

No. of original lines: 6, No. Of projected lines: 2, No. of constructed lines: 2.
No. of original (including original) points: 6, No. of projected points: 2.
No. of constructed points: 2, No. Of hotspots: 1, No. of Component: 1 type L-Component.
No. of components: 1 type L-Region = L-Core-bar = 2 Connector bars type (a).

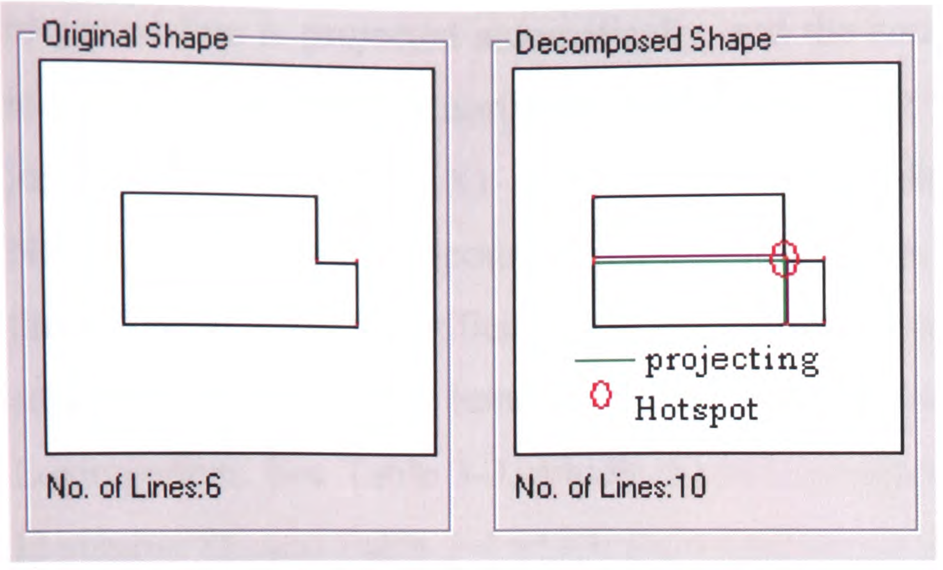


Fig.3.9 illustrates three projecting lines (green) and two hotspots.

The next section presents a number of experiment processes to provide the information detail of the shape decomposition, and at the same time to prove the idea of the decomposition process.

The test is illustrated by examples. All shape examples have been drawn through CAD application packages and sliced into dissimilar views, shape decomposed automatically into rectangular components through ShapeCBR system. Section 3.7 shows experiments for the decomposition process and illustrates by shape examples, the shape automatically generated through the decomposition algorithm.

3.7 The Decomposition Experiments

In this section, the proposed part of the decomposition algorithm is tested on 100 2D cross-sections or views, which represent a 3D complex model. A computer is using the AutoCAD application to create models and slice them. An Example shape is shown in Figures 3.8 and 3.9. View 1 of Mug (of Figure 3.9) made of Bars, L-junctions and T-Junctions.

An algorithm was devised to provide for the automatic decomposition of shapes into the generic components used in this research. This algorithm starts by projecting each vertex (Hotspot) to any side that is directly opposite to it. This provides a decomposition of the area of the shape into a set of rectangles and triangles. Then reconciled and merged if sides defined by internal points only connect them. A set of rules then identifies each element as one of the generic components needed for the componentisation of the shape. Finally, the projected line is projected automatically, and the constructed line drawn themselves parallel to vertical projected and horizontal projected, then it "stems" where appropriate (typically to joins, such as L, T and X). Figure 3.10 and 3.8 shows examples of such decomposition. Notice the top left L component. In the middle figure, the algorithm has identified a rectangle there. The rule that identifies this as an L component on the fact that this rectangle has two adjacent sides (right and bottom) that are internal lines. This identifies the component as an L-component. See Table 3-3, which shows decomposition products from Fig 3.10 for shape Id number 85, and Table 3-4 which shows decomposition products from Fig 3.11 for arbitrary 3D Shapes. These data are generated by the ShapeCBR system for the decomposing process that has been developed for this research.

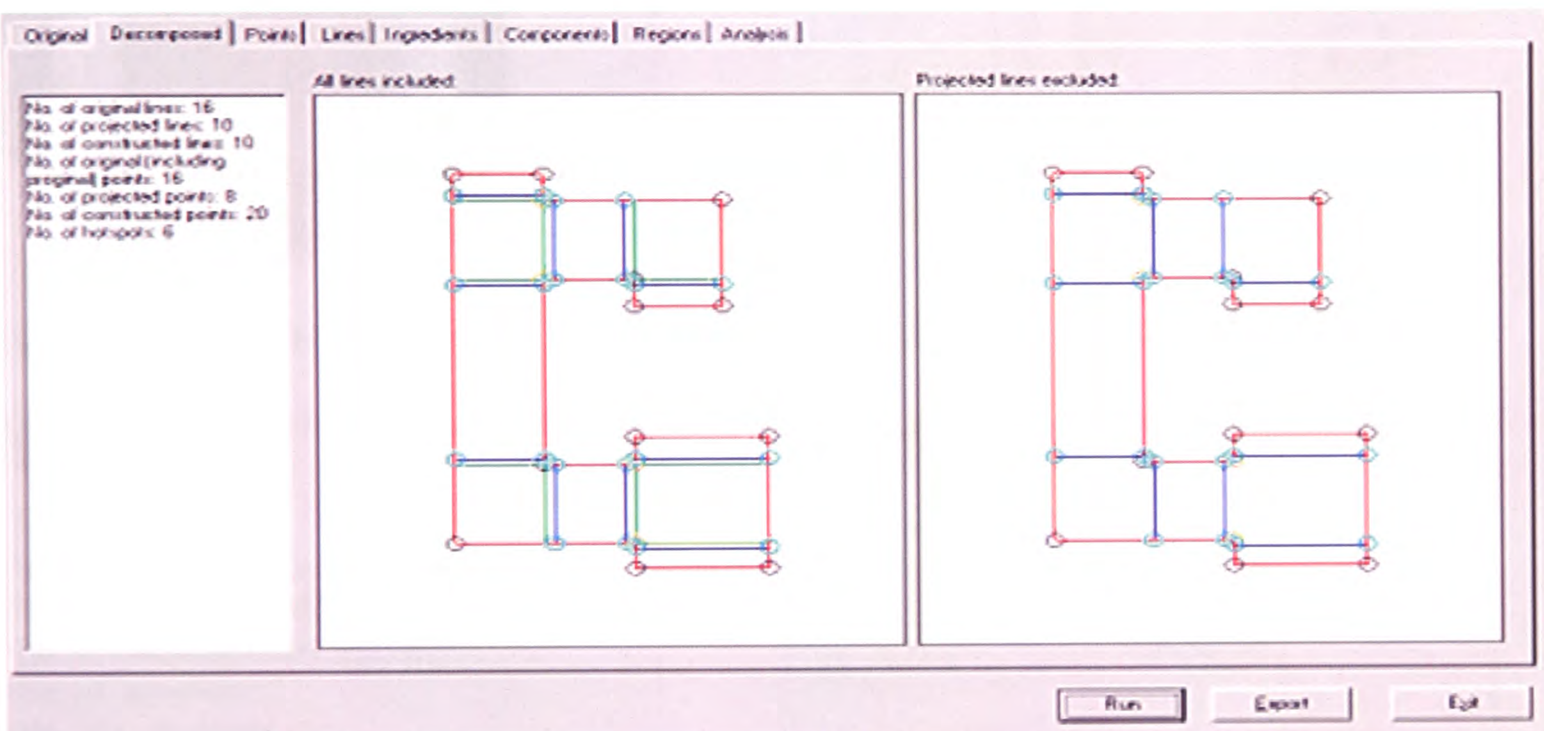


Fig.3.10 Shows an example (1) for Shape Decomposition that is generated by the ShapeCBR System.

The original shape was axisymmetric 3D shape and has been sliced into views which are stored in the case base, ready for the decomposition process. The Figure 3.8 illustrates the process that have been generated by the decomposition algorithm as has been explained above and the geometrical details of the shape shows in Table 3-3. The first column to the left represents the Case ID of the shape, the middle column are the product of points and the last to the right displays the lines type. The projected lines represent constructed lines.

Shapes Name	Points type and numbers	Line type and numbers
Shape ID = 85	No. of constructed lines: 10 No. of original (including proginal) points: 16 No. of projected points: 8 No. of constructed points: 20 No. of hotspots: 6	No. of original lines: 16 No. of projected lines: 10

Table-3.3 shows decomposition products from Fig. 3.10 for shape ID = 85 is one of the 100 cases that have been provided by the pervious research [Mileman: 2000].

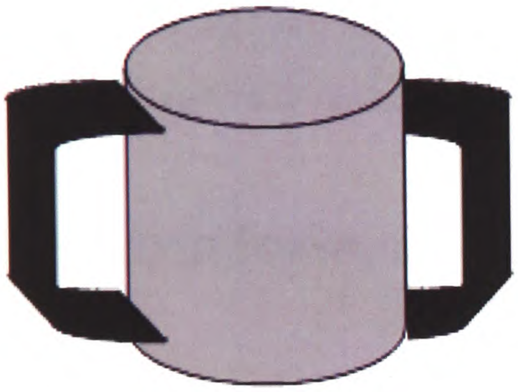


Fig.3.11 Arbitrary 3D Shape (mug).

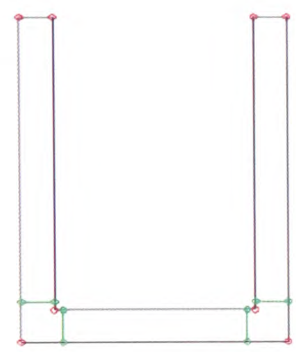


Fig.3.12 view 1 (a).

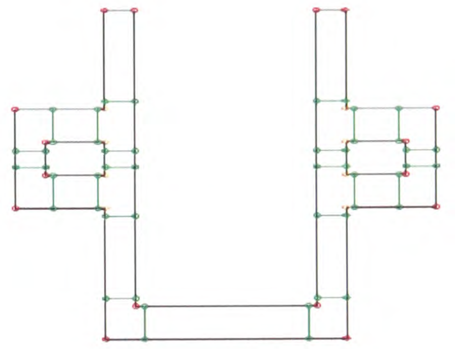


Fig.3.12 view 2(b).

Fig 3.12 view1 (a)	**Fig 3.12 view 2(b)**
No. of elements: 9	No. of elements: 49
No. of connector bars: 3	No. of connector bars: 15
No. of connector bars type A: 1	No. of connector bars type A: 5
No. of connector bars type B: 2	No. of connector bars type B: 10
No. of core-bars: 2	No. of core-bars: 10
No. of L core-bars: 2	No. of L core-bars: 6
No. of T core-bars: 0	No. of T core-bars: 4
No. of X core-bars: 0	No. of X core-bars: 0
No. of Taper core-bars: 0	No. of Taper core-bars: 0
No. of stem bars: 4	No. of stem bars: 24
No. of wings: 0	No. of wings: 0

Table-3.4 shows decomposition products from Fig 3.9 for arbitrary 3D Shape.

The table above shows the products of Figure 3.11. The 3D shape has been sliced into different views as you see in the above figures. Table-3.4 shows the internal geometry that have been generated through the decomposition algorithm. The first column to the left presents the geometrical information for the Fig. 3.12 View 2 (a) and the next column to the right presents the geometrical information for the Fig.3.12 View 1 (b). The two views represent the 3D shape that has been illustrated in Figure 3.11 These two columns show all internal geometrical details that have been generated through the decomposition algorithm. The next section presents an evaluation of the decomposition algorithm.

3.7.1 Evaluation of the decomposition algorithm

Mileman used 100 cases for evaluation of his research. These were manually decomposed by him and evaluated against a casting domain expert. For this research, the same 100 cases were fed into Case CBR and the resulting decomposition was compared to the manual decomposition that Mileman conducted. The result was that in all of the 100 cases, the ShapeCBR decomposition is identical to the Mileman manual decomposition. The next section presents conclusions of this shape decomposition chapter.

3.8 Conclusions

This chapter proposes a method to decompose shapes into separate parts, based on horizontal and vertical projecting techniques. The Framework is to decompose objects, represented as a number of 2D cross-sectional views, which represent 3D shapes. For the decomposition process, algorithms have been designed to perform efficiently with no user involvement. Furthermore, the framework of the application has been implemented to decompose 2D cross-sectional shapes (representing 3D objects) as a demonstration of its effectiveness in shape decomposition. This is demonstrated by examples. The outcome of these decompositions can carry the research to a further step "Classification process". The next chapter deals with shape classification to recognise and classify the decomposition products into identifiable components such as Bar, L, T and X-components. This problem will be investigated in detail in the following chapter 4.

Chapter 4

Shape Classification

Chapter 3 discussed the design of an efficient algorithm to automatically decompose a number of 2D cross-sections or 'views' into generically connected components. The aim of Chapter 4 is to discuss the design of several algorithms that can automatically classify the product of the decomposition process into generically connected and identifiable components. The classification process is based on **Hotspot** identification and **searching methods**: for the classification process using the algorithm known as "Full-scan" , identify the structural components such known as L-component, T-component and X-component; for the classification process using the algorithm known as "Semi-scan" to identify element known as bar and taper component.

4.0 Introduction

The objective of this part of the research is based on the results from the decomposition method discussed in Chapter 3. This Chapter seeks to identify and classify the decomposed shapes produced by the decomposition method into the six generic components (Fig.4.3) identified in previous research [Mileman: 2000 and Biederman et al 1992]. It is then possible to define similarity metrics to assist in efficient shape retrieval containing the relevant casting design knowledge. The final stages of this process will be discussed in Chapter 5. The next section presents the background for the shape classification problem.

4.1 Background to the problem

This section deals with 3D shape classification task for Case-base reasoning (CBR). The majority of experts favour 2D views as a matter of course. For example, civil engineers draw a 3D perspective of a house mainly for customer visualization purposes. But for other details of the house, such as beams, polls, doors, windows, there is a need to breakdown the 3D perspective design into a number of 2D cross-sections or views. The same is true for complex 3D casting designs [Aziz, M.: 2004]. (See the two examples below).

Fig.4.0 (a) represents an axisymmetric 3D casting design. Fig.4.0 (b) represents a cross-section for the casting design. This cross-section demonstrates the internal geometrical structure of the shape. For slicing process a CAD application have been used.

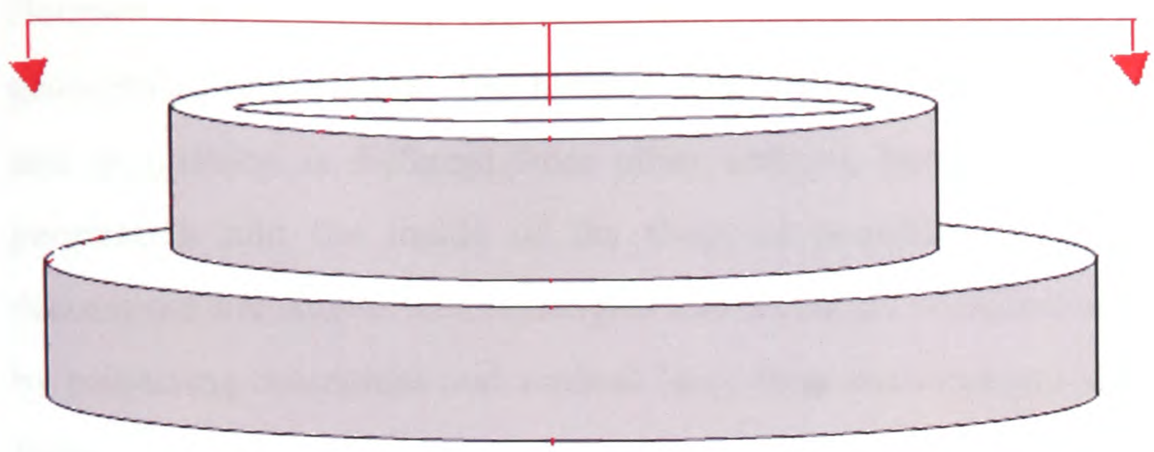


Fig.4.0 (a) shows an example of axisymmetric shape.

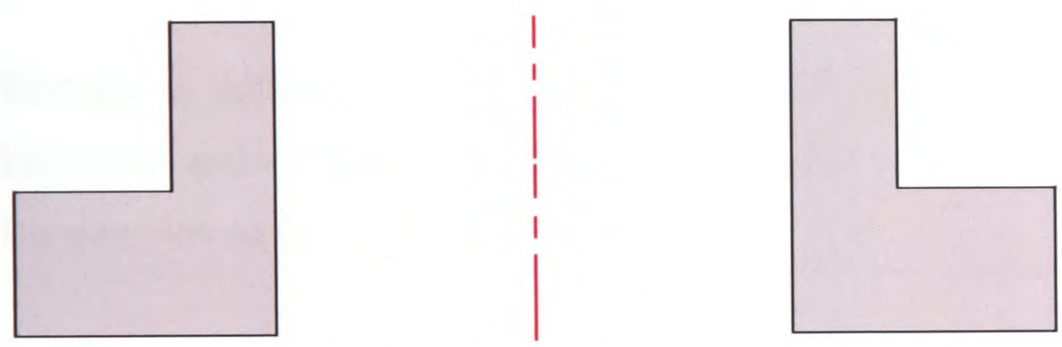


Fig.4.0 (b) illustrates the geometrical information for the above 3D.

As the first step, the decomposition process is a fundamental task in this research. Details of the decomposition process were given in the previous chapter. New generic components (rectangular and triangular) have been defined after applying shape decomposition on the shapes illustrated in Fig.4.1 and Fig.4.5 shows triangular component types. These represent structure of the shape and show internal geometrical information of the shape. This provides a possible solution in casting design for recognising the number of feeders and chills and other details of 3D objects. This chapter attempts to answer the second primary question of this research. Is possible to classify 3D shapes uniquely using generic components? (See on Fig. 4.3).

4.1.1 The classification algorithms

1. The **Hotspot** algorithm:

Hotspot is made up from two connected original lines and this point only concerns internal geometrical information. The hotspot is one of the vertices of the original point of a shape, and its position is different from other vertices, because it is only from these points that penetration into the inside of the shape is possible by using the projecting method to decompose the shapes into rectangles and rectangle primitive elements. This algorithm starts by projecting horizontal and vertical lines from each hotspot to the nearest existing original lines.

2. The **BarSpot** algorithm:

BarSpot is defined as a bar primitive type of component that is created by projecting a horizontal and a vertical line from the first hotspot and each component can only have one Barspot. See on the Figure 4.8.

3. The **Core-bars** algorithm:

Each component has a core-bar and each core-bar has a number of hotspots from one hotspot, which must have up to four Hotspots. Each hotspot in the core-bar represents a type of component.

All these have been discussed in this research. The third **(Core-bars)** algorithm is the most relevant, understandable and efficient because this algorithm has been tested with 100 cases that have been provide by previous research Mileman and tested over ShapeCBR system and compared with the other two algorithms shows better results. Before we discuss the algorithms for classification, the task requires an overview for shape decomposing in the next section.

4.2 An Overview of the Decomposition Method

Figure 4.1 illustrates the decomposition approach analysis. It shows the internal geometrical structure of 2D cross-section shape on the right side of the figure generated algorithm of shape decomposition. The figure shows the, original shape.

The green coloured lines and points show projected lines and the brown coloured lines and points are constructed lines, which are additional lines parallel to each projected line. Black lines are the original or boundary lines.

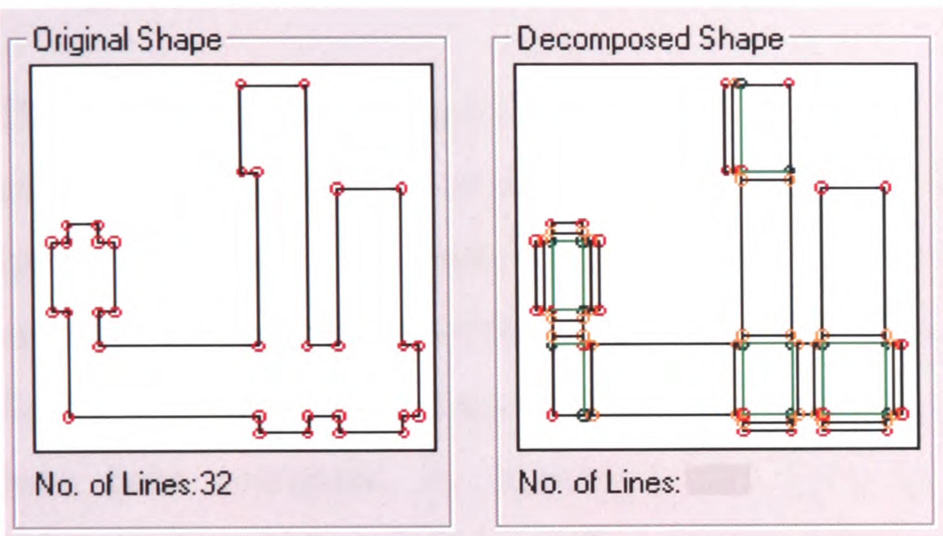


Fig.4.1 illustrates the shape decomposition process.

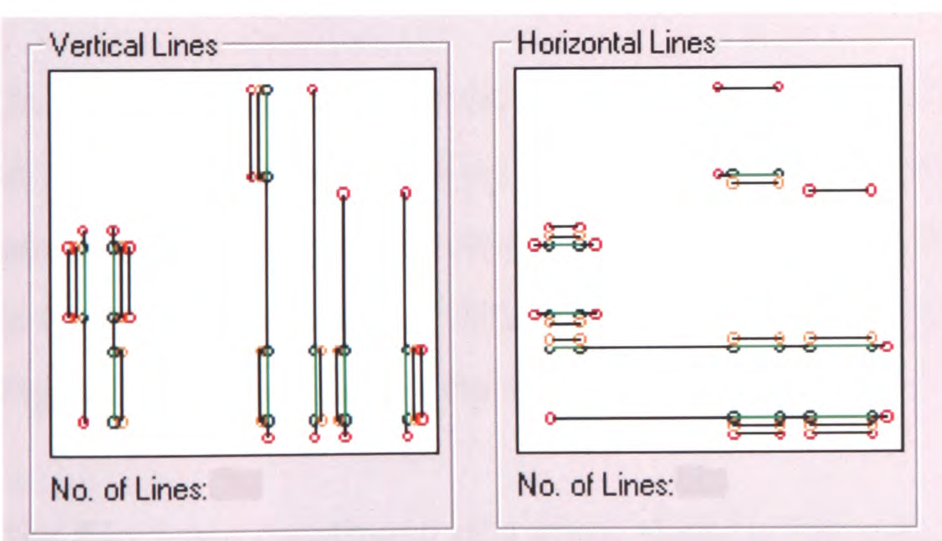


Fig.4.2 illustrates original lines, projecting lines and constructed lines.

Figure 4.2 illustrates all types of lines that have been generated by the decomposition algorithm such as:

- Original lines (Black colour) are the boundary of the shape.

- Projected Lines (Green colour) have been created through by projecting horizontal line and vertical line from each hotspot.
- Projected points (Blue colour) have been crated by projected lines when the projected lines hit opposite side of the nearest wall inside the shape Fig.4.15.
- Constructed lines or stem (Brown colour) have been created during projecting lines operation give a formation of the shape Fig 4.23.

Each colour represents types of line and points and these new lines and points have been generated through the algorithms of decomposition process. The next section discusses the method of the Shape-CBR classification.

Through the above (Fig.4.1 and Fig.4.2) process, the system generates new internal geometrical information, or new attributes such as new lines (projected lines) and points (projected points). This new information could lead the identification of new objects by diagnosing the structures of the shape. Therefore, in this way we are led to finding a solution for the classification of shapes. This is achieved through the way internal lines (structures) have been connected, or structured, and knowing the relationship between the new information and the original details. These questions were raised in the discussion of the decomposition process, and thus needs to be answered.

In order to discuss further the classification approach it needs to define the basic constitution of the shapes for investigating aspects of 3D shapes and 2D cross sections, as well as how we can go in further to break down a shape into basic elementary shapes (that have been shown in Chapter 3) and identify all elements of that shape Fig. 4.19, components Fig. 4.24, regions Fig. 4.25-4.28 and finally the shape Fig. 4.16 itself.

The first basic constituent of a given shape is elements, which represent the basic foundation for components. This component depends on the number of lines. The definitions of our 2D cross-section shapes are composed of elementary 2D objects that call "elements". Elements are either rectangular or right-angled triangles. The next section analyses the products of the shape in detail, along with their definitions, such as the elements, components and regions. These products have been demonstrated by examples and are shown in figures and tables for each product.

The products consist of elements, such as point types, line types, bar types, components, taper types and regions. *However region type will not be covered in this research (future work).* All example shapes within this chapter have been generated automatically from the ShapeCBR system, and their products can be seen in figures and tables in Chapter 4 and Chapter 5.

4.3 Analysis of the research Models

This section is a discussion on the first basic constituent of a given shape which is described in the following sections:

4.3.2 Elements

Fig 4.4 illustrates elements which are a collection of lines. During the decomposition and the classification process, new variant types of lines have been generated through the algorithm, namely constructed lines, projected lines and stem lines.
The following paragraphs describe the elements of the components:

Types of lines: Original lines, projected lines, constructed lines and (stem) segment lines.
The research proposes the following designation for variant types of elements: Table-4.0 below analyses the types of Bar and Wing (see on Fig.4.5) which are the basic constituents of the shape:

Elements	No. of points	No. of Hotspot	No. of original lines	No. of segment lines	No. of Projected lines	No. of constructed lines
Core-bar	4	4	0	0	0	4
Stem bar	4	0	0	2	1	1
Wing	3	0	0		0	0
Connector bar (A)	4	0	0	2	0	2
Connector bar (B)	4	---------	1	3	0	1

Table-4.0 analyses the basic elementary constitution for the shape.

The first columns from the left of Table-4.0 **shows the elements such as:** Core-bar (rectangular), Stem (constructed) bar (rectangular) or bar made up with 4 constructed lines that have been generated during the projection of lines that are meant for recognition of elementary shape. Connector bars for types (A) and (B) are either rectangular and/or wing triangles.

4.3.3 Components

4.3.4 .1 Component definitions

Components are a subset of shapes, which contain many parts. Current research components are three structure components that have different definitions. Table 4-1 shows the products of individual components.

Two perpendicular projected lines build L-Core see Fig 4.4a
Two parallel projected lines build Tp-Core see Fig 4.5 b
One projected line + one original (diagonal) + one segment build Wing see Fig. 4.5 c.
Table-4.2 shows analysis of the component definitions:

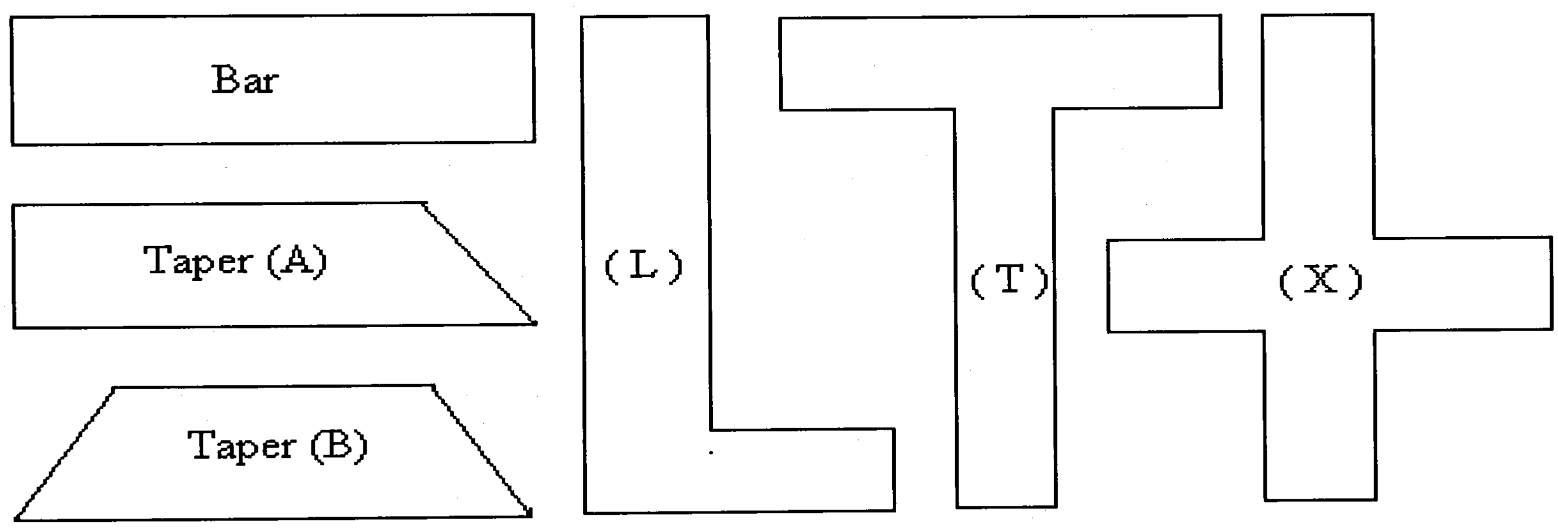


Fig.4.3 shows the proposed primitive components (L, T and X) and Bar and Tapers elements. [Mileman, Thesis: 2000].

Components	No. Of Points	No of Hotspot	No. of original lines	No. of segments	No. of Projected lines	No. of Constructed lines
L-Component	6	1	0	4	0	2
T-Component	8	2	0	5	0	3
X-Component	12	4	0	8	0	4

Table-4.1 analyses the component types.

Table-4.1 shows the analysis of each type of component. The first column shows the component type products, the second shows the points, the third the Hotspot with their numbers, the fourth the original lines, the fifth the segment lines, the sixth the projected lines and the last column shows the constructed lines. All these products in the table above generated through the classification algorithms. Some definitions for the Table 4-2 and definitions of Bar (A) and Bar (B):

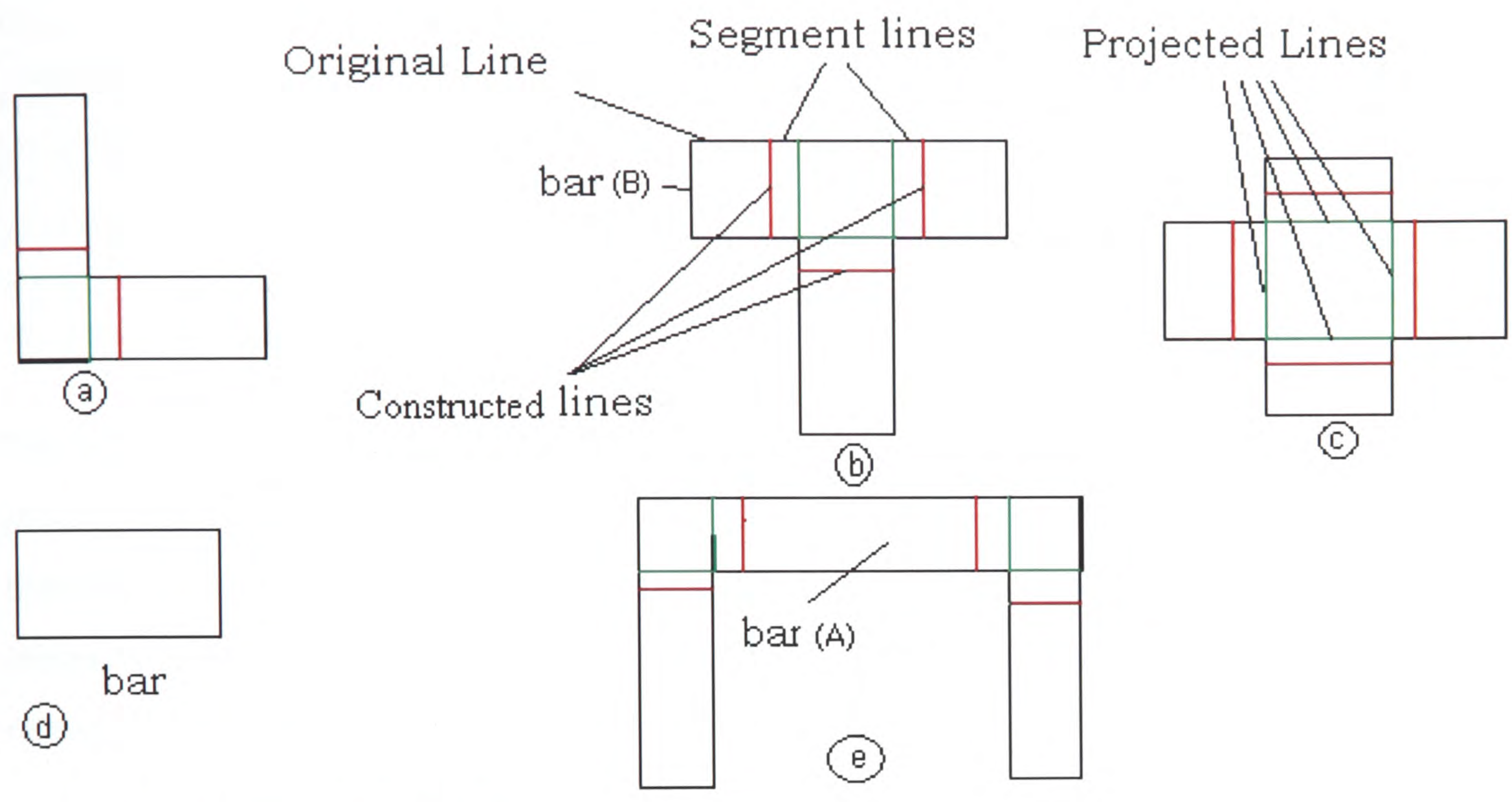


Fig.4.4 shows the research type components.

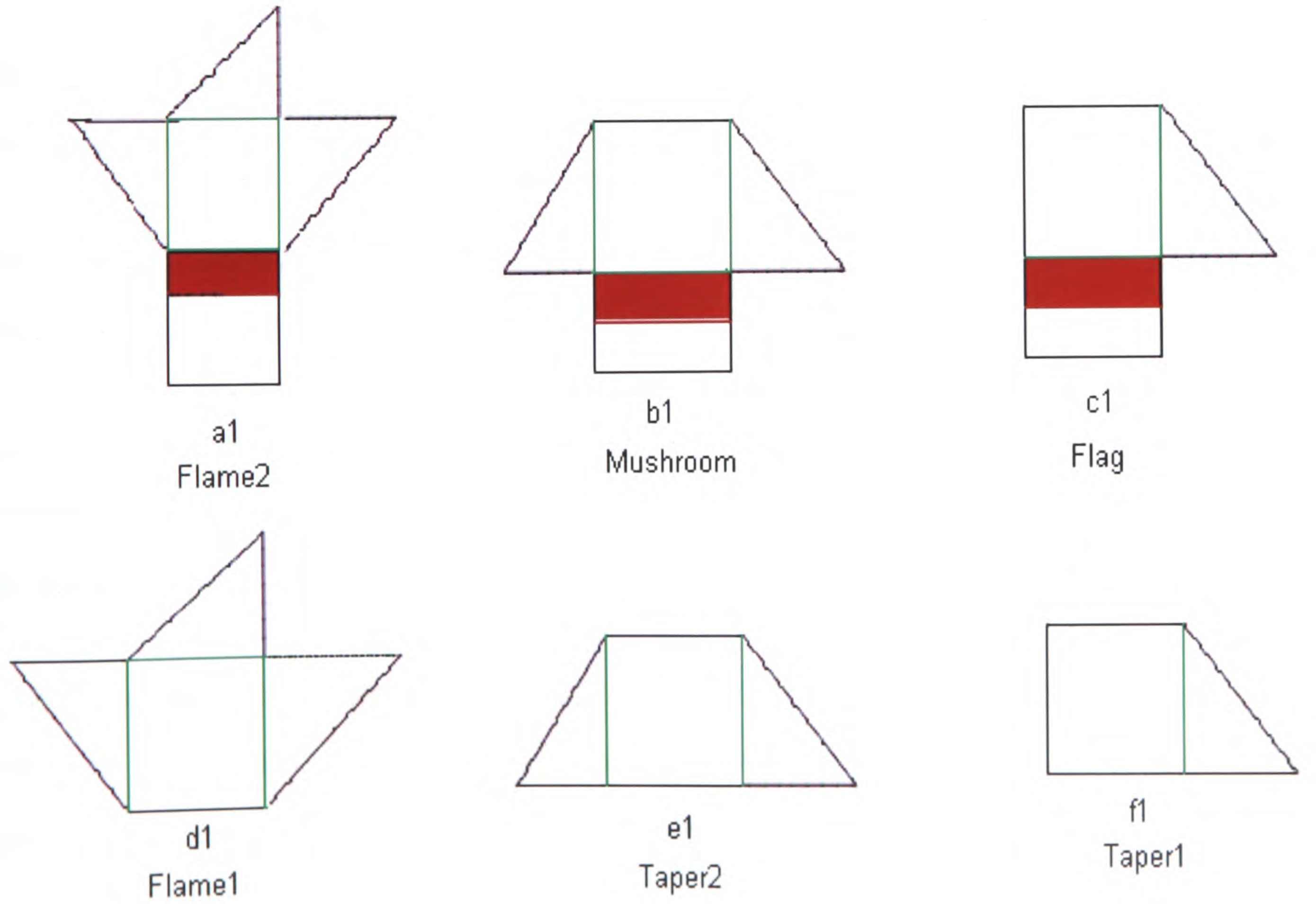


Fig.4.5 shows the different Types of Tapers.

Types of Taper Component	No. of points	No. of Hotspot	No. of original lines	No. of segments	No. of Projected lines	No. of constructed lines	No. of Wings
Tp1-Component	4	1			1	0	1
Tp1-Component	4	2			2	0	2
Flag-Component	6	1			2	1	1
Mushroom-Component	8	2			3	1	2
Flame1-Component	9	4			4	1	3
Flame2-Component	7	2			2	0	3

Table-4.2 shows types of possible Taper Components (in current research only two types have been introduced which are Taper type1 (f1) and Taper type2 (e1).

The above table shows the analyses of tapers types. The first column shows the taper types product, the second shows the points, the third the Hotspot with their numbers, the fourth displays projected lines and the last column shows the numbers of wings and all these products have been generated automatically through the classification algorithms. This research is only dealing with two types of taper (Fig. 4.5 e1 and- f1).

4.4 Matching Algorithms for Classification

This section presents the methodology of shape classification by using the matching technique; we address the shape classification problem in this chapter.

The matching technique is a method to tackle the classification process, which leads towards the final goal; the shape retrieval using (CBR) method.

Several efficient algorithms have been designed for the classification processing automatically identifies individual identifiable component types. This classification process has been designed manually by [Mileman 2000]. The algorithms are:

A: Full-scan: Full scanning is a scanning method where all lines and points are searched, until the first hotspot is reached where a rectangle called a "***Core-bar" created. Each component has a core-bar and each core-bar is made from one or more hotspots to must have up to 4 Hotspots.***

B: Semi scan: semi-scans are used for bar classifications. Semi scan, besides searching forwards and backwards on the same line for Hotspots to inform the previous component of any constructed lines shared between them and, also identifies elements such as bar This technique identifies two types of bars; the first type we called connector bar type (A) which, it links two structural components such as Ls, Ts, Tapers and Xs but the second type we called bar type (B), which has only one link to the structural components. These two algorithms will be discussed in section 4.6.1 and 4.6.2

A lot of the work discussed above has been conducted for the purpose of classifying shapes into generic types of component for various application areas. For the purposes of this research the required shape classification context is that of classifying casting shapes into generic types of component to allow for the re-use of useful casting design knowledge through CBR retrieval of similar shapes based on similarity metrics. Mileman [2000] demonstrated the feasibility of this approach, but he used a manual approach to this classification. However, in order to produce useful CBR based casting design tools, it will be important to automate the classification process. This will make the creation of new target (query) cases and maintenance of the case base more usable and efficient.

In the next section the second approach in the thesis called "classification method" is introduced and discussed. There are three algorithms associated with this method. Only the third algorithm has been implemented in the ShapeCBR system, the other two have been discussed to show that there are many way classify shapes.

4.5 The Algorithms for Classification- Process

In this research, three different algorithms have been designed, the reason being to demonstrate that there are many solutions for the particular problem. All three algorithms are primarily based on the identification of Hotspot for shapes, which are explained in chapter three.

Additionally, the steps for each algorithm have been explained in both theory and practice, through diagrams. But only the final one of these three has been implemented for the ShapeCBR system, as it is the most relevant and efficient algorithm that aids the research to final goal; shape retrieval using (CBR). The algorithms are introduced and explained in detail in sections 4.5.1, 4.5.2 and 4.5.3

4.5.1 The First Algorithm

This algorithm was the first suggestion by the author that was tried. The author thought it is relevant to add this to the ShapeCBR system, particularly for the shape classification process that identified component types and the attempt to answer the second (componentisation) question of this research.

The first step for this algorithm is primarily based on the identification of Hotspots for shapes. The hotspot process has already been explained in detail in Chapter3-p.47, with Figure 3.7 displaying a flow chart for the shape decomposition (hotspot algorithm). The type of components that have been suggested by [Mileman: 2000] and it can be identified through the *number of Hotspots*. For example, bar-components are made up of one hotspot. As shown in figure 4.6 (a), a bar component is made up of two parallel constructed lines, or one original line parallel to one constructed line. Figure 4.6 (b) shows that L-Components are made up of one vertical projected line adjacent to one horizontal line and connected by one hotspot. Figure 4.6 (c) shows that T-components are made up of one horizontal line, or one vertical line connected to an adjacent horizontal or vertical line, that are connected into two hotspot. Figure 4.6 (d) shows that X-components are made up of two horizontal lines or two vertical lines adjacent to two horizontal lines, or two vertical lines. Also, they are connected by four Hotspots. To conclude, Fig. 4.6 and Fig.4.7 summarises the factors that dictate the types of components, the first one being the number of Hotspot and the second being the type of connection between the neighbour lines. Additionally, the lines are shown to be either original or constructed lines.

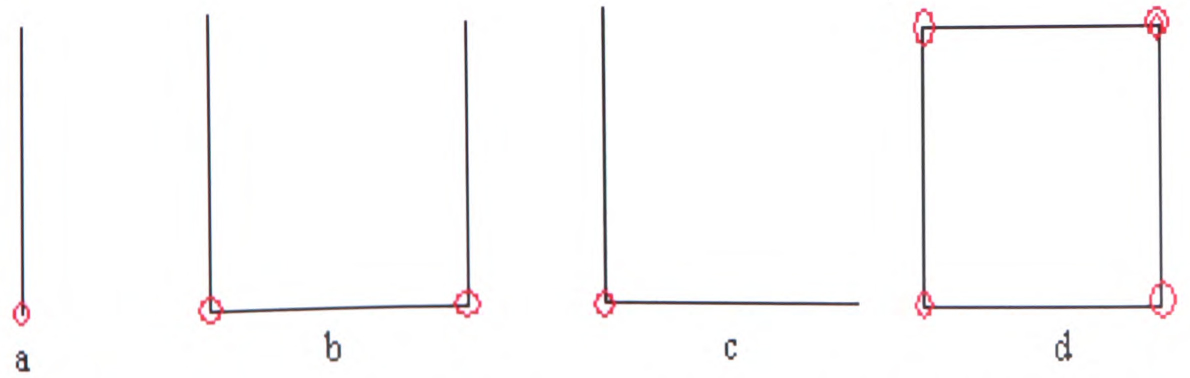


Fig.4.6 shows analysis types of components by the hotspot and their adjacent lines.

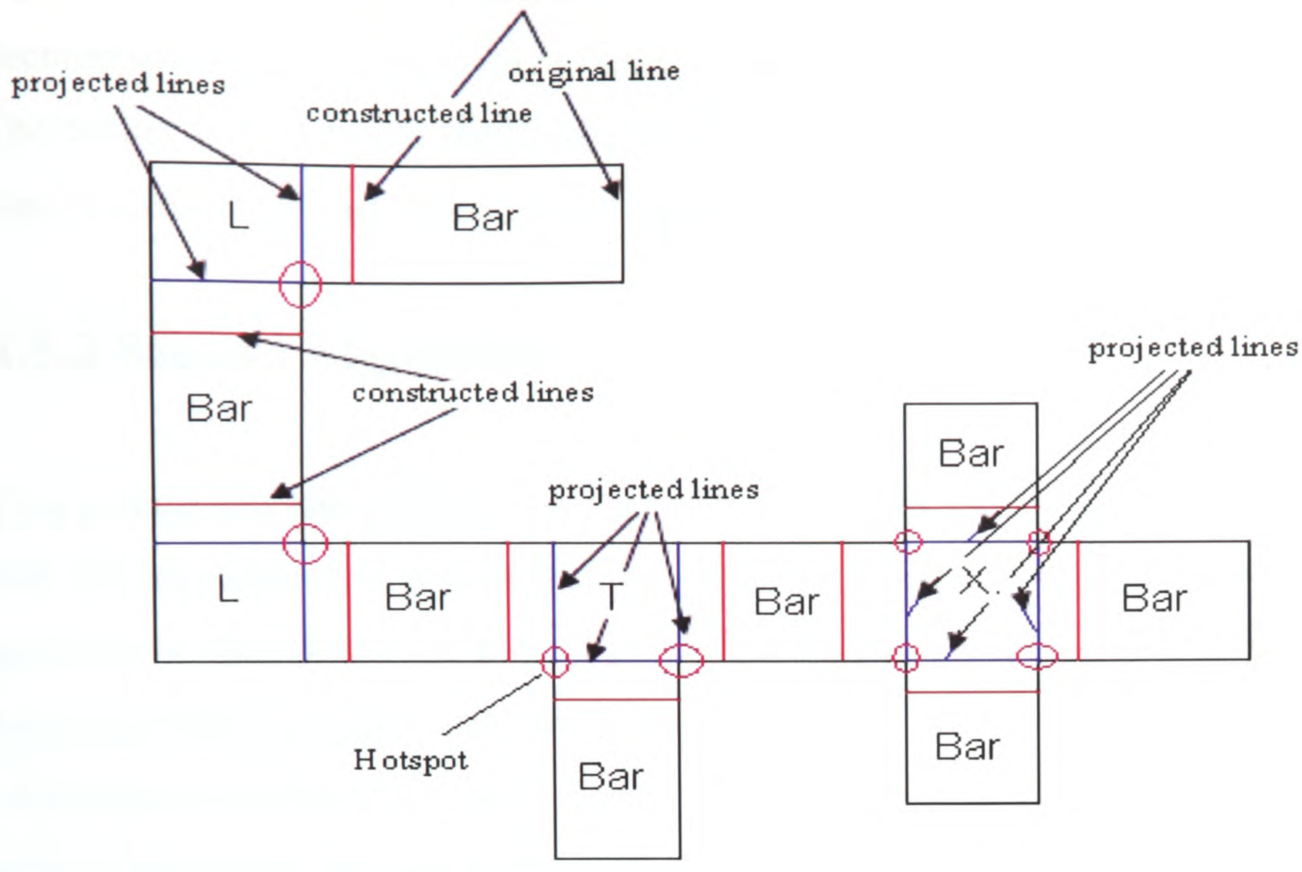


Fig.4.7 shows component types for the first algorithm.

The steps for the first Algorithm

- Retrieve decomposed case (shape) from the case base.
- Read all points from collection points class for the shape.
- Read all lines from collection lines class for the shape.
- Call the hotspots of the shape from collection points.
- Sort all lines.
- If there are two parallel constructed lines or one original line parallel to one constructed line and connected to one hotspot then it is a **Bar.**
- if one vertical projected line adjacent to one horizontal line and connected by one hotspot then it is an **L-shaped.**
- if one horizontal line or one vertical line connected to an adjacent horizontal or vertical line, which is connected by two Hotspots then it is an **T-shaped.**
- if two horizontal lines or two vertical lines adjacent to two horizontal lines, or two vertical lines. Also, they are connected by four Hotspots then it is an **X-shaped.**
- End IF. (End of Classification).

In the next section, the second approach is presented, which is designing an efficient algorithm for the shape classification process. The algorithm's task is to classify the decomposition products into identifiable components Fig. 4.3.

The connectivity between the identified components is done though visiting each constructed line. A constructed line connects two adjacent components.

4.5.2 Second Algorithm

This section discusses the second suggested algorithm for the shape classification. The first step for this algorithm is primarily based on the identification of Hotspots for shapes, and secondly on the creation of a first bar which originates from the first spotted hotspot, which has been called by author for simplicity a ***"BarSpot"***.

A BarSpot is defined as a bar primitive type that is created by projecting a horizontal and a vertical lines from the first spotted hotspot see on Figure 4.8 shows B1 as being the BarSpot. It is important to observe that any given component may only have one BarSpot.

Given an angle relative to the BarSpot and a direction, the second step of the algorithm is to start searching for neighbouring bars. Once one has been located, the nearest neighbour technique will be used for their identification.

Figure 4.8 shows an example of an X-component that has been identified by this algorithm. The primitive types are one BarSpot and four matching bars. Other examples include the L-component being made up of three bars, the T-component being made up of four bars, and the X-component being made up of five bars. Section 4.5.3 introduces the third algorithm.

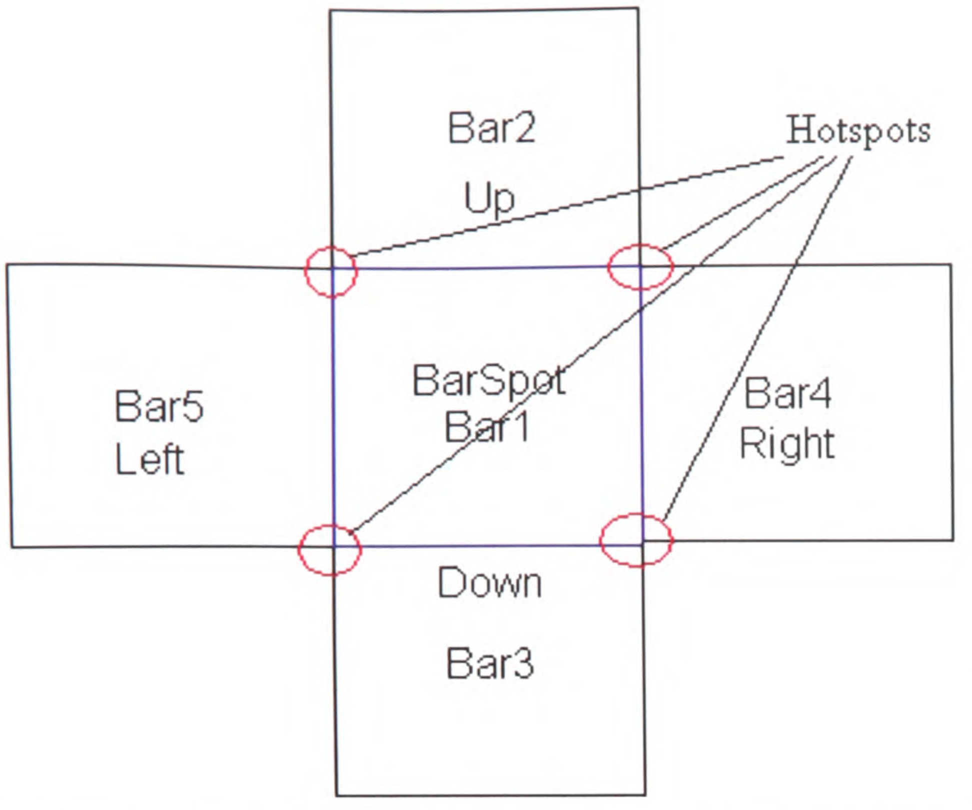


Fig.4.8 shows classification method by using the BarSpot as a central point to identify neighbour bars.

The steps for second algorithm

- Retrieve decomposed case (shape) from the case base.
- Read all points from collection points class for the shape.
- Read all lines from collection lines class for the shape.
- Initialise direction of reading the data of the shape (clockwise or anticlockwise)
- Call hotspots of the shape from collection points.
- Sort all lines.
- Projecting a horizontal line and a vertical line from the first hotspot then draw Barspot.(Barspot made up of one horizontal line connected to one of the neighbours vertical lines.
- From Barspot search down-right-up-left even diagonally with 45 degree to find neighbours bar if there are no bars then exit.
- Else search from (Barspot) D, R, U and L and Diagonally 45 degree.
- If was two bars found RT, TL, LD and DR then create an L-shaped.
- If three bar found from RDC, RTC, DRT and DCT then create an T-shaped.
- If four bar found RTLD, TLDR< LDRT and DRTL then create an X-shaped.

End IF. (End of Classification)

(Note: (U) represents up, (D) represents Down, (R) represents Right and (L) represents Left).

4.5.3 Third Algorithm

This algorithm is the one that has been applied in this research on the classification process to identifying component types such as bars, Ls, Ts, Xs and tapers.

In Chapter 3 the decomposition process has been solved by projecting horizontal and vertical lines from each ****hotspot*** point in four different directions (Up-Right, Up-Left, Down-Right and Down-Left) see Table-3.2. These projections line "hit" the opposite side in the shape. The projection process generates a new point inside the shape, termed a projected point.

**The hotspot* (represents vertices/original points) *is a critical point for both the decomposition process and the classification process.* Figure 4.9 shows the hotspot and projected lines. The projecting horizontal and vertical lines start from the hotspot and these two lines decomposed the shape into rectangular-shaped.

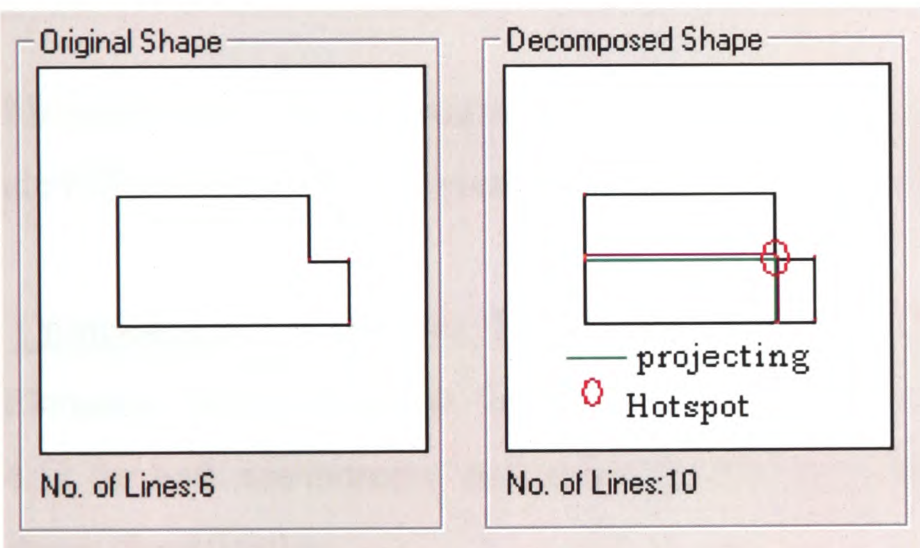


Fig.4.9 illustrates a hotspot and projecting lines.

The next section discusses the classification methods by using two different techniques and gives details of the decomposition products.

The Elements and the components for the Classification

The classification is made up of types of points and lines that have been generated through the decomposing algorithms. These products are:

(i) Types of point: Original points, Projected points, constructed points, Hotspot and proginal points (Original point + hotspot = *Proginal points).

*Proginal points consist of one of the Hotspot made up from projected points and original points (Fig. 4.15).

(ii) **Types of lines**: Original lines, projected lines, constructed lines and segmented lines (Fig. 4.17 and Fig. 4.18).

(iii) **Elements** are the basic foundation for components and they depend on the number of lines (Fig. 4.19).

(iv) The research is dealing with 3D shapes. And a number of 2D views represents 3D and these views are composed from elementary 2D objects that we call elements.

Elements are either rectangles or right-angled triangles. The author proposes the following identification for variant types of elements that will be used for the classification process:

Components: L-Core- bar, T-Core- bar, X-Core-bar and Tp-Core-bar.
Elements: Wing, Stem-bar, Satellite-bar type (A) and bar type (B) see on figures 4.4, 4.5 and 4.12 for both components and elements. The next section presents the engineering steps for shape classification.

4.6 Classification (Algorithms) Methods

1. Full Scanning Algorithm (FSA)

Full scanning is a scanning method where all lines and points are searched, until the first hotspot is reached where a rectangle called a "***Core-bar" created. Each component has a core-bar and each core-bar is made from one or more hotspots to must have up to 4 Hotspots.*** . Then the "corebar" cycle continues to find for other Hotspots. If one Hotspot is found, then an L-component is created (Fig. 4.12) if two Hotspots are found then there are two possible of type components:

1- Then a T-Component, when there are two perpendicular projected lines and two constructed lines.
2- Else a Tp (T-taper)-Component, when there are two parallel projected lines (Fig.4.10).

When four Hotspots are found, then an X-component is created. See Fig. 4.12 for Full Scanning.

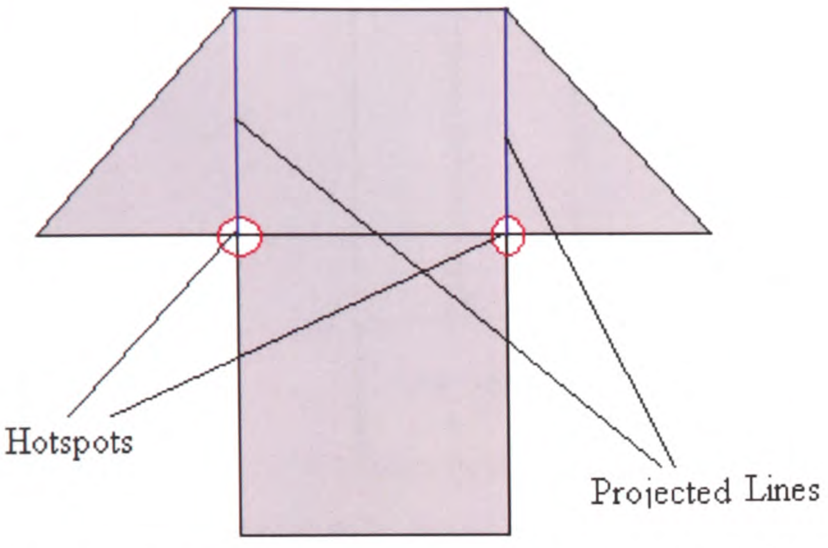


Fig.4.10. Example shows T-taper, the two red collared represent the hotspots and the two blue lines represent projected lines.

How a core-bar draws itself:

This design is called (N) Algorithm to illustrate four possible start points that make a core-bar of components, for all types of core-bars (taking four points at a time that make core-bar). Once the full scan spots the first hotspot then the algorithm immediately draws the core-bar. The (N) Algorithm for core-bar identification has been called because it searches through the points in a path into shape of the letter "N". (See on Fig.4.11).

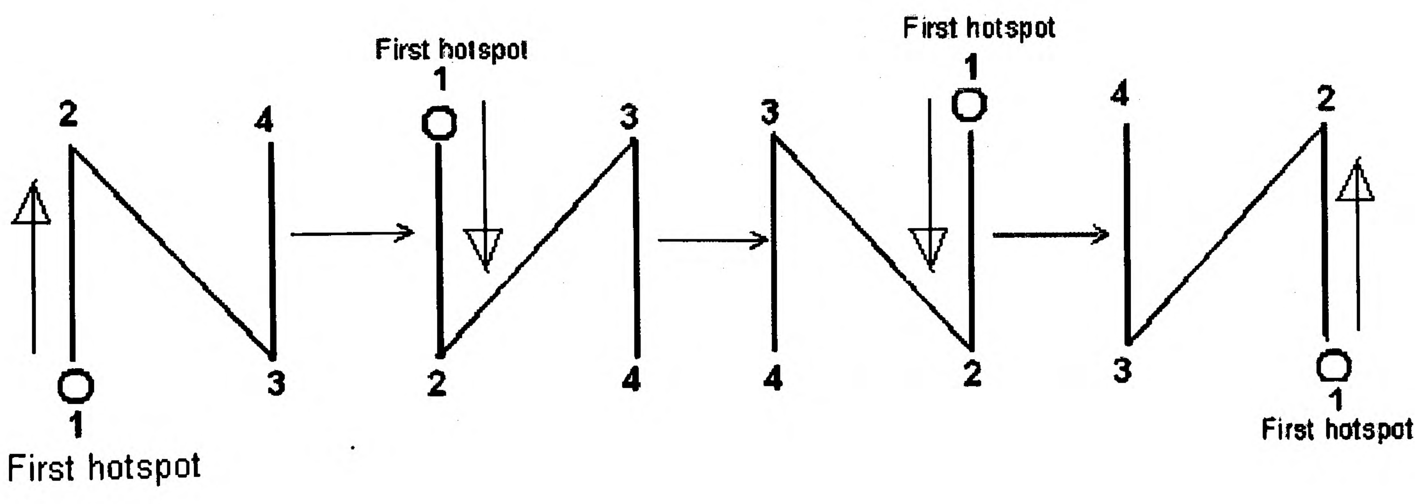


Fig.4.11 shows four possible start points to draw core-bars and the circle shows the first Hotspot that have been spotted by the algorithm.

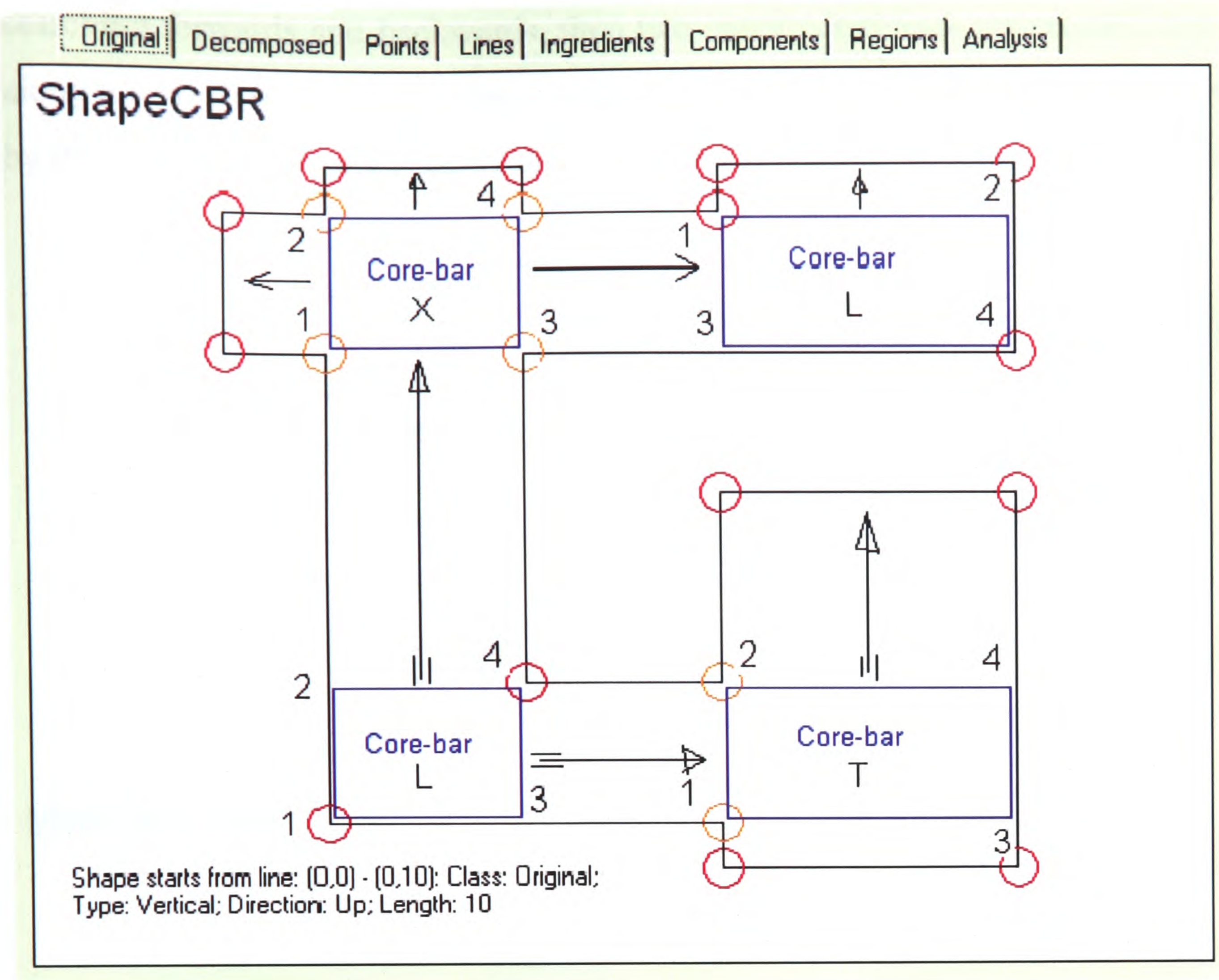


Fig.4.12 shows Full-Scanning method for core-bar types for component.

2. Semi Scanning Algorithm (SSA)

Semi scanning is a second scanning method, which has two operations:

A: Searching all points forwards and backwards on the same line to inform the previous region of any constructed lines shared between them. If there is one then the semi scan does not create a new line because one already exists. Afterwards, the searching continues in other regions until the end.

B: Semi scan: semi-scans are used for bar classifications. Semi scan, besides searching forwards and backwards on the same line for Hotspot, also identifies elements such as bar. This technique identifies two types of bars; the first type we called connector bar type A which, it links two structural components such as Ls, Ts, Tapers and Xs but the second type we called Satellites bar type B, which has only one link to the structural components.

Semi scan technique: if semi-scan spots any hotspot, then a constructed line is created half way between the hotspot and the nearest point and then a bar type (B) will have been identified by the algorithm. But if two adjacent Hotspots on the same line are found by searching forwards and backwards, then two constructed lines are created at points one third along the line connecting the two hotspots and in this case a bar type (A) has been identified by the system Figure 4.13 shows the movement searching for semi scan method.

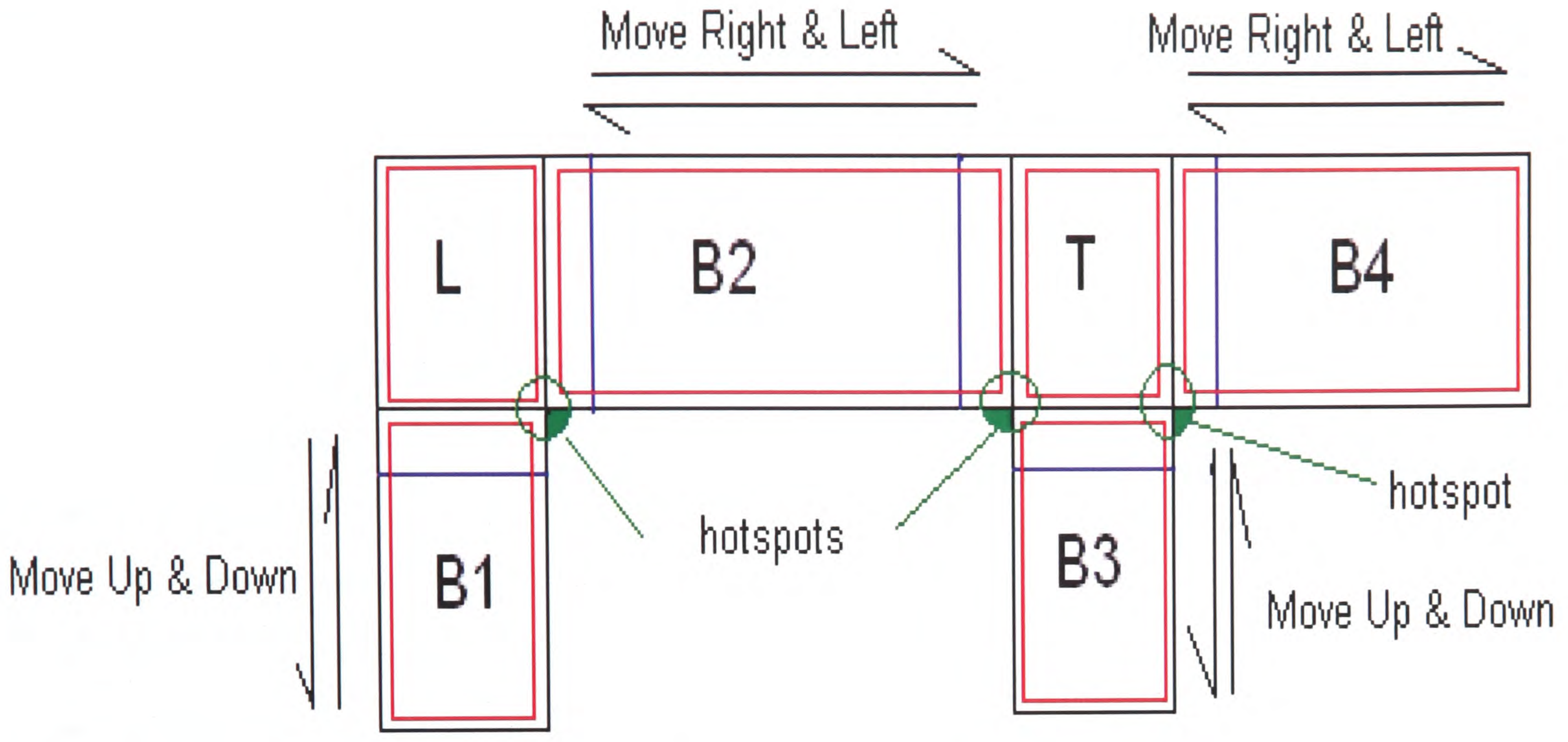


Fig.4.13 shows semi-scanning engineering method.

Fig 4.13 shows an example of such decomposition and classification methods. Notice that the top left is an L-component. The algorithm has decomposed the shape into rectangles and it is very clear that it identified the area as a rectangle. By using a rule such as additional lines parallel to projected lines, this component is identified as an L-component by the fact that this rectangle has two adjacent sides (right and button) and is connected by one hotspot. The two adjacent sides (projected lines) are internal lines and this identifies the L-component by just one hotspot.

The next section reminds the reader about the shape types (such as Bar, L, T and X-components) that have been proposed for this dissertation.

4.7 The primitive Components

Componentisation in this research is based on two types of object. First, structural components (L-components, T-components and X-components) and secondly, joining elements such as bar types and taper types (Fig. 4.14).

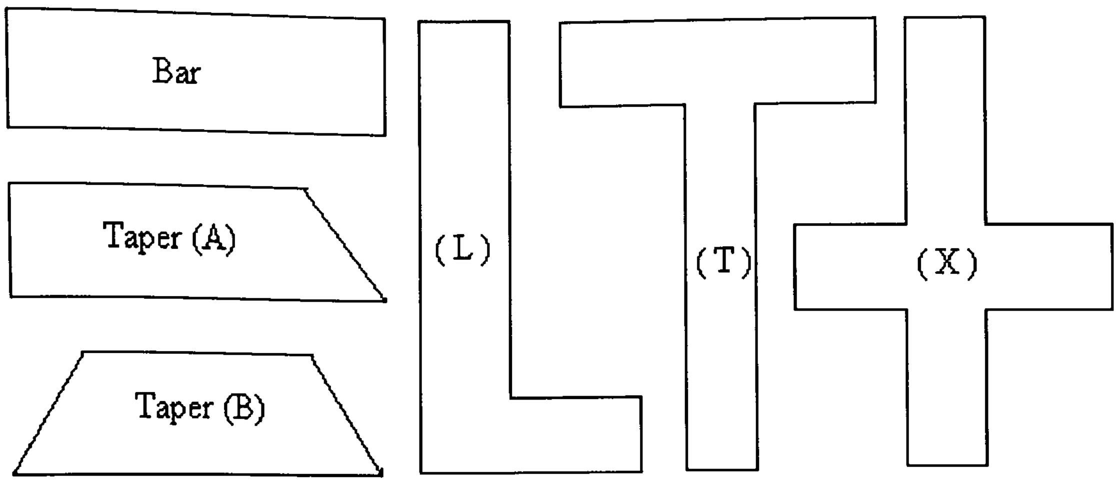


Fig.4.14 shows types of object for current research [Mileman T., thesis: 2000].

Constructions of components are:

- L-Components made up L-Core-bar +2 stem-bars.
- T-Components made up T-Core-bar +3 stem-bars.
- X-Components made up X-Core-bar +4 stem-bars.
- Tp-Components made up Tp-Core-bar + 1-3 wings.
- Components were created by adding additional lines (stems or constructed lines) parallel to projected lines and adding a (stem) weighted value dynamically between projected lines and constructed lines.

The proposed classifications of components into types are: (L, T, X and Tp-component) (Fig. 4.11). The results for this chapter are presented in the next section.

4.8 Classification Experiments

The first test was to give 100 cases to be decomposed into rectangular shapes by using the horizontal and vertical method. The idea of this technique is to find the internal geometric information for the shape, then use the scan approach (Full scan and Semi scan) to classify the rectangular products into identifiable components which have been posed in this research from the very beginning. These tests generated through ShapeCBR system. The results of the testing can be seen on the following sections.

Preliminary tests show that it is possible to automate this process and produce a classification. This algorithm classified the decomposition product into connected types of identifiable components. Figures 4.16-4.28 show the steps for the classification process, which were automatically generated by the ShapeCBR system. Table-4.3 shows the numbers and types of the components that were produced, by both the decomposition and classification algorithms. These products were tested over shape ID number 10, from the cases.

Classification					
Shape_Id 10	Type of the components	Number of components	Type of connections	Number of Hotspot	Number of Nodes
	Bar-Component	4 type (A) and 4 type (B)	1:1 relationships	0 hotspot	40 Nodes. See on Fig. 4.16
	L-Component	0		0 hotspot	
	T-Component	4		8 hotspot	
	X-Component	0		0 hotspot	
	Tp-Component	0		0	

Table-4.3 illustrates shape decomposition and classification.

Figures (4.15- 4.28) illustrate the decomposition and classification processing over an example test for shape Id No.10 (See on figure 4.16). The first step of the process is drawing 3D shapes, the second is to automatically go through slicing into a number of 2D views using the AutoCAD application, the third step automatically decomposes the shape into generic rectangular shapes, and the fourth classifies the decomposition products into well-identifiable components. These products (the new points) are automatically generated by the ShapeCBR system and can be seen in Figure 4.15- 4.28 below step by step.

Fig 4.15 shows the Hotspots and other new points for the shape ID_10, these points have been generated through the decomposition algorithm and classification algorithms; once these Hotspots and other new points have been identified then stores into the collection library for recalling for shape processing.

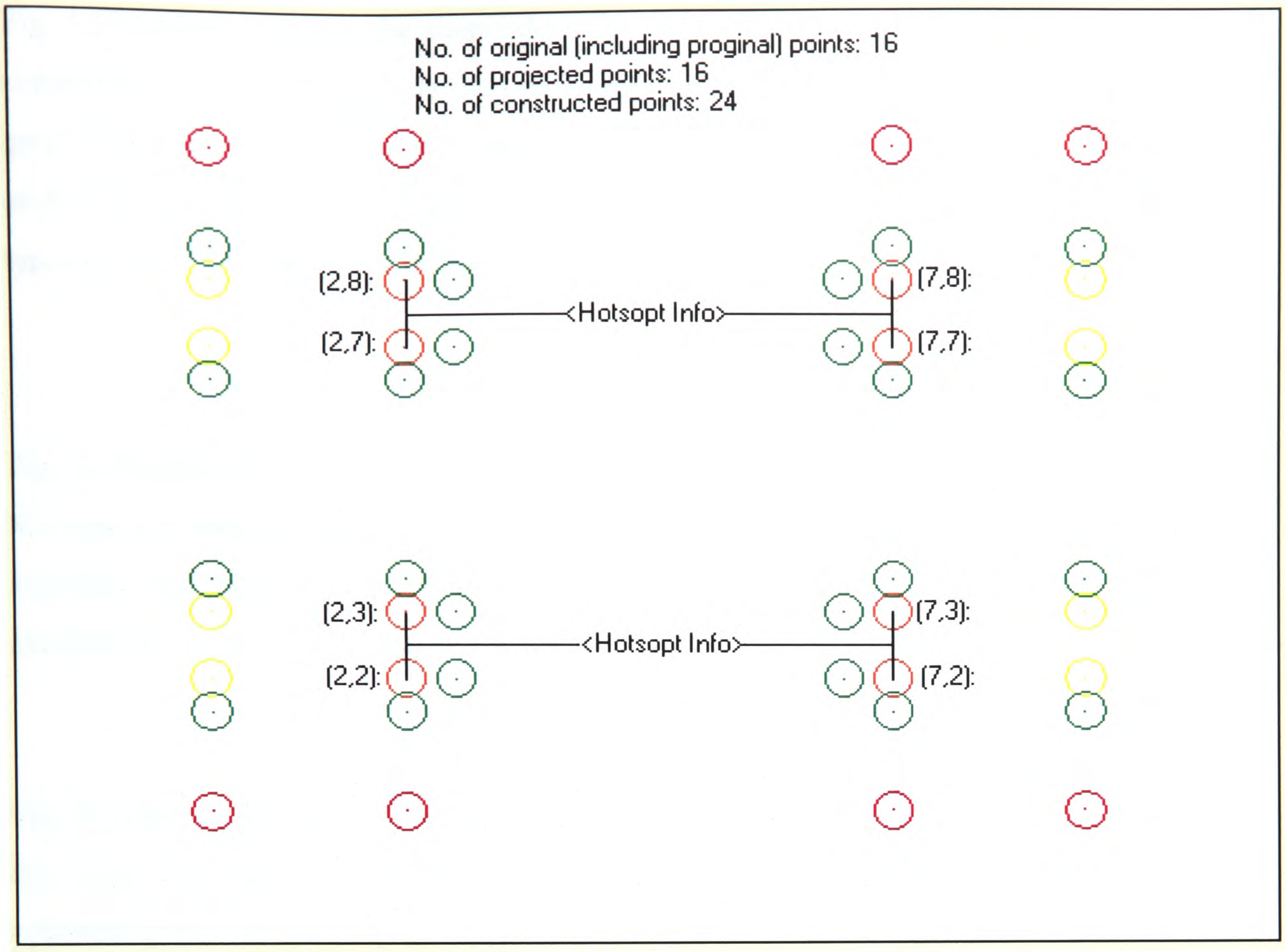


Fig.4.15 the ShapeCBR system shows hotspot and other points analysis for Fig.4.16.

Fig. 4.16 shows the 2D cross- section which have been drawn by ShapeCBR and ready to go through the decomposition process. This view represents shape ID_10.

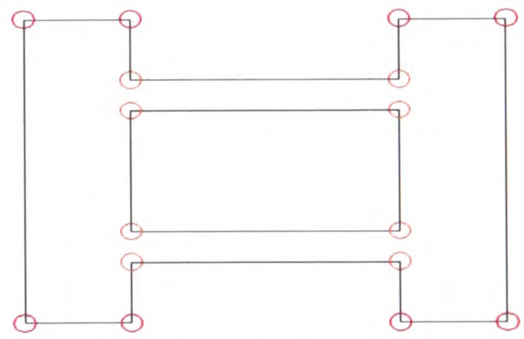


Fig. 4.16 shows 2D cross-section view.

Fig. 4.17 shows the positions of constructed lines and projected lines. These new lines are generated through the algorithm. The constructed lines participate in identifying the component types.

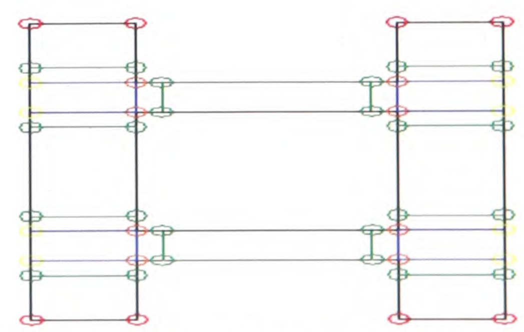


Fig.4.17 shows decomposition processing.

Fig. 4.18 shows that when the shape identifies types of component the system can hide the projecting lines to give a picture of four types of T-shape and between each two T-shapes a type (A) bar can be seen and four types of type (B) as well can be identified.

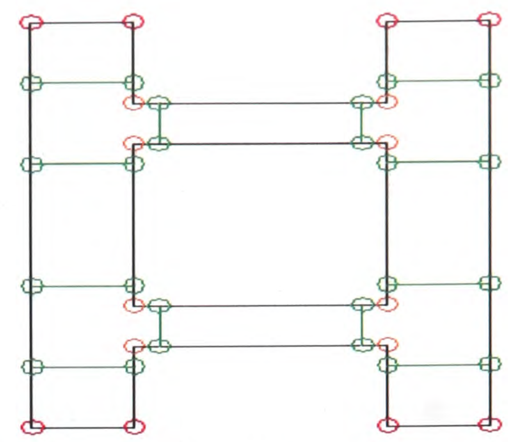


Fig. 4.18 shows classification process.

Fig. 4.19 shows the elements types such as bar types: Bar type (A) has two interfaces. Bar type (B) has one interface and both types are elements. ShapeCBR identified the types of bars automatically.

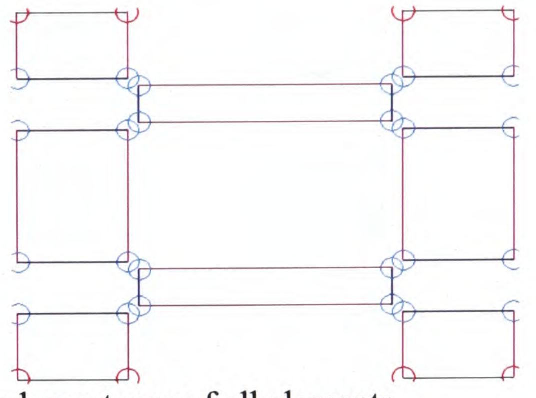


Fig.4.19 shows types of all elements.

Fig. 4.20 shows the elements such as bars of Type (A). Bar type (A) has two interfaces and its role is connecting two structure components.

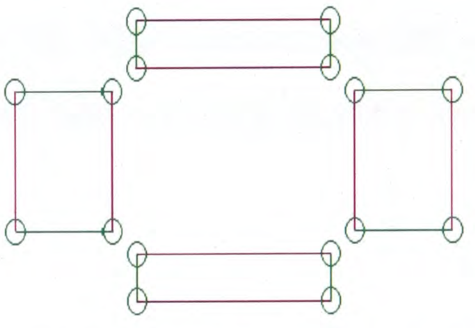


Fig.4.20 shows connector bar Type (A).

Fig. 4.21 shows the elements such as bar types (B).Bar type (B) has one interfaces and their role is different from type (A) but still is a connectors with one face with a structure components.

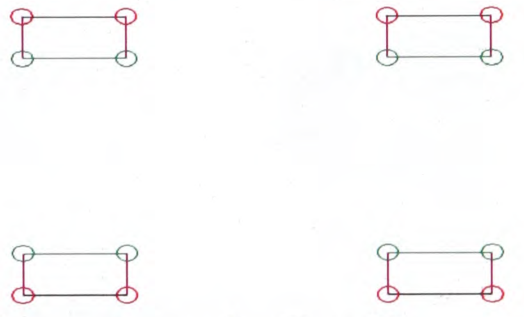


Fig.4.21 shows connector bar Type (B).

Fig. 4.22 shows the core-bars for T-shapes. In the Full Scan section each component has a core-bar and each core-bar has a number of Hotspot from one hotspot, which must have up to four Hotspots. Each hotspot in the core-bar represents a type of component: L-core-bar has one hot spot, T-core-bar has two and X-core-bar has four Hotspots.

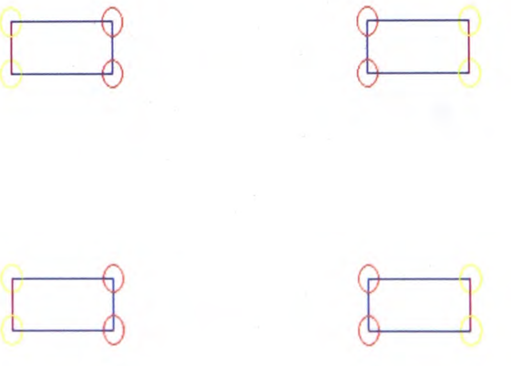


Fig.4.22 shows Stem T-core-bars.

Figure 4.23 shows the stem (constructed) bars. The stem is composed of constructed lines and has been created during the operation of projecting lines.

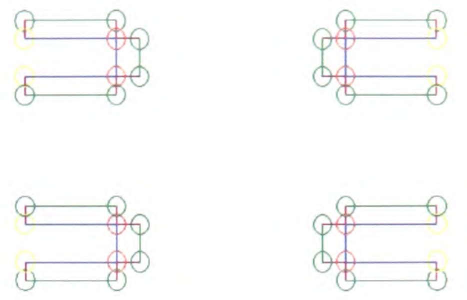


Fig.4.23 shows Stem (constructed)

Fig. 4.24 shows the final output for both the decomposition and classification process by the ShapeCBR system which automatically identified four components of type (T) for current shape example.

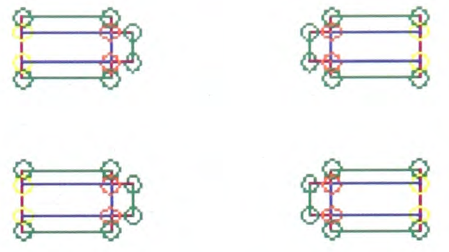


Fig.4.24 shows Type T-Components.

Finally: Figures 4.25, 4.26, 4.27and 4. 28 show the four T-regions, each region being made up from: T-Core and four connector bars of type (A), plus one shared connecting component, which are connector bars of type B. The region shapes are not include for investigation for current research but is a suggestion for future investigation.

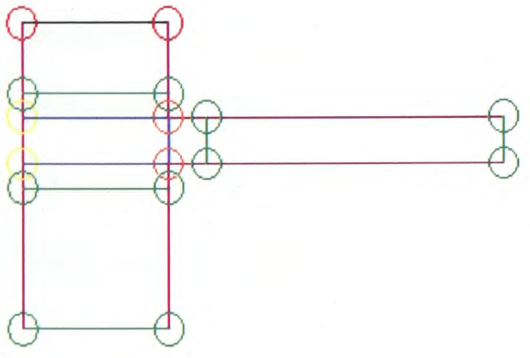


Fig. 4.25 shows Region No.1 for shape 10.

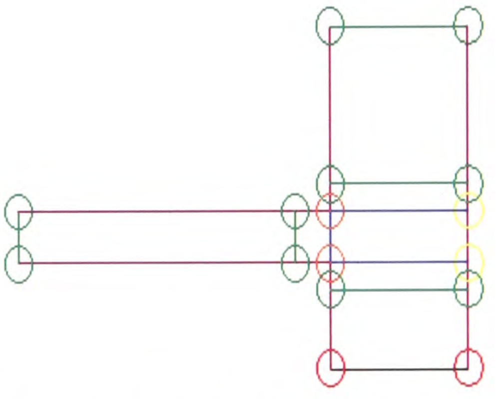


Fig. 4.26 shows Region No.2 for shape 10.

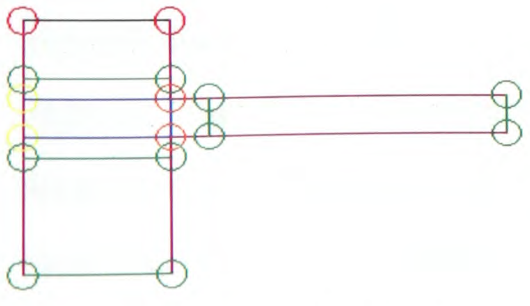


Fig. 4.27 shows Region No. 4.

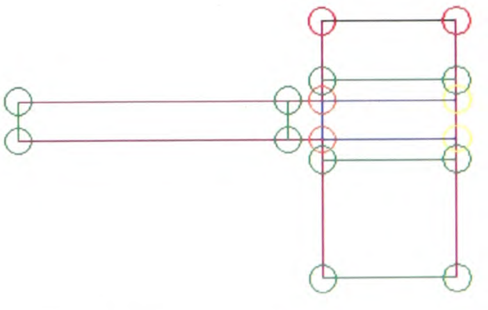


Fig. 4.28 shows Region No. 3.

The preliminary the tests showed that it is possible to automate this process and produce a classification. This algorithm classified the decomposed product into connected types of identifiable components. The above Figures 4.16-4.28 illustrated the steps for the classification process, which have been automatically generated by the ShapeCBR system. Table-4.4 shows the numbers and the types of components that are produced by both the decomposing and classification algorithms. These products were tested over shape one of the 100 cases that have been provided by the experts.

4.9 Evaluation of the classification algorithms

Mileman used 100 cases for evaluation of his research. These were manually classified by him and evaluated against a casting domain expert. For this research, the same 100 cases were fed into Case CBR and the resulting classification was compared to the manual classification that Mileman conducted. The result was that in all of the 100 cases, the ShapeCBR classification is identical to the Mileman's classification [Mileman thesis 2000].

4.10 Conclusion

This Chapter is closely related to Chapter 3. It describes the algorithms used in an operational system for automatic decomposition of shapes into generic components for CBR matching purposes. A number of cross-sections or views represent 3D shape geometry with a subset of disjoined identifiable primitive elements, components and regions. For shape classification process, three algorithms have been suggested, but the third one has been fully developed, used and evaluated.

One of the important points in the shape classification algorithms is to identify Hotspots which already have been discussed in chapters 3. Algorithms have been designed for Hotspot identification. The main use for Hotspot is to define the internal geometry information of the objects. The third algorithm has two parts and has been developed to help classify the decomposed products into types of component. The one is **"Full Scan"**, involving identification of the structure component types such as L, T and X components. The other is **"Semi scan"** identifying the connectors' components such as bar types (A) and (B).

The algorithms developed in this research have been evaluated using a set of shapes used in previous research in CBR for casting design. The procedure of classifying shapes is quite flexible. In general, a shape has many different components. Various processing steps can improve classification of these components as outlined in this research. The next chapter gives a detail study at the similarity metrics is comparing 3D shapes, by designing an efficient algorithm for shape matching. The six metrics have been introduced to increase the efficiency and performance of the similarity metrics for shape retrieval.

Chapter 5

Similarity Metrics (Distance Metrics)

The aim of this chapter is to discuss the similarity metrics for shape matching using graphical representations to achieve the efficient shape retrieval in case-based reasoning (CBR) containing useful contextual casting knowledge. This chapter will attempt to answer the final question of this research " Can a competent similarity metric between 3D casting shapes be defined to allow for retrieval of useful methoding advice associated with 3D shapes? ".

In order to answer this question, several metrics have been posited by previous researcher [Mileman :2000] and a new metric has been created by the author calls ' Component Types Metric' (CTP) to achieve useful, suitable and competent shape retrieval from the case base for given a target shape.

The use of a similarity metric is a measure of the degree of similarity between shape structures expressed as graphs and properties sorting both properties and structure in the order of the case-based knowledge.

5.0 Introduction

Qunni posited that "similarity is the foundation of learning, knowledge and thought, for only our sense of similarity allows us to order things into kinds so that they can equation as an incentive for all kinds of circumstances. Our tendency is to expect that similar causes will have similar effects" [Qunni 1969: 114].

Some researchers in the field of casting design argue that using similarity metrics for shapes retrieval systems for casting design is an example of structure based case of shape retrieval. [Anandan S. and Summers D. J.:2006] Anandan proposed four distinct similarity metrics for shape retrieval in an interactive modeling environment; entity similarity, relationship similarity, attributes similarity, and structural similarity. Gebhardt [Gebhardt, F: 1997] argues that for retrieval systems, features representing complex structures are difficult to define, and similarity must be derived from structure directly. For the sub-class of graphical structures, Gebhart reviews several retrieval systems. These contain group detection as in the Fabel component Topo [Coulon, C.H.:1995], largest common subgraph [Tammer, E.C at el. 1995] and hamming distance [Bunke, and Messmer: 1994].

This chapter focuses mainly on the similarity metrics for the shape retrieval problem by using graph representation. The aim of this is to allow the method to be able to use knowledge from previous design cases by adapting this to be applied to a new target case. *As a result of this retrieval process*, only the case best-suited to the target case is considered for adaptation and the later reuse by the user. For this purpose, an attempt is made to design an efficient algorithm to produce a new metric called ' Component Types Metric' (CTM) to assist the shape matching process and consequently improve the efficiency of the shape retrieval.

The similarity measures are based on features extracted from the structural graphs. Perfect similarity between shapes is obtained when they have identical structural graphs. For graphs that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.

Mileman in his thesis [2000] assumed that similarity metrics for shape retrieval are based on **just the structure** of the shape, but this research is based on the structure and property of the shape to achieve the competent and efficiency shape comparison. The properties and structure (graph) for shape retrieval are very important for improving the quality and efficiency of similarity between two shapes (graphs). In order to discuss the similarity metrics a brief literature review in general about graph and graph matching will be presented within this chapter. A graph representing the shapes is made up of vertices and edges. The vertices represent the structural component types such as L-component, T-component or X-components. The shape (graphs) and the edges also represent connector component types such as bar and taper. The next section a discussion on graph matching is presented.

5.1 Graph Matching

This section is concerned with shape matching using content-based graphs and geometrical information for the shape retrieval process.

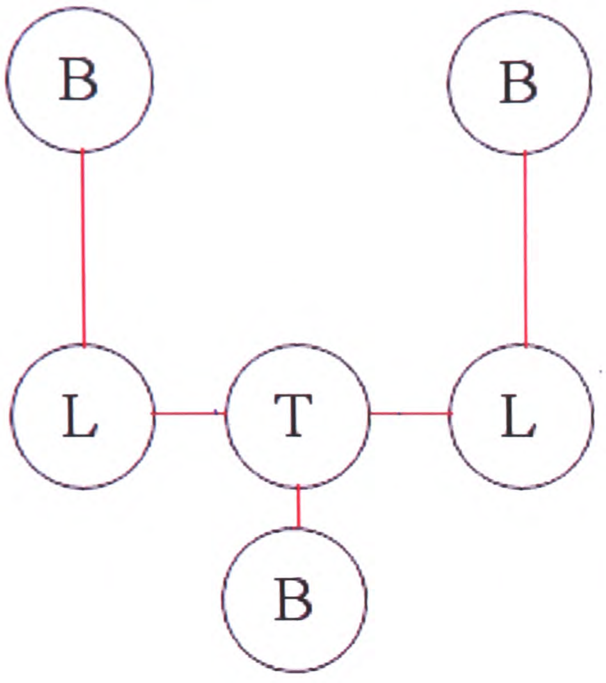
Conceptual graphs have been used to model knowledge representations since their introduction in the early 80's. The formalism of conceptual graphs allowed the introduction of graph matching between pairs of graphs [Sowa, 1984], from either a theoretical or a practical view point, in combination with matching graphs [Eroh and Schultz, 1998], minimal condition subgraphs [Gao and Shah, 1998], finite graphs [Bacik R: 1997], weighted mean of a pair of graphs [Bunke and Gunter, 2001]. Messmer and Bunke present a new graph structure which is better suited for representing parameterised image features. Bunke [Bunke, 2001] Different ways of representing patterns have been analysed in terms of symbolic data structures such as strings, trees, and graphs.

Assume that we have a large number of shapes in the case-based knowledge (database). Given a target shape, can we retrieve a list of shapes from the database, which are closely similar to the query shape? To solve this problem, two aspects are considered. Firstly, the properties of the shape which, represents the geometrical information of the shape and secondly the structure graph of the shape. . A structure graph is a symbolic representation of a 2D-3D shape. The next section presents the notation and terms of graph formulas for graph matching. The next section is discussion on the basic vocabulary for graphs.

5.2 Basic Terminologies for Graphs

Schemer [Schemer et al: 2003] defines a graph $G = (V, E)$ in its basic form as composed of vertices and edges. V is the set of vertices (nodes) and $E \subseteq V \times V$ is the set of edges of graph G. The distinction between a graph G and its set of vertices V is not always made strictly, and sometimes a vertex u is said to be in G when it should be said to be in V. The order (or size) of a graph G is defined as the number of vertices of G and it is represented as (V) and the number of edges as (E) 1. If two vertices in G, say u, v 2 V, are connected by an edge e 2 E, this is denoted by e = (u, v) and the two vertices are said to be adjacent or neighbours. Edges are said to be undirected when they have no direction, and a graph G containing only such types of graphs is called undirected. When all edges have directions and therefore (u, v) and (v, u) can be distinguished, the graph is said to be directed (see on Fig.5.1). *Usually, the term arc is used when the graph is directed*, and the *term edge is used when it is undirected.* In this thesis we will mainly use directed graphs, but graph matching can also be applied to undirected.

In addition, a directed graph G = (V, E) is called complete when there is always an edge (u, u0) 2 E = V × V between any two vertices u, u0 in the graph.



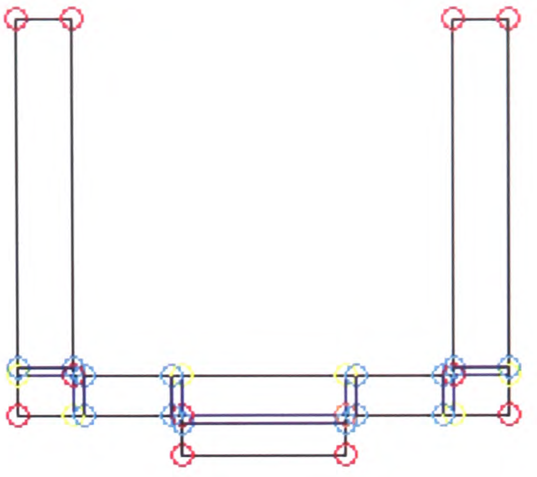


Fig.5.0 shows graphs that represent the 2D shape from the right side of the graph.

Graph vertices and edges can also contain information. When this information is a simple label (i.e. a name or number B (3 bars), L (2 L-components), and T (1 T-component) represents name of components) the graph is called labelled graph (see on Fig.5.1). Other vertices and edges contain some more information. These are called vertex and edge attributes, and the graph is called attributed graph. More usually, this concept is further specified by distinguishing between vertex-attributed (or weighted graphs) and edge-attributed graphs. Next section presents some definition and some classification of graph matching.

5.3 Definition and categorisation of graph matching

Many fields such as computer graphics, computer vision, scene analysis, chemistry and molecular biology have applications in which objects have to be processed and some regions have to be searched for and identified. When this processing is to be performed by a computer automatically without the assistance of a human expert, a useful way of representing the knowledge is by using graphs. Graphs have been proved as an effective way of representing objects [Eshera and Fu, 1986]. When using graphs to represent objects, vertices usually represent features of the object, and edges between them represent the relations (connections) between features or regions. As an example, we can use a graph to represent a shape using the graph shown in Figure 5.0: here all the main physical parts that one expects in a drawing of a shape are shown in the form of vertices in a graph, while edges represent adjacency between the vertices. In this section, we will consider shape matching problems, where the shape is represented as a graph (the Shape graph, SG), and another graph (the data graph, DG) represents the image where recognition has to be performed.

The graph in Figure 5.0 could serve as the shape graph in a graph matching problem. Similar graphs can be used for representing objects or general knowledge, and they can be either directed or undirected. When edges are undirected, they simply indicate the existence of a relation between two vertices. On the other hand, directed edges are used when relations between vertices are considered when not symmetric. Note that the graph in Figure 5.0 is undirected is, and therefore the attributes on each edge are not specified to be vice-versa.

5.4 Related work in Maximum Common subgraph (MCS)

The 'graph distance' between two graphs is defined as the number of modifications that one has to undertake to arrive from one graph to be the other. The distance between two graphs is defined as the weighted sum of the costs of edit operations (insert, delete, and re-label the vertices and edges) to transform one graph to the other. The process of applying these concepts, removing vertices or edges in graphs, is analysed in many works; as removal will lead to smaller graphs the graph matching problem can be reduced in complexity. [Fernandez and Valiente, 2001] proposes a way of representing attributed relational graphs, the maximum common subgraph and the minimum common supergraph of two graphs by means of simple constructions, which allow to obtain the maximum common subgraph from the minimum common supergraph, and vice versa. A distance measure between pairs of circular edges and relations among them is introduced in [Foggia et al., 1999]. This measure is to be applied in domains with high variability in the shape of the visual patterns (i.e. where a structural approach is particularly useful). In [Bunke, 1997] the relation between graph edit distance and the maximum common subgraph is analysed, showing that under a metric equation for MCS graph distance computation is equivalent to solving the maximum common subgraph problem.

5.5 Related Algorithms for Graph matching

Several graph distance techniques rely on finding the "Maximum Common Subgraph" (MCS) [Schenker et al: 2003]. The Maximum Common Subgraph of two graphs is the set of all linked nodes that the two have in common.

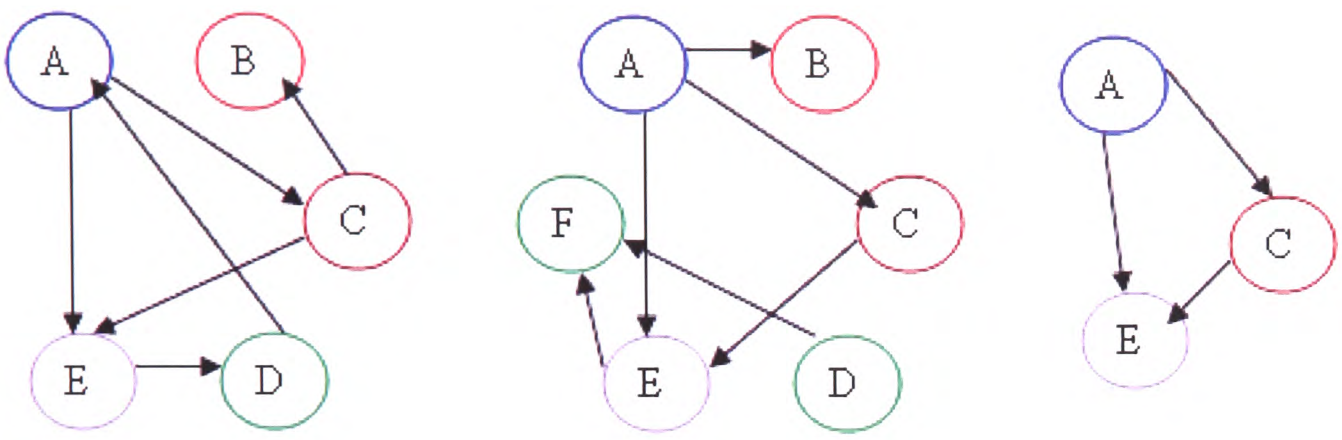


Fig. 5.1 shows an example on MCS (directed graphs).

In Figure 5.1, the nodes of the graphs are labelled (A), (B), (C), etc.; these would be components in the graph representation of a shape. The arrows indicate component order in the original shape. For instance, the shape represented by graph G1 in the figure has at least two occurrences of component (A), one of which is followed by component (C) and the other by component E. Note that components (B) and (D) appear in both graphs, but they are connected differently, so are not part of the MCS. Collectively, distance techniques that use MCS are called MCS-based techniques.

In [Schenker et al: 2003], the authors also refer to one particular distance formula as MCS. In order to distinguish MCS-based techniques from this formula, we refer to the formula as Bunke Largest Graph (BLG). They used BLG [Bunke and Sheare: 1998] and WGU [Wallis et al: 2001], which require finding the maximum common subgraph.

BLG distance is determined by dividing the size (number of vertexes plus number of edges, denoted by (...) in the equations below) of the maximum common subgraph by the size of the larger of the two graphs being compared, and then subtracting the

$$\mathrm{BLG} = d_{BLG}(G1, G2)\sum(1 - \frac{\mathrm{MCS(G1,G2)}}{Max((S1),(S2))})$$

Unlike the BLG distance, WGU (Wallis Graph Union) [Wallis et al: 2001] distance is not sensitive to graphs of disparate sizes. The WGU distance is determined by dividing the size of the maximum common subgraph by the sum of the sizes of the two graphs being considered minus the size of the MCS (so those nodes are not counted twice), and then subtracting the quotient from 5.1, as shown in the next paragraph.

The range for both the BLG and WGU distances is from 0.0 (identical) to 1.0 (MCS is null and the graphs have no nodes in common). For example, referring to the graphs in figure 5.1 above, G1 = 5 vertices + 6 edges = 11; similarly, G2= 12, and
MCS (G1, G2) = 6. The (BLG) distance between G1 and G2 = 1 - (6/12) = 0.5, but the WGU distance WGU [Wallis et al: 2001] is 1 - (6/17) = 0.647.

5.6 Using Graph matching for the shape retrieval

Some researcher such as Sundra [Sundra, H.: 2003], used graph matching as a method for searching and comparing 3D objects. The method encodes the geometric and topological information in the form of a skeletal graph and uses graph matching techniques to match the skeletons and to compare them. The skeletal graphs can be manually annotated to refine or restructure the search.

In this research the similarity measures based on features extracted from the structural graphs are needed for shape retrieval. Perfect similarity between shapes Source and Target is obtained when they have identical structural graphs. However for graphs that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.
For limitation of graph matching is the lack of discernment; large components of objects share the same graphs. For example, bars and tapers have connector role between structural components (such as L, T and X). The connector component in the graphs is sharing information between the neighbouring structural components. The next section is discussion on similarity measurement.

5.7 Similarity Measures

This section describes similarity measures that are based and extracted from the structural graphs and their properties. Perfect similarity between two shapes, for example S1 (represents target shape) and S2 (represents source shape) is obtained when they have identical structural graphs and shape properties. Further for graphs and properties that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.

The similarity measurement in this research deals with 3D shapes for shape retrieval based on structure and shape properties. The structure and shape properties are defined as follows:

- Structure: of the graph is made up of firstly a set of **nodes**, which represent component types such as (L-components, T-components-components and Taper) with their data dimensional information, and secondly, a set of **edges**, represent connector bars and tapers. The connection between the nodes and edges as have been explained in chapter 4 is constructed lines which have been generated through decomposition algorithm. Constructed line role is sharing information between two adjacent nodes

- A property of the shapes represents the features of the shapes and the geometrical information of the shapes.

The next section is a discussion on similarity metric.

5.8 Similarity Metrics (Distance Metrics)

Similarity measures are equations, describing the degree of "likeliness" (or dissimilarity) between two objects (shapes). In other words a similarity measure is a representation of knowledge about patterns, which measure the relative degrees of perceptual similarity between objects in that domain. In this section, several metrics have been created, one of the metrics is new (component types metric has replaced the component number metric, proposed in previous research by [Mileman: 2000]. The reason for using these metrics is to ensure better results in retrieving useful casting information efficiently and automatically from a "similar" existing three-dimensional casting design to a given target shape.

Similarity metrics play an important role in case-based reasoning systems. Wang [Wang and Ishii, 1997] case-based reasoning can be applied to many areas, such as the chemical field, the bio-technical field, the multimedia field, the businesses field and the heavy industries field.

To come up with a reliable and fast algorithm for similarity metrics, requires an understanding of existing similarity theories as well as human expert judgments. An ability to assess similarity lies close to the core of human cognition. Experts use human cognition as well as expert judgment in order to declare or conclusively approve that two shapes (objects) under study are similar.

However there are some important parameters relating to the properties of the shapes in which affect the value of the metric in terms of quality of shape matching in casting designs.

The approach of this research has been to construct a retrieval tool for shape matching or more precisely graph matching. : (ShapeCBR) has been designed to improve the efficiency of the various metrics with respect to different casting design problems. The tool employs a generalised similarity measure $\sigma(S1, S2)$ between shapes S1 and S2, representing a weighted sum of the similarity measures based on different features extracted from the graphs of S1 (target shape) and S2 (Source shape):

$$\sigma\,(S1,S2) = w_{comp\text{-}type}\sigma_{comp\text{-}type} + w_{mcs}\sigma_{mcs} + w_{cycle}\sigma_{cycle} + w_{leaf}\sigma_{leaf} \qquad (5.0)$$

Variation of the weights in this formula allows a general test of retrieval against any given casting design problem.

The individual similarity metrics are defined as follows and the details discussions can be seen in sections 5.7.1 – 5.7.5:

1. **Cycle metric**

$$\sigma_{cycles}(S1,S2) = 1 - \frac{|\text{ncycles(S1) - ncycles(S2)}|}{\max(\text{ncycles(S1)}, \text{ncycles(S2)})} \qquad (5.1)$$

σ_{ncycle} (S1,S2) is based on a count of elementary graph cycles Section 5.7.1.

2. **Component types metric**

σ_{Comptype} (S1, S2) is a measure based on the types of component with their numbers that are common to the two graphs. .

This similarity metric is given by:

$$\sigma_{compType}(S1, S2) = \frac{\sum_{ComponentType} \sigma_i \,(S1, S2)}{noTypes} \tag{5.2}$$

3. **Maximum common Sub-graph (MCS) metric.**

σ_{mcs} (S1,S2) is a measure based on the length of the maximum matching subgraph. If two graphs are nearly identical, σ_{mcs} will also be close to 1.

This similarity metric is given by:

$$\sigma_{mcs}(S1, S2) = \frac{(\text{Length(S)})^{2}}{\text{length(S1)length(S2)}} \tag{5.3}$$

Where S' is the maximal common subgraph of S1 and S2, i.e. the largest graph which is a subgraph of both S1 and S2. The problem of finding S' is related to that of the well-known graph isomorphism problem. For small graphs of up to 10 arcs, a search based on direct comparison of all subgraphs of S1 with those of S2 is possible. *For larger graphs a strategy based on a preliminary comparison of node types and degree can help to reduce the search time.* A detail discussion is on section 5.7.4

4. **Leaves metric σ_{leaves}** is based on a count of leaf nodes, and gives the number of branches to a tree. The Leaf divides into four types:

$$\sigma_{leaves}(S1, S2) = 1 - \frac{|\text{nleaves(S1) - nleaves(S2)}|}{\max(\text{nleaves(S1)}, \text{nleaves(S2)})} \tag{5.4}$$

(a) Bar type (B) represent leaf has one face is connected and the other face is free Fig. 5.5.

(b) Bar type (A) represent have two faces connected to two structural components and it is not leaf see on Fig. 5.5.

(c) Taper type (A) and (B) represented leaf, type (A) is made up of one wing + core-bar-taper and Type (B) is made up of two wings + Core-bar-taper. See on chapter 4 Fig. 4.5. A detail discussion can be seen on Section 5.7.5 and Section 5.7.6.

The individual similarity metrics in are defined as follows in the next page in details such as the cycle metric, the component types metric, the component number metric , the maximum common subgraph (MCS) metric and the leaf metric :

The details discussions on the research metrics

5.8.1 Cycle Metric

Cycles are defined as a number of sets of shapes or loops within a shape. With current data that has been provided, a cycle metric does not seem to be an important metric for similarity, but it is possible that it gives better matching results when the test data deals with more than two or three cycles. However our test data only contains one or two cycles.

Basically, to recognise the number of cycles means allocating value to specify whether it is a cycle or not. If number cycle is not found, then the two shapes are not identical in terms of cycle metrics, *but a percentile value is still given with 0.5*. If just one cycle spotted the shape, and again they are not identical and the calculation will give zero percentage.

Equation for the cycle metric:

$$\sigma_{cycles}(S1,S2) = 1 - \frac{|\text{ncycles(S1) - ncycles(S2)}|}{\text{max(ncycles(S1), ncycles(S2))}} \qquad (5.1)$$

Where max (ncycles (S1), ncycles (S2)) >0,

If a cycle is not found, then the two shapes are not identical in terms of cycles , but a percentile value is still given as 0.5 i.e.

$$\sigma_{cycles}(S1,S2) = 0.5 \qquad (5.1.1)$$

Where max (ncycles (S1), ncycles (S2) =0,

where S1 and S2 represents source shapes.. Section 5.7.2 contains a discussion on the component type metric.

5.8.2 Component Type Metric (CTM)

Chapters 3 and 4 were concerned with the decomposition and classification algorithms. Over 100 shapes have been tested with the automatic componentisation process. The shape is made up of different products or (features) such as elements, components and regions. Some of these features such as properties (geometrical information of the shape) of the shape were not considered in pervious research for shape retrieval [Mileman: 2000].

In this research, each component types have been considered because a component has an attribute with constant numerical value.

The measure of component is based on finding the ratio for each component faces from both target shape (S1) and source shape (S2) and the faces is depended on component types for example: L-components has two faces, R1 and R2, T-component has three faces, R1, R2 and R3. X-component has four faces R1, R2, R3 and R4. Finally bar and taper has two faces, R1 and R2. (See on Fig.5.2). The R1 represents the first face of the component and for the R2 represents the second face.

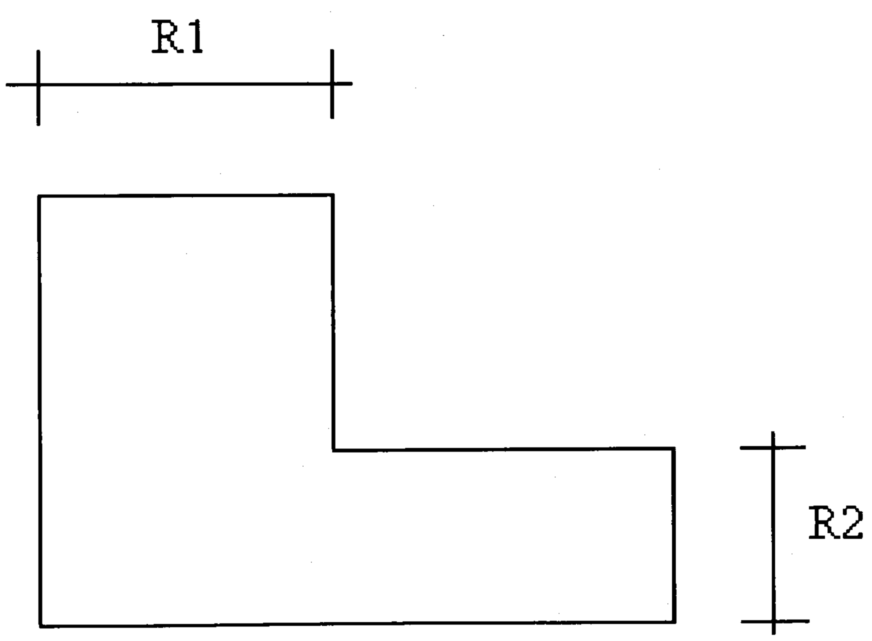


Fig. 5.2 aspect ratio for L-component, R1 represents face1 one and R2 represents face 2.

Equation to calculate an aspect ratio:

$$R_L = \frac{R_1}{R_2}, where, R_1 < R_2$$

This research dealing with two types of primitive components:

- The structure components such as (L-components, T-component and X-component, see on Fig 5.3).
- The connector such as Bars type ((B) and (A)) and Tapers, see on Fig 5.4).

Type (A) connector role is connecting two structure components together. Structure components participate with a minimum of two faces or more But Type (B) connector role is only one face has been connected to structure component and the other face of (B) type is free.

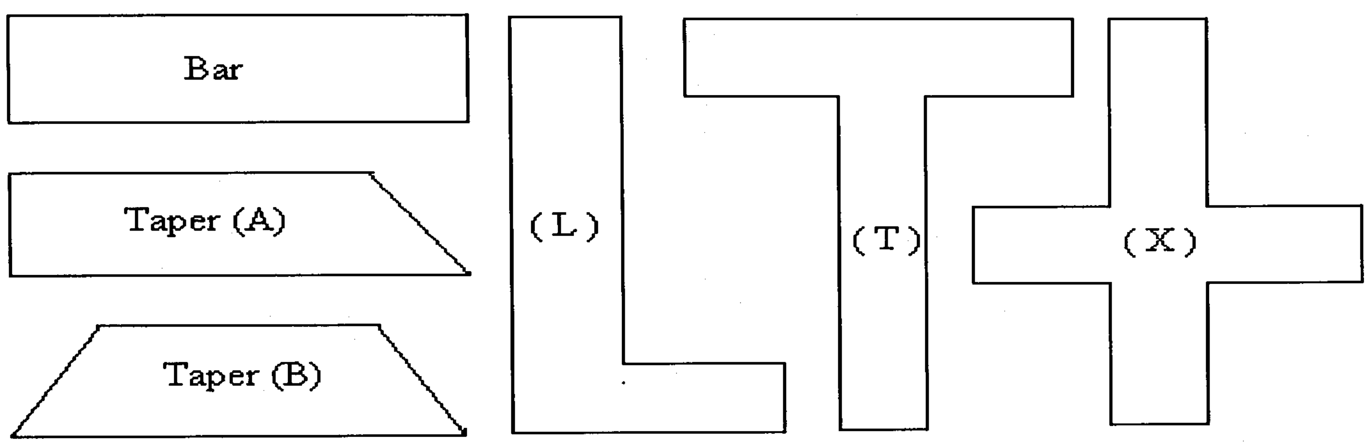


Fig. 5.3 shows component types.

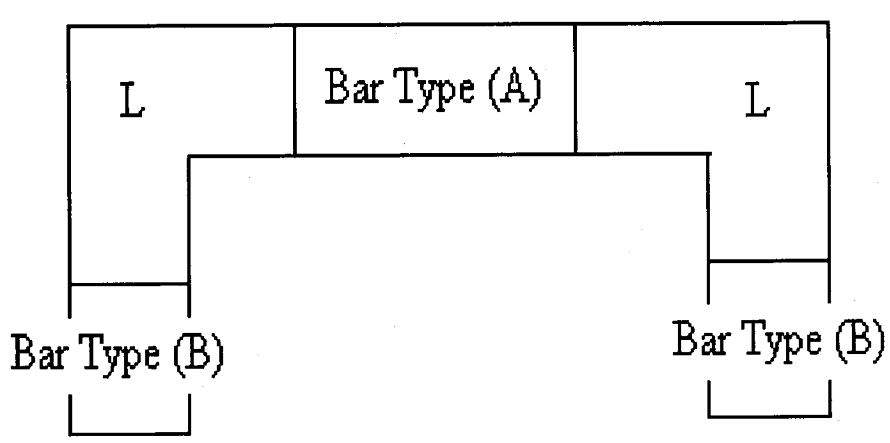


Fig. 5.4 shows bar type components.

In order to refine the metrics used in past research, a new set of metric has been introduced. This is the Component Types Metric (CTM). Previous research assumed that the similarity between two components of the same type was 100%. However, from discussion with casting experts it became apparent that the actual size and the proportions of a particular component are important in determining relevant useful casting knowledge. Therefore, there was an apparent need for determining a similarity metric between components of the same type.

One of the challenging metrics is the CTM. To define it, an equation has been proposed (see below).

The algorithm for component type metric:

For each target component, the first step is to ascertain its geometry. After determining the geometry the second step is, as follows: each target component is compared with all source components to determine their similarity values and calculate the aspect ratio of each component type. For the third step an average is obtained, by adding the similarity metric value for each the source shape comparison and dividing by the number of component types in source shapes..

$$R_L = \frac{R_1}{R_2}, where, R_1 < R_2 \ldots\ldots Equation\ to\ calculate\ an\ aspect\ ratio.$$

An example for L-component types size calculation can be seen see on the table-5.0 and figure 5.5.

L S1 T1	L S2 T1	LS3 T1
L S1 T2	L S2 T2	LS3 T2
L S1 T3	L S2 T3	LS3 T3
L S1 T4	L S2 T4	LS3 T4

$$Sum_{(S1,S2)}\sum\nolimits_{L-Component} \frac{(LS1T \div No.T\arg etComp) + (LS2T \div No.T\arg etComp) + (LS3T \div No.T\arg etComp)}{No.CompTypeSource}$$

Table-5.0 shows the calculations of both Target components and Source components.

The rows are represented the target components (T) and the columns are represented the sources components (S).

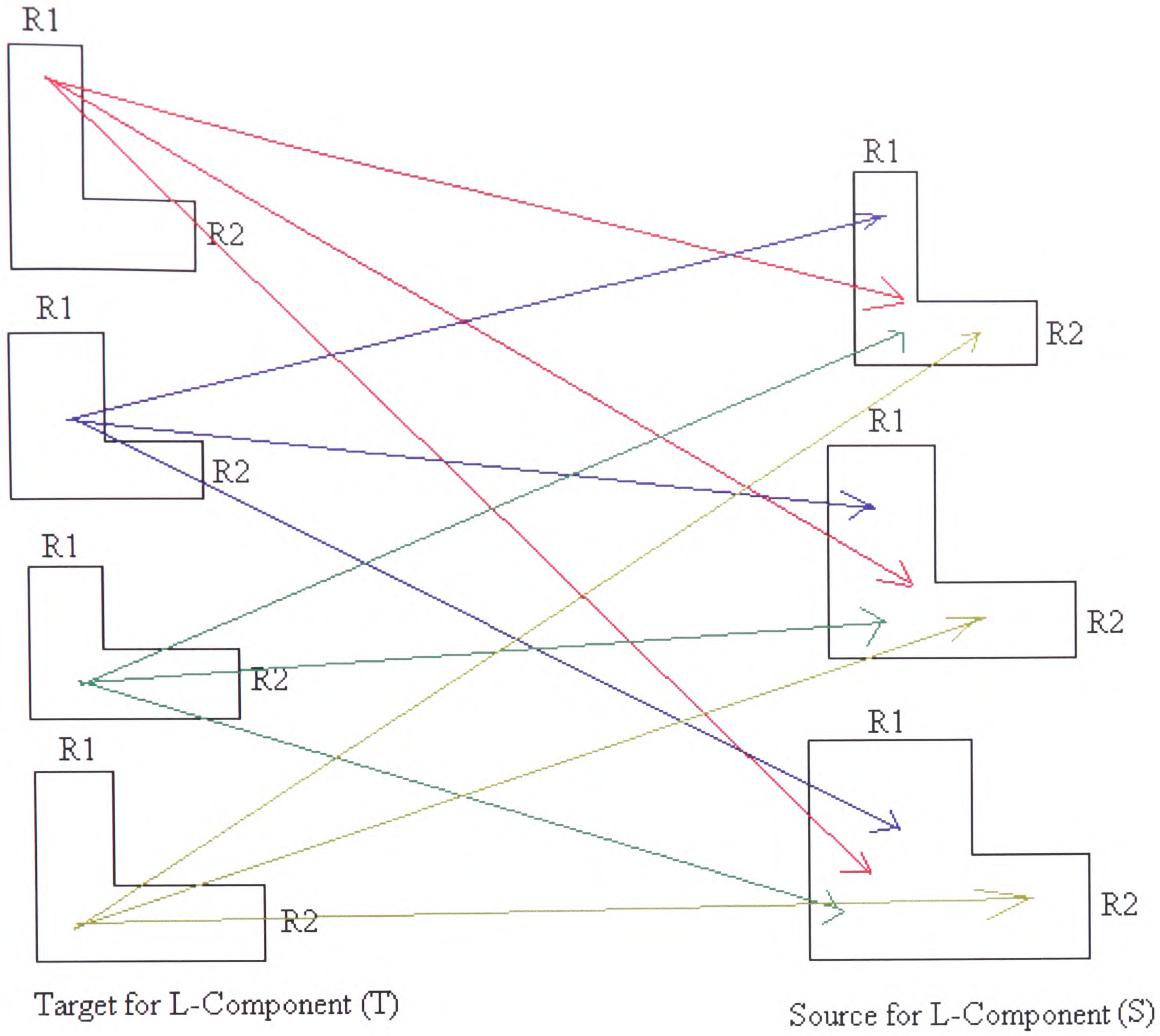


Fig.5.5 shows L-components for Target L's to the left side and Source Ls to the right side.

On the Table-5.0 and Figure 5.5 shows the process of the calculation for each target shape to all source components in terms of properties and the size of each component.

The first column from the table shows all the target components (T1, T2, T3 and T4) and compares to the first L-components source (S1) then it divides over the numbers of the targets. The second column from the table is shown all the target components (T1, T2, T3 and T4) and compares to the L-components source (S2) then it divides over the numbers of the targets. The third column from the table is shown all the target components (T1, T2, T3 and T4) and compares to the L-components source (S3) then divides over the numbers of the targets. Finally the fourth column from the table is shown all the target components (T1, T2, T3 and T4) and compares to the L-components source (S4) and divides over the numbers of the targets. In order to get the similarities between for target and source components, an average is obtained by adding the value metric for each comparison and dividing by the number of type's component.

$\sigma_{Comptype}$ (S1, S2) is a measure based on the types of component with their numbers that are common to the two graphs.

The equation of the component type metric is:

$$\sigma_{compType}(S1, S2) = \frac{\sum_{ComponentType} \sigma_i \ (S1, S2)}{noTypes} \tag{5.2}$$

Table-5.5 shows the average for example L-components for both (S1, S2). S1 represents (source shape) and S2 represents (target shape).

Additionally, there could be an argument for creating a similarity metric between components of different types (e.g. between a Bar and a taper). However, as the role of a particular component type is a structural one, it was decided to assume that components of different types are dissimilar. Further work could investigate this assumption further. The next section 5.7.4 is discussion on the "Maximum Common Subgraph" metric.

5.8.3 Maximum Common Subgraph Metric (MCS)

Gebhardt [Gebhardt, F: 1997] argues for retrieval of shapes for casting design representing complex structures. Features are difficult to define, and similarity must be derived from structure directly. It is not unexpected when allowing for important information can be collected from how single components are connected spatially (see Fig5.4).

Maximum common subgraph is one of the stringent metric for similarity measuring in the sense that it has the feature that is closely related to the layout of the components of the shape. Two shapes can have the same number of each type of component (that is the same total weight) and still be far different graphically, for example MCS, identify the largest sub-graph common to a pair of graphs

The algorithm takes two graphs for example: G1 represents target graph and G2 represents source graph. The step is the system takes two nodes, we assume (a) and (b) and both of them represents component types. The algorithm checks whether the nodes are match or not. If not then the algorithm search for the neighbouring nodes that having been searching before by recursively and also search for the subgraphs. When the two nodes (a) and (b) are not match, then the maximum common subgraph will remain from present couple of nodes, is set at zero and the algorithm returns the maximum remaining subgraph from present couple of nodes.

The equation of the Maximum Common Sub-graph metric (MCS)

$$\sigma_{mcs}(S1,S2) = \sum_{comp} \frac{\text{length}(\text{S}')^2}{length(S1)length(S2)} \qquad (5.3)$$

Example Assuming that S1 and S2 represents two different shapes.

The system first reads all nodes (nodes represents the components) from data files for S1 and then identifies the type of first components, with their data dimension of the component and at the same time, the number of connections is given for these particular components then the system searches for a second node of each 3D shape are used only once, so that the above measure is maximised for a target and case 3D shape. We find the most similar tuple of sections from each shape s1 and s2, .Then remove these sections or views and find the second most similar section. In the care the overall measure of similarity between two 3D shapes needs to be redefined as equation 3D.

An Example of the concept in graph matching for shape retrieval is the maximum common subgraph. A maximum common subgraph of two graphs, (G1) and (G2), is defined as a graph $\sqrt{G}$ that is a subgraph of both (G1) and G2, which has the maximum number of nodes as compared to all the possible subgraphs of (G1) an (G2) [Anandan S. and Summers D. J 2006]. Figure 7.0 below shows an example on a maximum common subgraph of two graphs; the graphs represents two 2D shapes for ((G1) and (G2)) and the nodes represents, components. G1 (T, B) matching with graph G2 (T, B). The (T) represents T-shaped and The (B) represents Bar type (see on Fig.5.6 and the solution of this diagram can be seeing on equation 5.2a.

The Vertex (Nodes) represents the components and the Edges represent the connector bars and tapers between the structural components such as L, T, and X-components).
The result of the MCS between these two graphs (S1, S2) is:
The equation below 5.2a presents the solution for Fig.5.2 example.

$$\sigma_{mcs}(S1,S2) = \sum_{comp} \frac{(2)^2}{(7)*(6)} \times 100 = 0.952\% \qquad 5.3\text{ a}$$

In next section a discussion on similarity measures for shape retrieval using graphical representations for shape matching is given.

In next section a discussion on similarity measures for shape retrieval using graphical representations for shape matching is given.

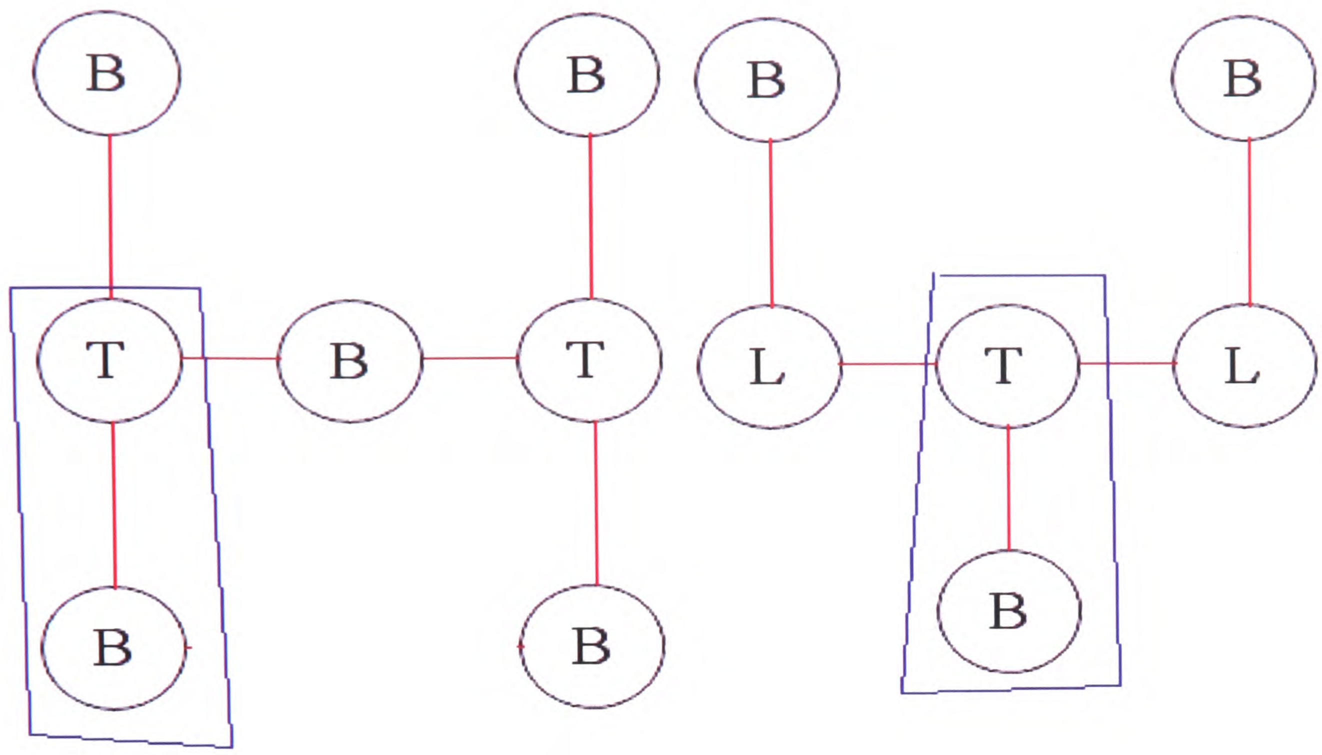


Figure 5.6 Shows graphs that represent two 2D shapes. (G1 to the left side1 and G2 to the right side2).

The next Section 5.7.4 introduces the bar type (B) metric and we called it the leaves metric.

5.8.4 Leaves Metric (for Bar Type (B))

Some components of shapes are only stuck to other components. That means only one of their interfaces is used. Those components in the shape layout are represented at the border of the shape. A shape always has two or more leaf components. It is therefore important to know how many leaf components a shape has, because it can help us in knowing the layout. These metrics return to us a ratio less than one (1); and it is the result of the division of the total number of leaf components of the target shape divided by the total number of leaf components of the retrieved shape. The equations below calculate a ratio for all connector bar type (B) for both source and target shapes.

Equation for the leaves metric is:
If max value of the leaves > 0, then

$$\sigma_{leaves}(S1, S2) = 1 - \frac{|\text{nleaves(S1)} - \text{nleaves(S2)}|}{\max(\text{nleaves(S1)}, \text{nleaves(S2)})} \quad \textbf{(5.3)}$$

If there are no leaves then the value is set to 0.5.

All those features participate in the general similarity measure. The result of general similarity measure is also within 0 and 100. It is the probability of resemblance of those two shapes being compared. The value one is the certainty of resemblance, while zero represents the certainty of dissemblance. Each feature participates to the general similarity metric by the importance of its influence in the resemblance. And to each of these features is associated a constant of ratio w, which belonging to [0, 100] w multiplies the ratio of the similarity metric of that feature to give its exact contribution in the general similarity ratio. And the general similarity ratio is the sum of similarity ratios of features multiplied by their correspondent constancy of ratio of importance (Wi). See **equation of Watson** (Watson 1996).

$$Sim_{(T,S)} \sum_{i=1}^{n} f(T_i, S_i) \times W_i \tag{5.5}$$

where T (target shape) and S (source shape) are shapes, n is the number of features (attributes) (i) is an individual feature going from 1 to n; f is a similarity equation of feature in shape T and S; w(weight) is the importance of influence of feature (i).

However the approach in this chapter carry out to design an algorithm to construct an automating retrieval tool with which to investigate the efficiency of various metrics with respect to different casting designs, as have been illustrate in figure 5.8 for CBR retrieval system. The system employs a generalised similarity measure between two shapes S1 and S2, representing a weighted sum of similarity measures based on different features extracted from both graphs structural (MCS) and dimensional data (property of S1 and S2) see the equation below for generalising the similarity measure between shape S1 and S2.

$$Sim_{(S1,S2)} = W_{CompType} \times Sim_{CompType} + W_{msc} \times Sim_{msc} + W_{leaf} \times Sim_{leaf} \tag{5.6}$$

Variation of the weights (w) in this formula allows a general test of retrieval against any given casting target (see the example general test on Fig.5.7 (a) and Fig.5.7 (b). This section shows the architecture for shape retrieval for similarity metric.

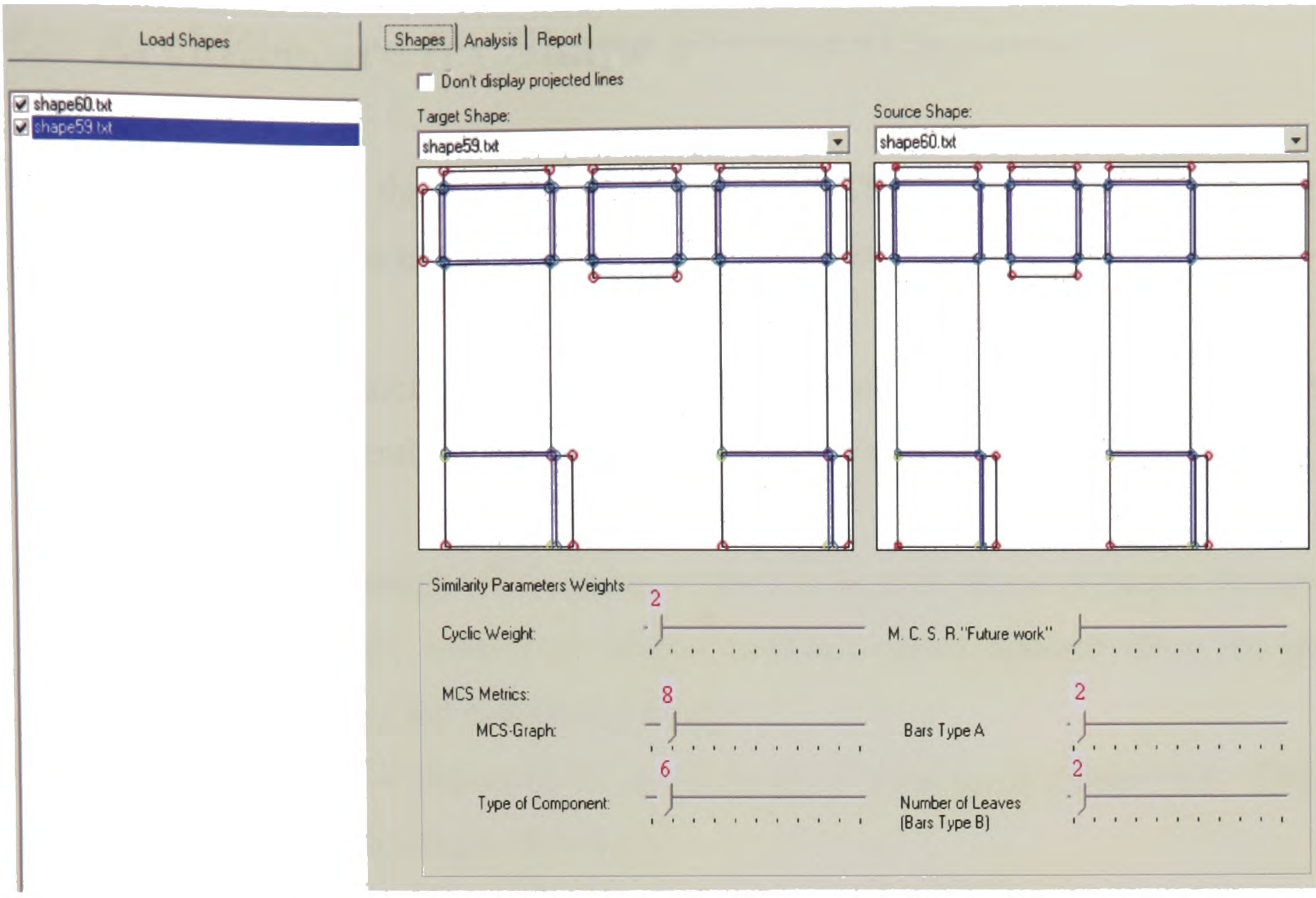


Fig.5.7 (a) shows similarities (in terms of Cyclic, MCS, Component Types and Leaves) between Targets ShapeID_59 (S1) and Source shapeID_60 (S2).

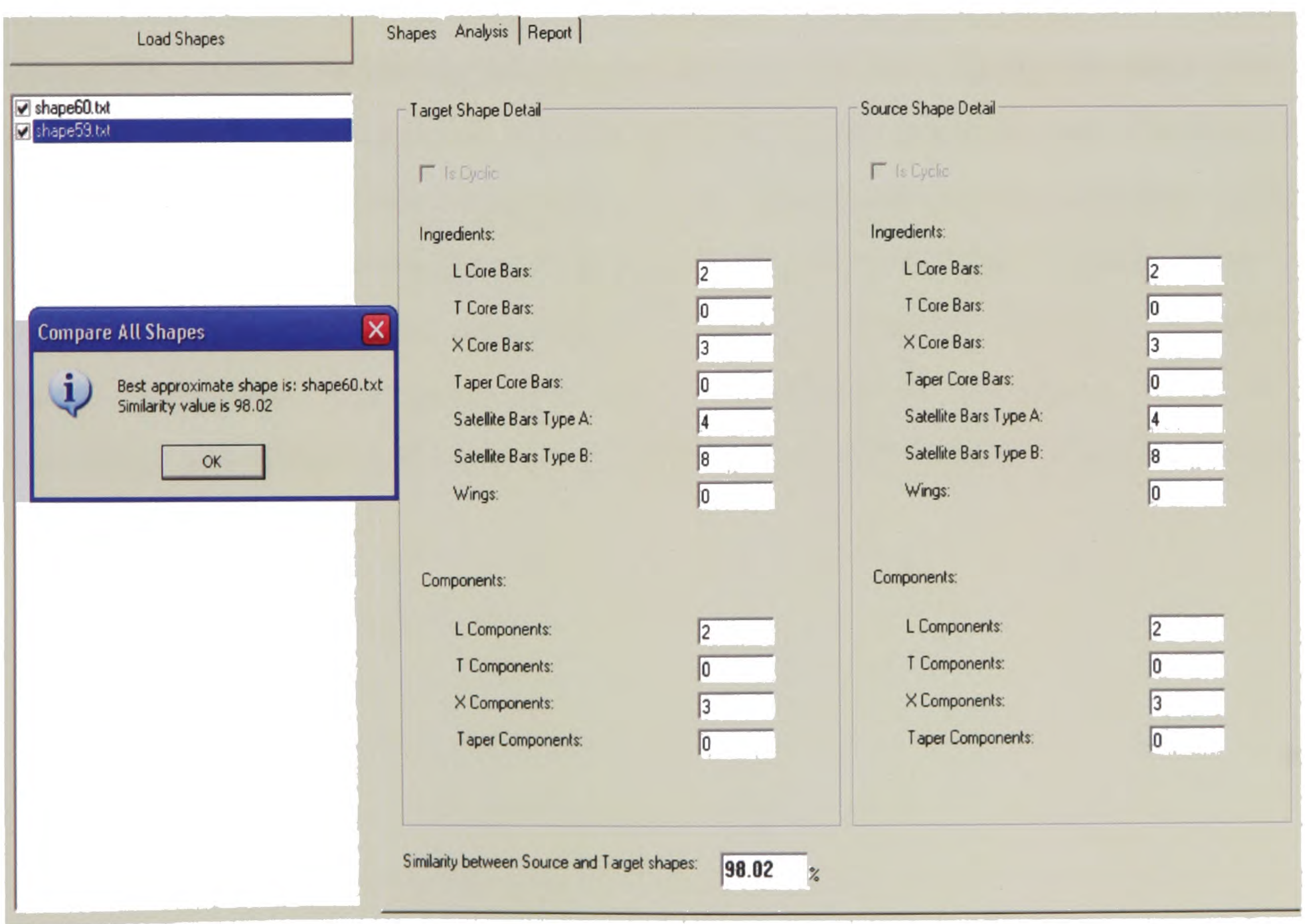


Fig.5.7 (b) shows the results of sum similarities between Target shape ID_59(S1) and shapeID_ 60(S2).

Equal weight (0.25) Q (S1, S2) = 100 + 92.08 + 100 + 100 = 98.02%.
Equal weight (0.25) Q (S1, S2) = 94.10% [Mileman: 2000].

5.9 Architecture for Shape Retrieval System

This section discusses the architecture of the ShapeCBR system used for the retrieval of similar shapes to a given target containing relevant casting design advice.

For the current research, a number of algorithms have been developed for shape decomposition, type identification, and similarity based retrieval.

The system searches through all shapes in a typically large database of shapes that are similar to a target query shape. Usually all shapes within a given distance from the query are determined (Fig. 5.8 on CBR Retrieval system).
The experiments will be discussed in detail in the evaluation in chapter 6.0. This section explains the algorithm of shape retrieval.
Figure 5.7 describes the steps of the shape processing CBR retrieval system Architect.

In the first step is the user draws a 3D shape in a CAD package and inputs into the ShapeCBR system. Following this, the second step involves slicing the shape into a set of different views (cross sections). Each view is separated manually and decomposed into a generic set of disconnected components. Then the components are classified automatically into identifiable component types such as L, T, X, bars and Tapers. Finally, the retrieved K nearest shapes are retrieved and presented to the user together with any associated casting design knowledge. The target case can be annotated by an expert user with casting knowledge and added to the case base if it adds to the knowledge base and diversity of the case base.

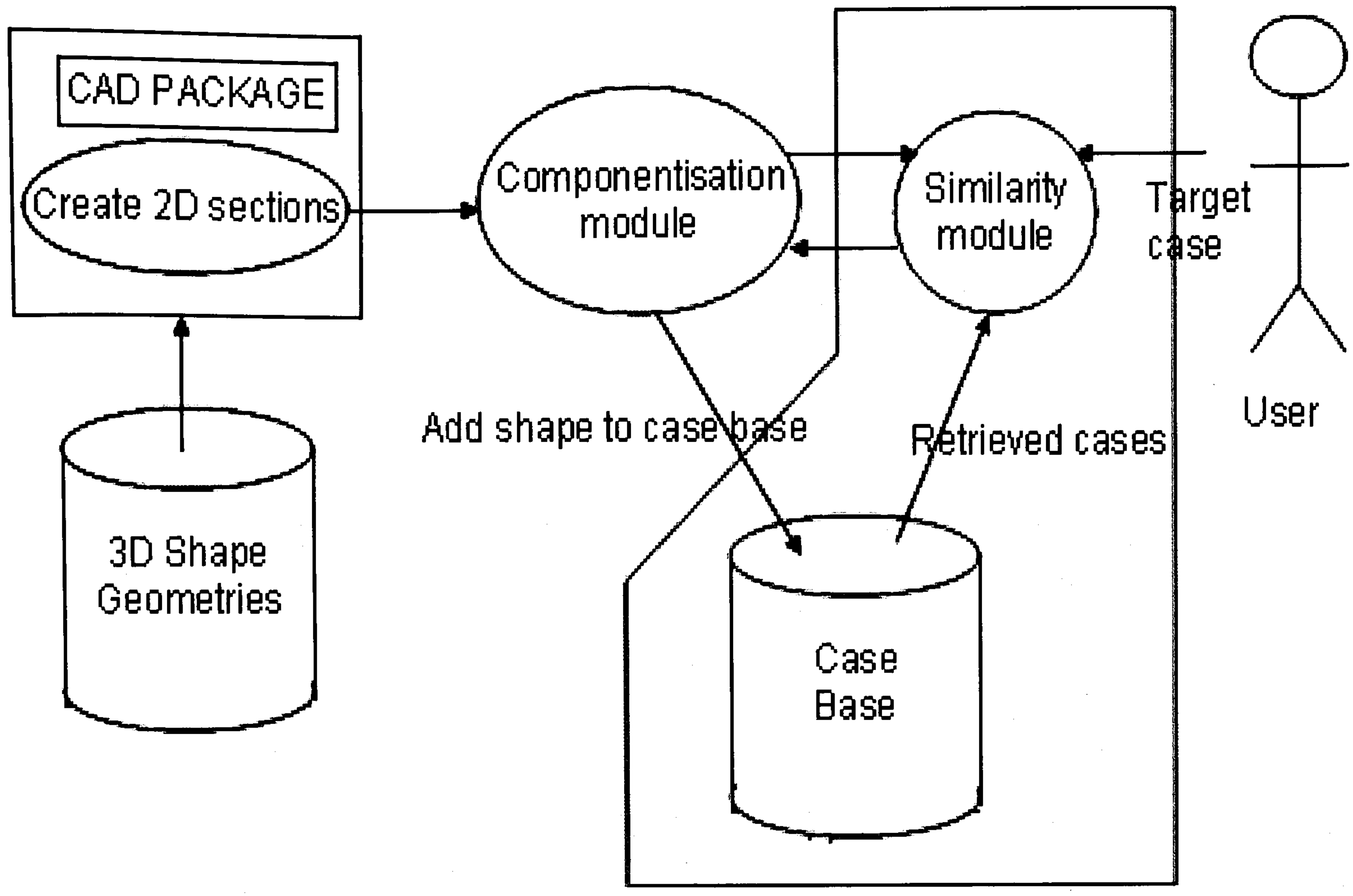


Fig. 5.8 shows the CBR Retrieval System Architecture.

The next section presents the overall similarity metrics between 3D shapes. This section can attempt to answer one of the similarity metrics questions.

5.10 Overall Similarity Metrics between 3D Shapes

This section presents several references some of them are close to this area (shape matching for shape retrieval), of how to perform efficiently graph matching to all the moulds. Graph matching problems are based on the idea of having more than one view, and of performing graph matching to a database of views so that the view that best approaches the features of the data graph is selected. Therefore, the aim here is to recognise a view rather than going deeply to recognise each of the sections of the data view, Messmer [Messmer and Bunke, 1999]. Several references can be found on performing efficiently graph matching to all the models. For instance, in from a more theoretical point of view, [Williams et al., 1997] describes the development of a Bayesian framework for multiple graph matching. The starting point of this proposal is the Bayesian consistency measure developed by Wilson and Hancock [Wilson and Hancock, 1996] which is generalised from *matching graph pairs to multiple graphs*.

In [Huet and Hancock, 1999] a graph-matching technique for recognising line-pattern shapes in large image databases is described matching algorithm that uses edge-consistency and vertex attribute similarity.

Multiple graph matching has also been applied to many other problems such as the comparison of saliency map graphs [Shokoufandeh et al., 1999 and 2000].

The overall similarity is carried by comparing the views(multiple graphs) of the target shape with views of source shapes to find the most similar shape in the case base using the metric, overall similarity. See equation 5.8 and fig 5.0. We discuss the process in the following paragraph.

The first step of the experiment starts with slicing the two 3D shapes into a number of views, as have been explained in literature reviews chapter 2 on CAD application, saving that complex shapes such as 3D shapes have multi views. It is easer to slice into a number of views to identify the internal geometrical information of the shapes. Therefore, for current research on 3D shapes and 2D shapes have been used the same method (as a number of 2D views representing 3D shapes).

Over 20 new cases (shapes) as targets have been tested over 100 cases from a large database of shapes that are similar to a query case, comparing target views and source views to find similarity metrics between the two 3D shapes. For this purpose, a equation (Equation 5.7) has been created to calculate the overall similarity metrics between the target case and source case.

The equation calculates each individual view from the target case against each individual view from source case. The overall results calculations for all views are divided over the number of sections for both target case and source case. For example, Table-5.0 shows as an example for overall similarity between target (S1) and Source (S2) with their calculations resulted that have been generated through ShapeCBR system. In this case, the overall measure of similarity between two 3D shapes S1 and S2 (in terms of the component type's metric, the MCS metric, the leaves metric and the cycles metric. It needs to be redefined as:

Equation for overall similarity metric

$$\sigma_{3D}\,(S1,S2) = \Sigma\ w_i\sigma(S1_{2Dsection\ n},S2_{2Dsection\ m}) \qquad (5.7)$$

Where each 2D section (n, m) of each 3D session is used only one so that the above measure is maximised.

- 3D (S1, S2): S1 represents 3D target shape as have been suggested before and
- S2 represents the 3D source.
- Wi: represents the weighting for both target cases and sources cases.
- S1 2D section (n) represents the number of target views or cross section.
- S2 2D section (m) represents the number of source views or cross section.

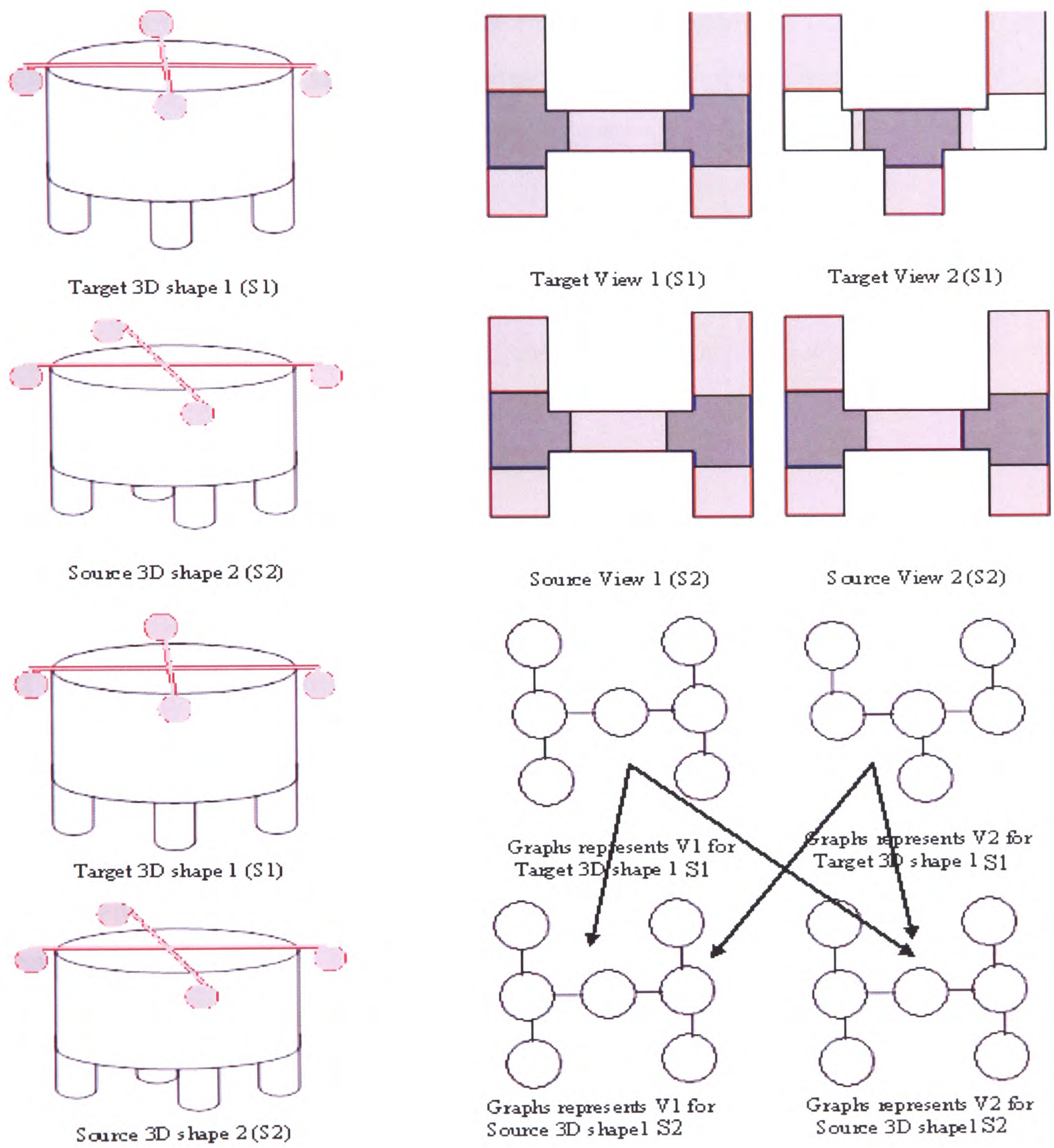


Table-5.1 shows overall similarity between two arbitry3D shapes.

The right site of table 5-0 shows a comparison between two 3D shapes (S1, S2). The first row shows the 3D target shape (S1) that has been sliced over CAD application into enough dissimilar cross sections or views and the result was, there are two different views (V1, V2) and second row shows the 3D source case (S2) that has been sliced over CAD application into enough dissimilar cross sections or views and the result was, there are two views (V1, V2 and these two views are between 85-100% similar by showing them to the expert [Aziz: 2004]). The (V1) from target (S1) shape is comparing with (V1, V2) from source (S2) in terms of features calculation aspect ratio ($R_L = \frac{R_1}{R_2}, where, R_1 < R_2$) for each target features or components and source component types for both shapes (S1, S2). The comprising process is searching for features such as a numbers of leaves, a number of cycle, a number of MCS and a number of component types. For (V2) from target (S1) compares with both (V1, V2) from source (S2) shape and the comprising process is searching for a numbers of leaves, a number of cycle, a number of MCS and a number of component types. The final result is the sum of features for the four views (V1, V2 for (S1)) and (V1, V2 for (S2)) then the result for the four views are divided over the number of source (S2) views. For overall similarity for the two examples of 3D shapes have been set up, one is represents target shape (S1) and the other one represents source shape (S2). The example is shown on table-5.0.

The Solution for overall similarity for the trial examples two 3D shapes (S1, S2) in table-5.0:

1. **Component Type Metric with their numbers (S1, S2):**

V1 (S2) contains: "5"bars and "2"T-components.
V2 (S2) contains: "5"bars and "2"T-components.
V1 (S1) contains: "5"bars and "2"T-components.
V2 (S1) contains: "5"bars, "2"L-components and "1" T-component.
The similarity for target V1 (S2) with V1and V2 in (S1) is **100%** similar in terms of component types.
The similarities for target V2 (S2) with V1and V2 in (S1) are **75%** similar in terms of component types.

The overall similarity (S1, S2) in terms of component types are:

(a) $$\sigma_{compType}(S1,S2) = \frac{\sum_{ComponentType} \sigma_i \ (S1,S2)}{noTypes} \frac{75\% + 100\%}{2t \arg et} = 87.50\%$$

2. **MSC Metric (S1, S2):**

The similarity for target V1 (S2) with V1and V2 in (S1) is **100%** similar in terms of MSC because they are identical.

The similarities for target V2 (S2) with V1and V2 in (S1) are **30.769%** similar in terms of component types.

(b) $\sigma_{mcs}(S1,S2) = \sum_{comp} \frac{(2)^2}{(7)*(6)} \times 100 = (0.952\%) + (100\%) \div 2taget = 54.76\%$

3. **Leaves Metric (S1, S2):**

V1 (S2) contains: "4"leaves.

V2 (S2) contains: "4"Leaves.

V1 (S1) contains: "4"Leaves.

V2 (S1) contains: "3"Leaves.

The similarity for target V1 (S2) with V1and V2 in (S1) is **100%** similar in terms of Leaves are identical.

$$\sigma_{leaves}(S1,S2) = 1 - \frac{|\text{nleaves(S1) - nleaves(S2)}|}{\max(\text{nleaves(S1), nleaves(S2)})} = 1$$

1= **100%** "according to leaves role (both views are equals in terms of leave)".

The similarities for target V2 (S2 has 3 Leaves) with V1and V2 in (S1) are **75%**

$$\sigma_{leaves}(S1,S2) = 1 - \frac{|\text{nleaves(S1) - nleaves(S2)}|}{\max(\text{nleaves(S1), nleaves(S2)})} = 75\%$$

(c) Overall similarity (S1, S2) in terms of Leaves is: 100% + 75% = 87.5%

4. **Cycles Metric (S1, S2):**

None of them cycles and equal to Zero (zero it means 100%) according to the cycle role (see on the section 5.7.1).

$\sigma_{cycles}(S1,S2) = 1 - \frac{|\text{ncycles(S1) - ncycles(S2)}|}{\max(\text{ncycles(S1), ncycles(S2)})}$ **(d)** Overall similarity (S1, S2) in terms of cycles is: 100%

The final overall similarity between S1 (V1, V2), S2 (V1, V2) =

σ_{3D} (S1, S2) = Σ w_i σ(S$1_{2Dsection\ n}$, S22Dsection$_m$) =

(a) + (b) + (c) + (d) = 87.5+54.76+87.5+100= 82.44%.

The similarity test between the target cases and the source cases calculates in terms of: the component types and their numbers, the maximum common subgraph, the leaves and the cycles. Final results for the overall similarity between the two trial 3D shapes Or between the cross-sections for (S1, S2) equal to 82.44%.

Overall similarity equation have been created and can be seen on equation 5.7 and an Example solution can be seen on 5.7.1.

$$\sigma_{3D}\,(S1,S2) = \Sigma\ w_i\sigma(S1_{2Dsection\ n}, S2_{2Dsection\ m}) \qquad (5.7)$$

$$Overall_{Sim}\,(S1_{2Dsection\ n}, S2_{2Dsection\ n}) = w_{componentType} \times Sim_{ComponentType} + W_{msc} \times Sim_{msc} + W_{leaf} \times Sim_{leaf} \qquad (5.7.1)$$

Equation (5.7.1) has been generated from equation (5.7).

Variation of the weights (w) in this formula allows a general test of retrieval against any given casting target.

Where each 2D section (n, m) of each 3D session is used only one so that the above measure is maximised.

The equation of overall similarity abbreviations have been defined as:

- 3D (S1, S2): S1 represents 3D target shape as have been suggested before and
- S2 represents the 3D source.
- w: represents the weighting for both target cases and sources cases.
- S1 2D section **(n)** represents the number of target views or cross section.
- S2 2D section **(m)** represents the number of source views or cross section.

5.11 Evaluation of over all similarity algorithm

A number of new 3D cases have been tested in ShapeCBR and evaluated against a casting domain expert. For this research, 100 new 3D cases were fed into Case CBR and the resulting overall similarity was compared to the human expert [Aziz Muhammad: Jun. 2003] The result was that in all of the cases can be seen in table-5.2 (and see on table-6.4a and table-6.4b). In order to evaluate the 20 new 3D cases and personal conduct have been done through visiting a small foundry fabric (Slemani Cement) in metal casting in Slemani to get the advices for these 3D shapes and the advice was scored with value percentage as shown in the table-5.2.(More details on chapter 6).

Expert	Excellent %	Satisfactory %	Indifferent %	Bad %
Casting design Engineer: Aziz, M.	75% - 80% of the shapes.	75- 80% of the 3D shapes	None	None

Table-5.2 shows the advices score for 3D shapes.

5.12 Conclusion

This chapter addressed the problem of similarity metrics for shape retrieval using graphical representation for shape matching in case-based reasoning.

Similarity metrics have become a valuable instrument in the case-based reasoning and shape-matching fields. Additionally, they can offer a supplement to other methods of analysing patterns and behaviour similarity behaviour.

The problem of similarity metrics for shape retrieval is based on two extremely important keys: the structure (graphs) of the shape and the properties (features) of the shape. The structure and the properties in pattern recognition play an important role for the shape retrieval process.

This chapter also presented a new metric for similarity measures and it is "the number of component types", and it is a new contribution furthering the Mileman [Mileman: 2000] research.

A number of metrics for shape retrieval have been posed by Mileman [Mileman: 2000] and additional new metric have been created by the author calls" Component Type Metric" (CTM) for improving the efficiency of the similarity measurement for shape matching and shape retrieval. A number of 3D shapes have been tested in ShapeCBR system to test "Component type metric" and to see the differences with pervious metric. The result is promising with the new metric and this result can be seen in chapter evaluation.

The metrics describe a set of equations to calculate the structure (graph) and properties (features) of the shape in terms of the sum of: the MCS metric, the component type metric (CTM), the cycle metric and the leaves metric.

The next chapter is dissections on the ShapeCBR system evaluation and the second expert evaluation on 3D shapes.

Chapter 6

Acquiring and evaluating casting design knowledge using ShapeCBR

For evaluation purposes it is important to determine how consistent and reliable is expert knowledge elicited from various domain experts in casting design. This chapter presents the conclusion of the main goals; similarity metrics between 3D shapes for shape retrieval in CBR.

Over 100 cases of casting shapes including casting design knowledge have been set up in previous research [Mileman: 2000] for tests and a number of new 3D shapes have been designed. The shapes have been tested using the ShapeCBR system to determine if there are improvements in the efficiency and performance of the system and the CBR method itself.

6.0 Introduction

The previous chapters presented similarity metrics for shape retrieval. The approaches adopted need to be evaluated in terms of their suitability to provide answers to the research questions posed at the beginning of this thesis. There is a body of research that looks into the evaluation of CBR systems and tools [Althoff K. D.: 1995].

The reason for the evaluation presented in this chapter is to verify the efficiency of the performance of shape retrieval to predict useful, competent designs of a given target case and to give solutions for new shape designs in terms of: orientation of the feeders, chills and the expert advice.

The final result from the achieved advice has been tested against human expert judgment to determine the performance of the approach.

This chapter examines the evaluation of similarity measurement for shape retrieval metrics which has already been discussed in detail in chapter 5, to achieve a best advice or combination advice of each separate metric performance in solving casting design problem.

Over one hundred shapes have been derived from a realistic domain of rotationally symmetric cases created and sliced into a number of sufficiently dissimilar cross -sections or views sliced through CAD modelling, taken from domain knowledge. Furthermore, the cases have been tested in the ShapeCBR system. ShapeCBR have been developed at the University of Greenwich to retrieve similar cases automatically by given a target shape, and can be used as a tool to identify the efficiency and performance of each metric advice.

The final results for each test for a casting design problem have been compared with two experts (first expert and second expert) and both casting designs methods will present within this chapter in details.

6.1 Review of the casting problem

Mileman [2000] assumed that a 3D shape has one view, and that the thickness (size) of the shape does not play an important role in casting design.

This research discriminates against Mileman assumptions; firstly, in this research, 3D shapes present as a number of dissimilar or similar views. Furthermore, the size of the shape matters; aspect ratio has been introduced for the thickness or face calculation. The expert pointed out that these issues are understandable in casting design and should therefore require a reason to be given. For the above reason, the second expert explains that the quality of casting designs depends on a number of factors to avoid casting problems such as porosity and shrinkage:

- **Casting process timing**, for example during the casting operation, when the metal drops slowly or faster, into the feeders. The two ways process may affect shrinkage, porosity and cracking.

- **Weight for casting designs** is another factor. For example, if the weight is over 20kg, their designed casting needs to utilise one to two feeders (Aziz expert in casting design). If the weight is above 50kg then more feeders are needed and a bigger number of chills required (the types of sand used for chilling down the metal and raise the quality of casting metal designs).

In above factors consider that the shape size and weight are two of the most important criteria for increasing or decreasing the number of feeders and the number of chills. It seems that the expert is right, regarding the weight and shape size. Also, In order to solve the metal shrinkage or cracks problem, the experts need re-check the casting design again to see the actual problem. The investigations and illustrations by the second expert show that it is important to consider the size of the shape and casting timing. The next Section introduces the role of CBR in shape retrieval.

6.2 The process of CBR in ShapeCBR

This section presents the role of CBR in evaluating the similarity metrics. The CBR technique has been used as a research key to support and solve research problems.

The Case-based reasoning (CBR) experts are personified in case-based knowledge of past cases, rather than being encoded in classical rules. Each case typically contains a description of the problem, plus a solution. The knowledge and reasoning process used by an expert to solve the problem is not recorded, but is implicit in the solution.

In order to evaluate ShapeCBR system over CBR, four experiments have been set up and they are: the first test was to test 20 target cases against 100 source cases from case-based knowledge to find similar or sufficiently close cases. The retrieved cases are used to suggest a solution which is reused and tested for success. If necessary, the solution is then revised. Finally the current problem and the final solution are retained as part of a new case. The next paragraph presents the CBR methods. The case-based reasoning methods are presented as follows:

1. Retrieve the most similar case (or cases), comparing the case to the case-based knowledge in which they have been stored in the past.
2. Reuse the retrieved case to try to solve the current problem.
3. Revise and adapt the proposed solution if necessary.
4. Retain the final solution as part of a new case.

There is a variety of different methods for organising, retrieving, utilising and indexing the knowledge retained in past cases.

Retrieving a case starts with a (possibly partial) problem description and ends when a best matching case has been found. The subtasks involve:

In identifying a set of relevant problem descriptors, matching the case and returning a set of sufficiently similar cases (given a similarity threshold of some kind); and selecting the best case from the set of cases returned.

Some systems retrieve cases based largely on superficial syntactic similarities among problem descriptors, while advanced systems use semantic similarities.

Reusing the retrieved case solution in the context of the new case focuses on: identifying the differences between the retrieved and the current case; and identifying the part of a retrieved case which can be transferred to the new case. Generally the solution of the retrieved case is transferred to the new case directly as its solution case. Revising the case solution generated by the reuse process is necessary when the solution proves incorrect. This provides an opportunity to learn from failure.

Retaining the case is the process of incorporating whatever is useful from the new case into the case base. This involves deciding what information to retain and in what form to retain it; how to index the case for future retrieval; and integrating the new case into the case-based knowledge. The constitution of the CBR system used in this research which matches this approach is shown is shown in Figure 6.0.

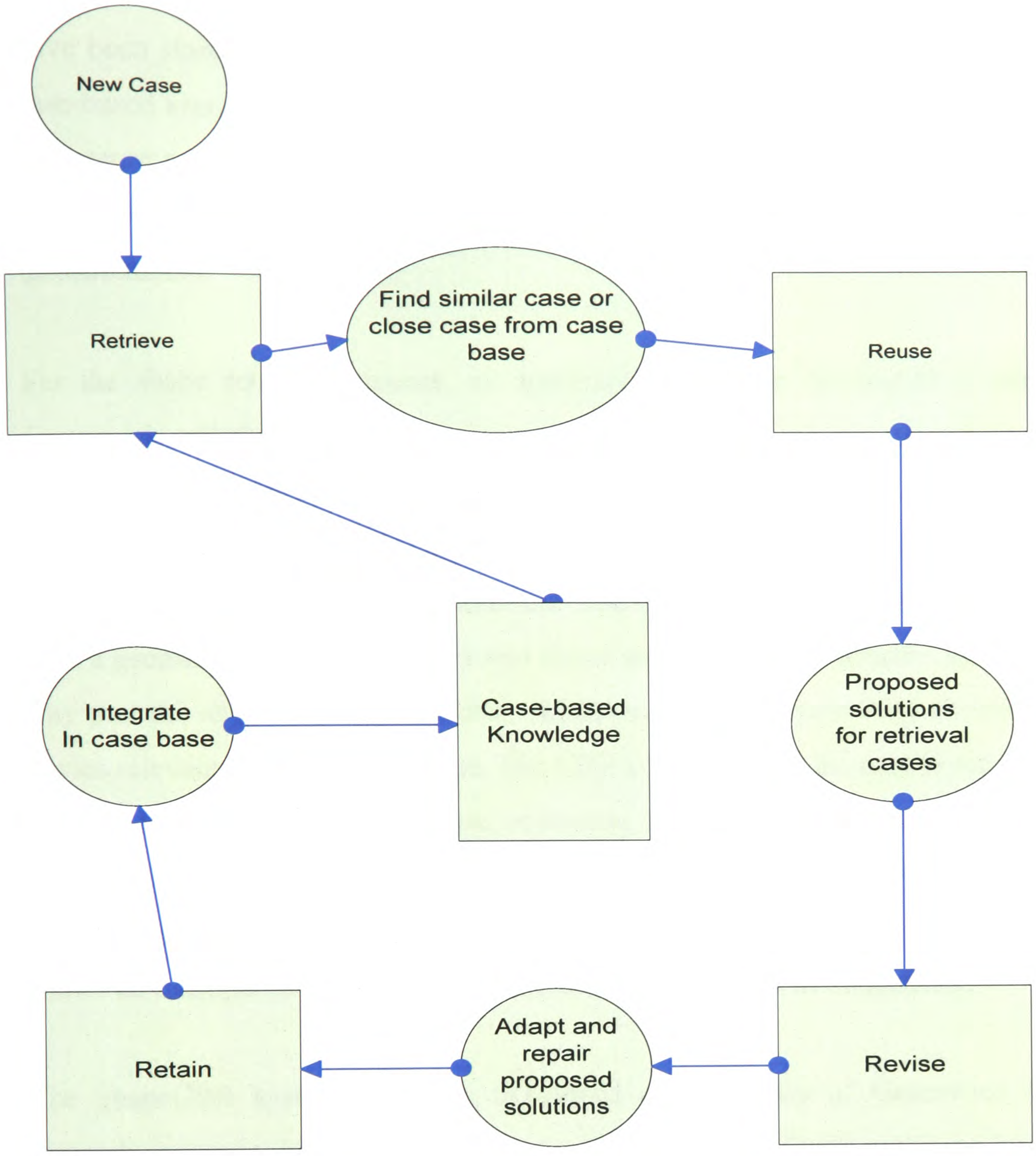


Fig.6.0 CBR techniques support shape design system in current research.

Case: in this research, the case represents a number of views or shapes of casting design and it represents a 3D shape. The shapes have been sliced using a CAD application. Each view is made up a list of components such as Bar, Taper, T-component, X-component and L-component.

The shapes are drawn in the CAD application, then sliced through CAD into different views and saved into a case base, to be ready for decomposition and classification process and the products of these two process are stored in case base to be ready for reuse in shape retrieval in CBR for remembering them for reusing. Case base has been defined by Taylor (Taylor: 1997: pp: 136) as "The memory of past experience".

Case-based knowledge, represents a database store, that contains a large number of cases that have been stored in the past, for the shape retrieval process for comparing each case in the case-based knowledge with a new target case and the outcome of the shape retrieval process, will return a measure of similarity metric between the target shape (new case) and the stored case (cases which have been stored in past). For more details see chapter five on similarity measurements.

For the shape retrieval process, an application has been developed at the University of Greenwich called ShapeCBR to automate the shape drawing process, shape decomposition process, shape classification and shape matching process by a given target shape to a case in case-based knowledge. The case base is populated with cases containing information relevant to real metal casting experience. The information contained in each case relates to both a geometrical description of a real shape and a domain of specific information about the way that the shape was actually cast. Additionally, some cases may contain general expert advice relevant to casting the shape. The CBR system allows the user to retrieve a shape from the case base to match a target case, according to a match on the **four contributing** Fig. 6.0 features as described in the previous section. Weighing factors can be applied by the user to attach varying importance to each of the similarity measures. The next page in figure 6.1 shows an example of shape matching a case to a target case in ShapeCBR.

The ShapeCBR system has been developed at University of Greenwich for the current research purposes. Figure 6.1 displays an example of the shape matching by giving a target case to find the similar or closest case. On the left side of the system interface, the source case is shown from the case-based knowledge and the next to the right display the target case.

Figure (6.1a.) and Figure (6.1b) shows the similarity parameter weightings for all similarity metrics. These weightings are applied manually by the users. Figure (6.1a.) illustrates the target shape from the right side of the GUI and result displays on the left side of the GUI.
Figure (6.1b.) from the right side is illustrating the elements and the components for the target shape. The results for similarities between the target and source shapes in terms of element and components can be seeing on the left side of the GUI. These elements and components have been generated through the ShapeCBR system.

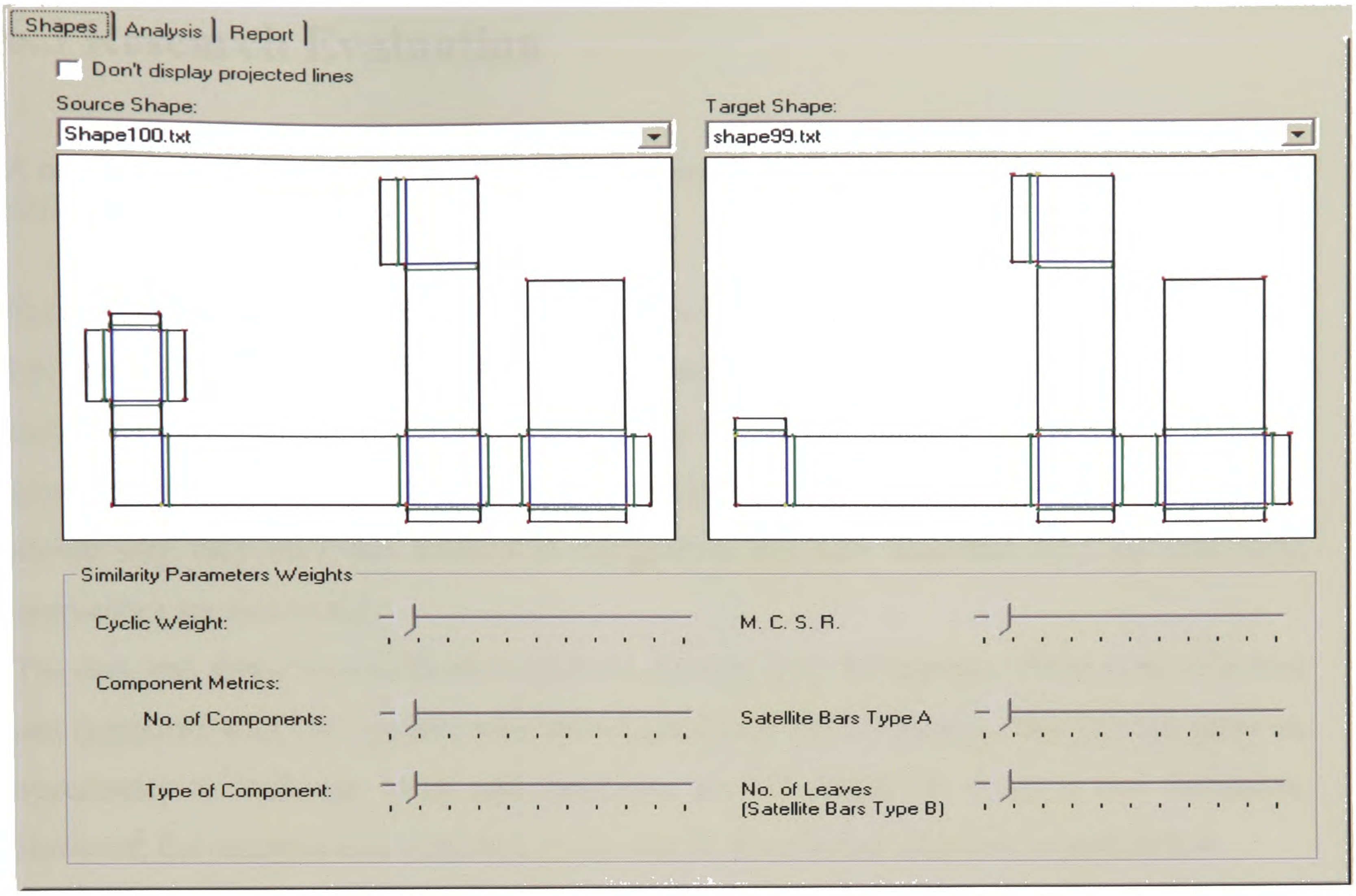


Fig.6.1a. shows an example in shape matching between target case and source case in ShapeCBR.

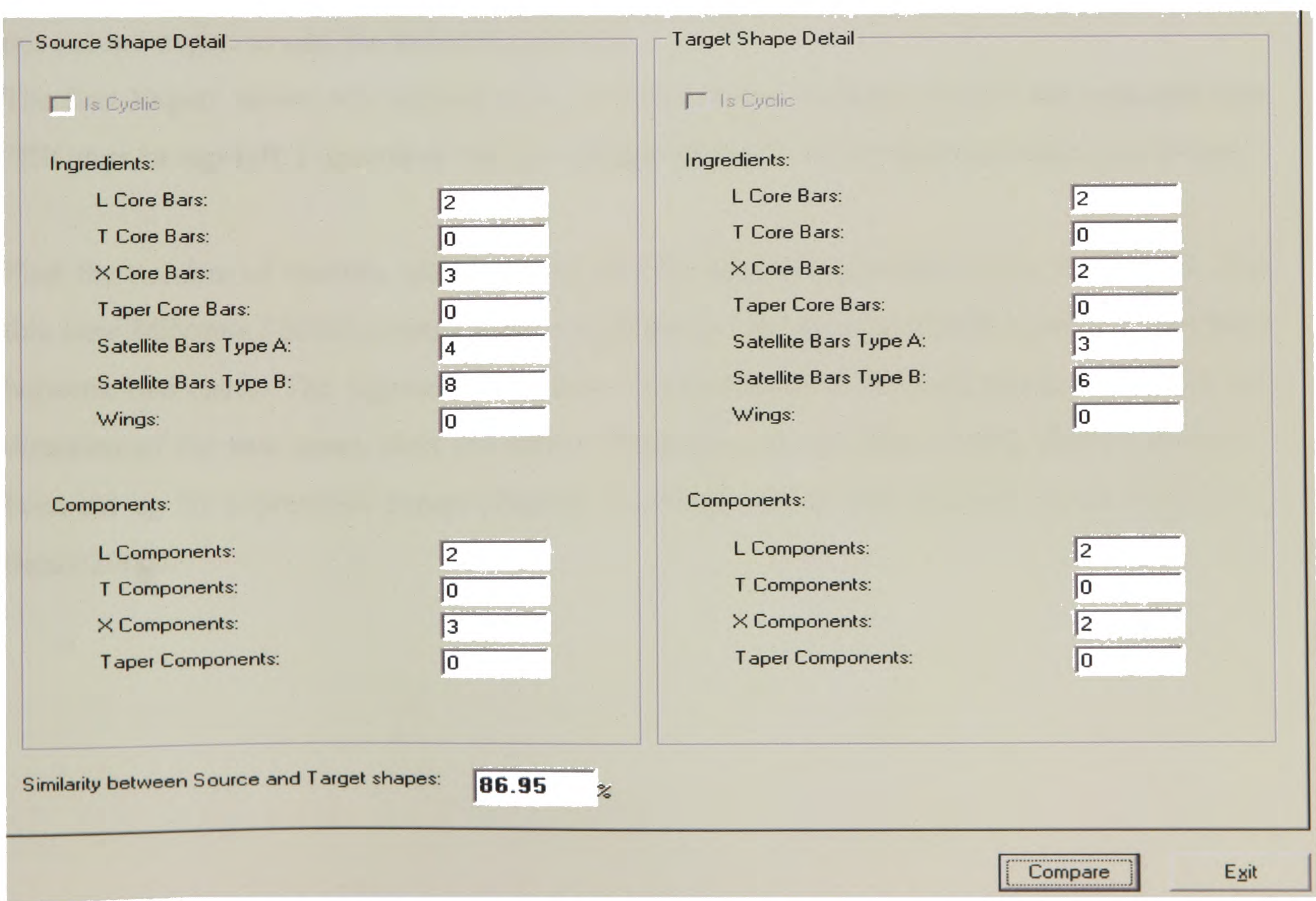


Fig.6.1b. shows an example of shape matching (analysis) between target case and source case in ShapeCBR.

6.3 Research Evaluation

A number of tests have been conducted to evaluate the efficacy and the performance of shape retrieval processing.

For the first evaluation: over 20 new cases have been given to different experts outside the UK and for the first time, this data has been introduced to the experts. These are the same cases used by Mileman for the evaluation of the CASTAID research [Mileman 2000].These new experiments or new challenges, give a new knowledge about casting design is given and shown why they used this method in engineering and how confident they are that these approaches are successful.
The data was tested with different experts in (Slemani) for 3D designs. These were collected and compared with UK experts, who investigated only the 2D shapes. This process gives an opportunity to exchange ideas and sometime, for 3D shapes, to create a new invention. However, the outcome was expected, as the results show below and sees on section 6.4.

The second evaluation concerned a number of 2D (cross section views) retrieval by existing shapes from the database [Mileman thesis: 2000]. Mileman evaluation has been reviewed by our second expert to add the following advice:
The first Expert advice was related to the feeders, chills, "general advice" **for example can "fill gaps in top-left T-junction" or" re-design advice"**) and orientation needs (see below).

First the number of feeders, chills, advice and the position of feeders were considered. For this case Mileman [2000] created three equations to calculate the feeders and chill numbers between two cases. The figures below show the positions of the feeders and chills and the direction of the two cases, then the advice depending on the shape itself. These roles have been set up by a previous expert [Preddy K: 1999] and an intermediate expert (Mileman, thesis 2000).

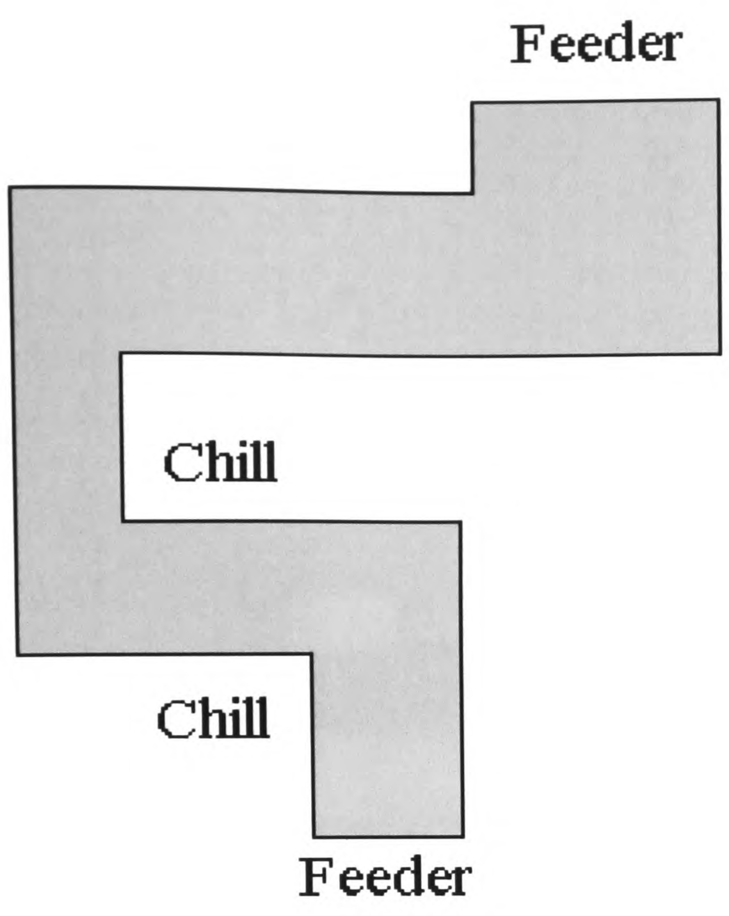


Fig.6.2 shows feeders and chills for case1.

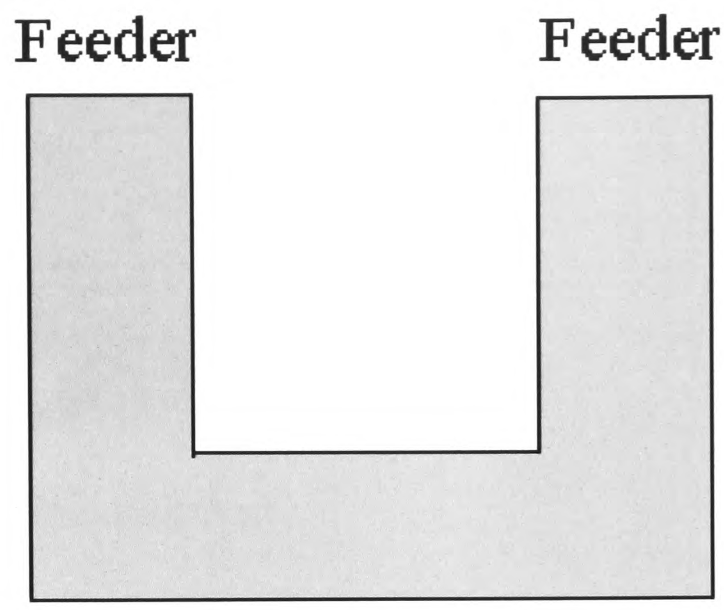


Fig.6.3 shows feeders and chills for case2.

Equation to calculate chills (1):

$$\frac{2 * (\text{Matching Feeders in Target and Retrieved})}{(\text{number of Target Feeders} + \text{Retrieved Feeders})}$$

For the Feeder calculation (case1, case2) = (2 * 1) / (2+2) * 50 = 50%
(See on Fig.6.2 and 6.3).

For solidification metal, the second expert compares the two cases (S1, S2) for casting metal by looking the number of the feeders, the chills and the orientation whether they are in correct position or not. In the two example Fig6.2, Fig.6.3 and Fig.6.4 shows the feeders and the chills for the two cases target case1 (S1) and source case2 (S2) in terms the feeder, the Chills, the orientation and the advices. The scores can be seen on the Table-6.0 is shown in the table below:

S1 (target shape)	S2 (Source shape)	Advices
2 chills	No chill	0.00
2 Feeders	2Feeders	50%
Orientation	Orientation different	0.00
Feed in right and left	Feed in right and left	100 %Advice

Table-6.0. shows the scours that have been set up by the second expert.

Fig.6.4 shows the finishing of the design and both castings have chills advice.

The equation calculates the chills scores (2):

$$\frac{2*(\text{Matching Chills in Target and Retrieved})}{(\text{number of Target Chills} + \text{Retrieved Chills})}$$

For the chill advice calculation (case1, case2) = (2 * 0) / (2+0) * 100 = 0.00% (see on Fig.6.2 and 6.3).

Fig.6.2 and Fig.6.3 shows the orientations advice for case1 and case2.

Calculation for orientation advice for Case1 and Case 2 = 100%

Preddy [Preddy, K.: 2000] the first expert set up four rules to calculate the advice (Mileman, 2000) and the second expert agreed with these rules and the advice calculation above shows that the casting example needs advice for chilling and that the advice will be 100%.

For the position rules have been set up by [Preddy, K.: 2000], that the target shape should predict the possible direction (orientation) during the shape retrieval process. The expert was taking one score of the three values: the target orientation can be correctly predicted from the retrieved case with 100% certainty §$_3$. It is not possible to predict the orientation of the target case from the retrieved case (50%) §$_2$and finally (0%) §$_1$the orientation of the target is predicted incorrectly; that is, it is the wrong way.

For the general advice, four rules have been set up by the first expert "Mileman" thesis [2000] and the four rules are:

§1 Advice in the retrieved case is not applicable to the target case.

§2 Advice in the retrieved case is applicable to the target case, but cannot be applied, because it would be inappropriate to apply the advice. The second role does not need full advice and it depends on whether a part of the target shape needs chilling advice.

§3 Advice in the retrieved case is applicable and can be applied to the target case

§4 The target case may have advice not covered by the retrieved case; that is, missing advice = 0% since there is no missing advice.

For Advice rules a function has been set up and it is $= \dfrac{\S 3}{(\S 1+\S 2+\S 3)-\S 4}$

These rules have been set up by experts and it was difficult for an intermediate expert to get this advice through academic experience. Therefore the use of an expert in AI cases is needed for the evaluation. Although the expert inherited this experience from past mistakes and from remembering the previous expert, he learned through his experiences to give advice for the important tasks in casting design. The next two Sections (6.4 and 6.5) are discussions on overseas experiments and home experiments are given.

6.4 Overseas Experiments

The purpose of these experiments was to determine how consistent casting design advice is between different experts. This was very important as there was a need for extending the evaluation of ShapeCBR to 3D shapes for which we did not have expert knowledge from the first expert. Three tests were conducted. The first 2 were used to calibrate the confidence on the casting advice and the third one was to evaluate how the CBR process for casting shapes can be used efficiently for 3D shapes.

6.4.1 Test 1

The first test was given 100 cases without advice for re-engineering or re-testing to identify the number of feeders, chills, advice and orientation, the reason being to test the previous expert and to see whether there are any differences in knowledge and engineering methods in casting design between the first expert and the second expert.

Test1	2D Shapes	Feeders	Feeders		Chills	Chills	
	Shape ID	Expert: Preddy	Expert: Aziz	% average	Expert: Preddy	Expert: Aziz	% average
1	20	2	2	1	1	3	0.5
2	21	2	2	1	2	2	1
3	25	2	2	1	1	1	1
4	60	3	3	1	2	2	1
5	62	2	3	0.8	3	3	1
6	63	3	3	1	0	2	0
7	64	2	3	0.8	2	2	1
8	94	1	3	0.5	4	4	1
9	95	1	3	0.5	0	1	0
10	100	3	2	0.8	3	3	1
			Total	**0.84**			**0.75**

Table-6.1 shows the comparison made for test 1 for casting.

Table-6.1 shows the comparison made for test 1 for casting advice provided by the two experts, Preddy and Aziz. The number of feeders and chills advised by each expert can be seen in the table. The comparison between the two expert's advices is calculated as:

$C_i = 2 * \frac{\min(A_p, A_a)}{(A_p + A_a)}, i = f, c$, where (Ap) is the advice on number of feeders (f) or chills (c) provided by the first expert and (Aa) is the equivalent advice provided by the second expert.

Orientation: The second expert advice on the orientation was an important issue in casting designs because it may affect the positions of the feeders and the chills (see on Figure 6.5). Figure 6.5 on positions: (a) shows the position of the feeder and there is no chill. For the same shape in different position (b) the number of feeder have been changed from one to two feeders further more chill is needed as well. For the position (c) the feeders and chill took different places. (See on Table-6.2 shows three casting designs example on orientation advice in terms of the feeders).

Test1	2D Shapes	Orientation	Orientation	
	Shape ID	Expert: Preddy	Expert: Aziz	% average
1	20	50	50	50
2	21	100	100	100
3	25	100	100	100
4	60	100	100	100
5	62	50	50	50
6	63	100	100	100
7	64	100	100	100
8	94	50	50	50
9	95	100	100	100
10	100	100	100	100
			Total	**85**

Table-6.2 shows the comparison made for test 1 for casting in terms of the orientation.

Table-6.2 shows the comparison made for test 1 for casting advice provided by the two experts, Preddy and Aziz. The orientation advised by each expert can be seen in the table. The comparison between the two expert's advices is calculated according to advice rules and this equation: $\frac{\S3}{(\S1+\S2+\S3)-\S4}$

The expert was taking one score of the three values: the target orientation can be correctly predicted from the retrieved case with 100% certainty $\S_3$. It is not possible to predict the orientation of the target case from the retrieved case (50%) $\S_2$and finally (0%) $\S_1$the orientation of the target is predicted incorrectly; that is, it is the wrong way.

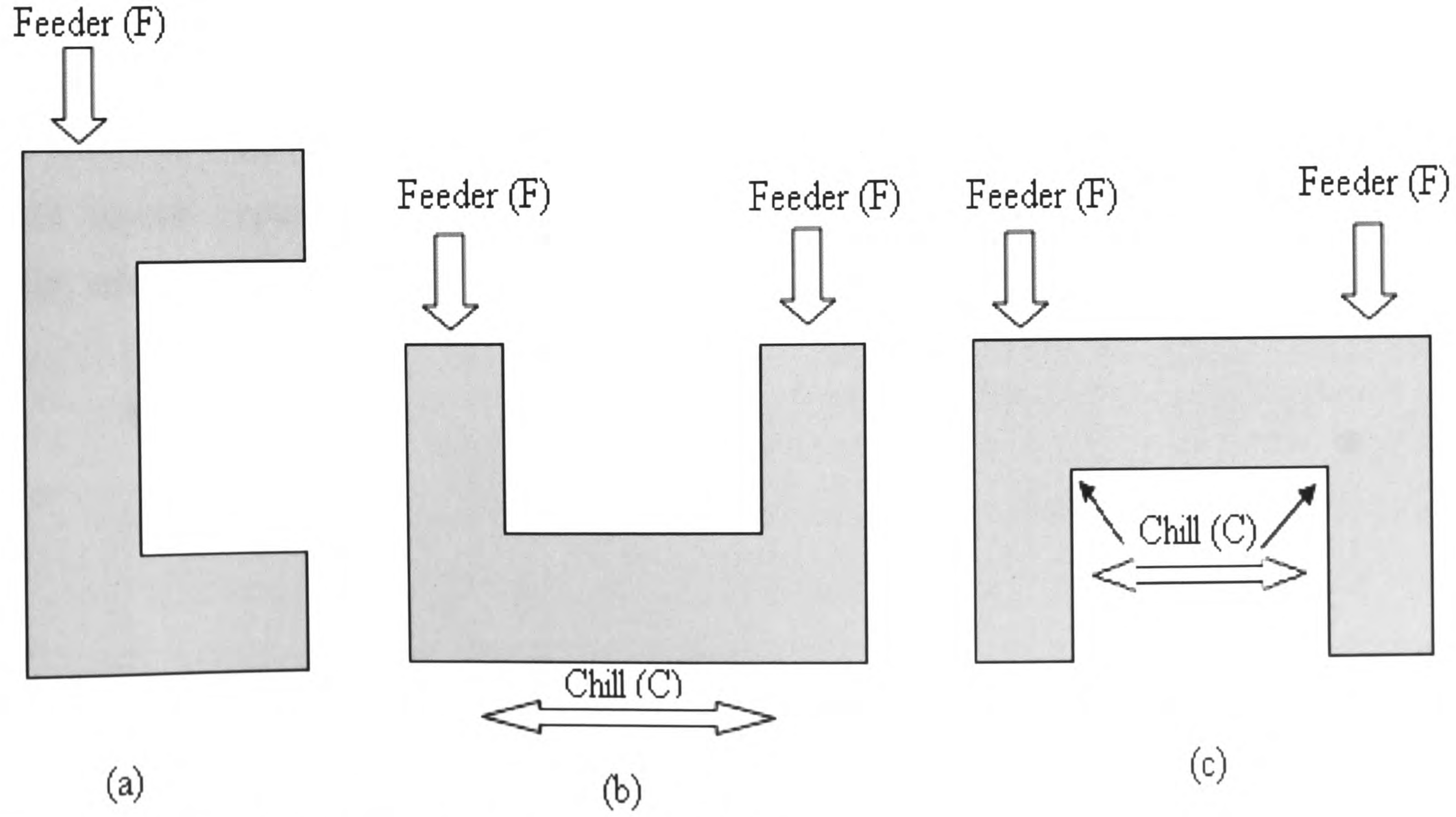


Fig.6.5 shows the orientations of the shape on three positions.

6.4.2 Test 2

The second expert re-examined a sample of the previously used 100 cases and gave marks for the casting advice on a scale of 1 (poor advice) to 5.

Test 2	2D Shapes	Average of advice scoring
	Shape-ID	Mark from Aziz
1	20	4.5
2	21	4.5
3	25	4
4	60	4
5	62	3
6	63	4
7	64	3
8	94	3
9	95	3
10	100	3.5
	Total	**3.65**

Table-6.3 shows the marks awarded by the second expert.

Table 6.3 shows the marks awarded by the second expert for the casting advice obtained from the first expert for each shape. The marks are on a scale of 1 (poor advice) to 5 (total agreement). The average mark is 3.65, which shows a good level of agreement. Comparing this table to table-6.1 it can be seen that the Second expert considers the advice of feeders as more important than the advice on chills.

6.4.3 Test 3

The third was testing the casting design for 20 new 3D cases. The 3D shapes were provided to the second expert for advice in terms of feeders, chills, orientations and other casting design advice.

3D shapes	View1	View2	View3	Overall advice	View 1	View 2	View 3	Overall	View 1 Comp	View 2 Comp	View 3 Comp	Average Comp	Total Comp
Shape-ID	Feeders	Feeders	Feeders		ShapeCBR	ShapeCBR	ShapeCBR						
Shape1	2	2		2	2	1		2	1	0.666666667		0.83333333	1
Shape2	2	1	2	2	2	1	3	3	1	1	0.8	0.93333333	0.8
Shape3	2	2		2	2	2		2	1	1		1	1
Shape4	2	1		2	1	1		1	0.666666667	1		0.83333333	0.666666667
Shape5	3	1		3	3	1		3	1	1		1	1
Shape6	2	1		2	2	1		2	1	1		1	1
Shape7	2	1		2	1	1		1	0.666666667	1		0.83333333	0.666666667
Shape8	2	1		2	2	1		2	1	1		1	1
Shape9	2	1		2	1	1		1	0.666666667	1		0.83333333	0.666666667
Shape10	3	2	2	3	2	2	1	2	0.8	1	0.666666667	0.82222222	0.8
Shape11	3	1		3	2	1		2	0.8	1		0.9	0.8
Shape12	3	1		3	2	1		2	0.8	1		0.9	0.8
Shape13	2	2		2	2	2		2	1	1		1	1
											Total	0.91452991	0.861538462

Table-6.4a shows the comparison of the advice on 3D shapes on number of feeders.

Table-64a shows the comparison of the advice on 3D shapes that the average comparisons of number of views or cross section in terms of the feeders are: **0.914529915. Total** comparison for the 3D shapes in terms of the feeders is: **0.861538462.**

3D shapes	Expert Aziz							Overall	View 1 Comp	View 2 Comp	View 3 Comp	Average Comp	Total Comp
Shape-ID	Chills (View1)	Chills (View2)	Chills (View3)	Overall advice	ShapeCBR	ShapeCBR	ShapeCBR		0	1		0.5	1
Shape1	0	3		3	1	3		3	1	1	1	1	1
Shape2	2	2	2	2	2	2	2	2	0	1		0.5	1
Shape3	0	2		2	1	2		2	1	0		0.5	1
Shape4	1	0		1	1	1		1	0.8	0.666666667		0.73333333	0.8
Shape5	3	1		3	2	2		2	1	1		1	1
Shape6	2	0		2	2	0		2	1	1		1	1
Shape7	2	0		2	2	0		2	1	0		0.5	1
Shape8	1	0		1	1	1		1	1	0		0.5	1
Shape9	2	0		2	2	1		2	1	1	1	1	1
Shape10	2	2	2	2	2	2	2	2	1	1		1	1
Shape11	2	0		2	2	0		2	1	0		0.5	1
Shape12	3	0		3	3	1		3	1	1		1	1
Shape13	0	0		0	0	0		0			**Total**	**0.74871795**	**0.984615385**

Table-6.4b shows the comparison of the advice on 3D shapes on number of chills.

Table-6.4b shows the comparison of the advice on 3D shapes between the second expert and ShapeCBR. We can see that that the average score in the comparison over a number of views or cross sections in terms of the chills is **0.74871795** and comparison of the overall advice for the 3D shape in terms of the chills is: **0.984615385.**

Tables (6.4a) and (6.4b) shows the comparison of the advice on 3D shapes on number of feeders obtained from our second expert to the advice obtained by the ShapeCBR system. The advice aggregates from advice on the various 2d views change as above.

6.5 Home (UK) Experiments

This section, an algorithm is presented that has been created to help the evaluation of the system. An individual weight has been set up for each feature and advice, such as Components, MCS, Leaves and Cycles. Section 6.5.1 describes the algorithms steps and figure 6.6 shows the Evaluation Flow chart to find the nearest case.6.5.1.

6.5.1 Evaluation Steps (Algorithm)

Figure 6.6 describes the steps of an algorithm to find the nearest cases, (X-top), (X%-top) case, combination advice and the best advice rate for the shape retrieval process.

The algorithm consists of four loops:

The first step initials with zero value for the first weight (see on table-6.0). The first run for the loop searches for the first weight which it represents by cycling weights with their nearest cases, then the output prints (0001).

The first (0) represents components advice the second (0) the common maximum sub-graph, the third (0) represents leaf advice and the last (1) gives the result of the Cycle's best weights advice.

The second loop output prints 0010 of which the **(1)** represents the leaves advice weights with their nearest cases. The third loop output prints 0100 of which the **(1)** represents MCS advice weights with their nearest cases. For Fourth loops output prints numbers between (1) and (0) such as: (1-0-0-0) which the (1) represents Components advice weights with their nearest-cases. The tests results can be seen in the Tables (6-1, 6-2 6-3, and 6-4a and table 6-4b). The next Section introduces the final experiments which were made up of five tests to ShapeCBR system to improve the efficiency and performance of the system and CBR method itself.

Evaluation Steps Flow chart

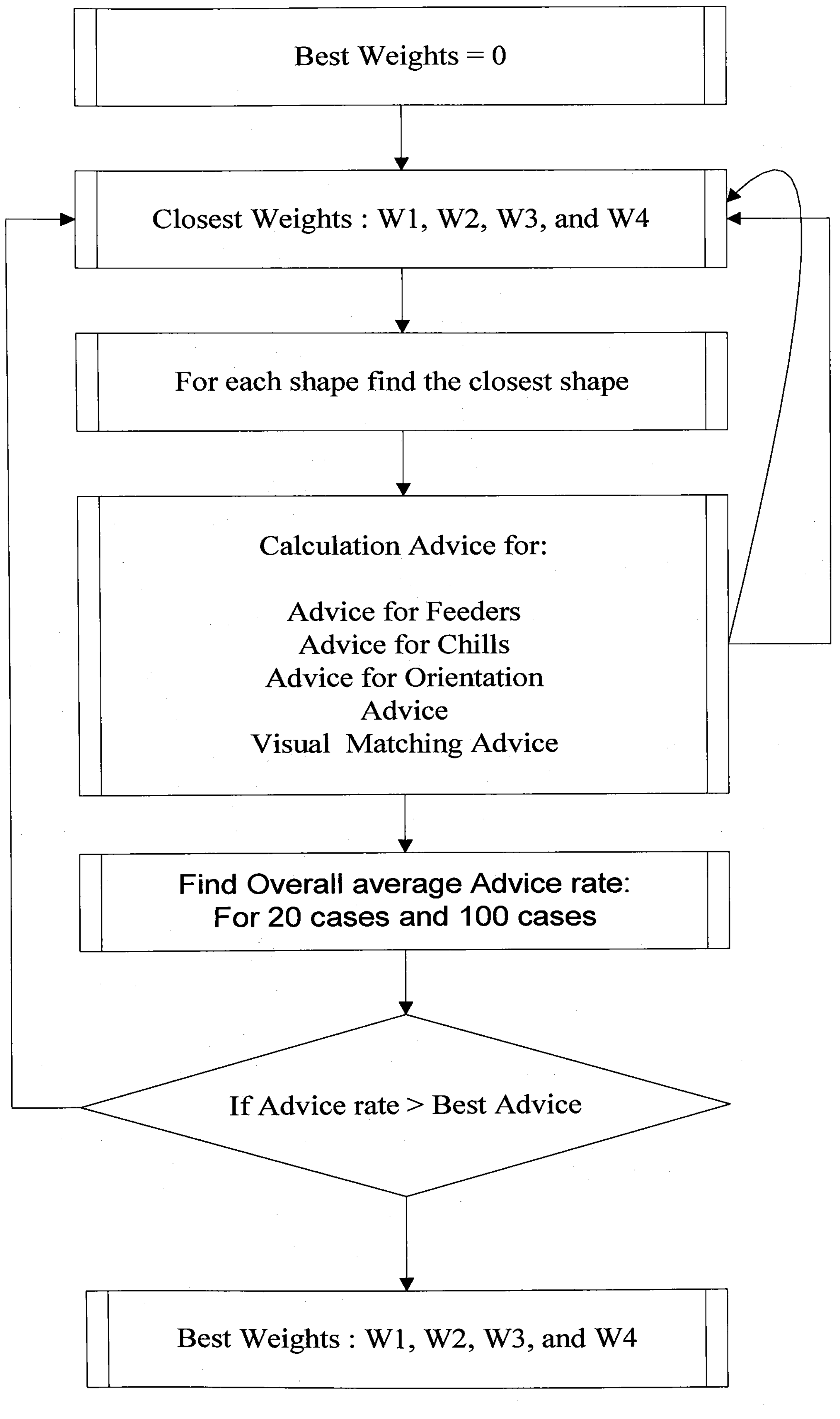


Fig. 6.6 shows the Evaluation Flow Chart to find the closes case with their advice in terms of Feeders, Chills, Orientation and visual matching advice.

6.5.2 Final Experiments

A case base of 100 methoded cases has been designed [Mileman thesis 2000] and will be used. These designs have been taken from the domain of rotationally symmetric shapes. All of the designs have been methoded by two experienced foundry engineers and these cases have been used for testing against 20 target cases.
We had 100 cases which had been given to a casting domain expert. The casting advice given was used to evaluate the suitability of the CBR system. 20 cases were used as targets. For each of the targets, the K-NN (K =1, 5) cases from the rest of the case base were retrieved and the advice given was compared to the expert advice given for that target case. The comparison using equation above (Feeders and Chills equations) were used to calculate the efficiency of that retrieval.

The tests initialised by testing the existing data for previous cases and the results were that over 300 errors were have been corrected from the previous research of human expert advice in terms of advice feeder's number, chills, general advice and orientation advice. First test was reviewing Mileman cases [Mileman: 2000].
The first loop for the algorithm, set up at zero advice for w1, representing weighting value for component type, w2 is representing weighting for MCS, w3 representing the number of leaves and finally w4 representing the number of cycles.

A prototype system has been developed for to test the ShapeCBR to improve the efficiency and performance of the system and CBR method itself.

The initial step was to start with zero advice, then the algorithm searches for the closest cases, for individual rate advice for (w1, w2, w3 and w4), best combination advice and finally the overall advice rating.

Arbitrary 3D shapes have been investigated by looking at the overall similarity metrics between arbitrary 3D shapes. These shapes can be treated from one view to a number of cross-sections or views. Often these shapes can provide valuable identifiers to enable accurate retrieval.

The previous section described the steps of the evaluation to help the system to evaluate the first test, which involves comparing 20 target cases with 100 cases from a large data base. This section presents the improvements for all ideas that have been thought to have attempted answer the questions that have been posed by this research. To prove the research attempts, three tests have been set up as follows.

6.5.2.1 First Test

The first test was to re-evaluate the past 20 (2D) target cases against the 100 (2D) cases (Mileman: 2000) from case-based knowledge then the final result to be examined by a second expert advice. The reason behind re-testing pervious cases was to be sure if there were any missing advices or any calculation errors. Test results showed that was over 300 cases were missing and more than 100 cases out of 10000 cases have been corrected. However, the testing results for re-testing are shown in Table-6.4.

Metrics	weights	weights	weights	weights	The overall advice for: Feeders, Chills , Advice and Orientation
Component number	1	0	0	0	***69.875***
MCS	0	1	0	0	89.20
Leaves	0	0	1	0	74.375
Cycle	0	0	0	1	71.875
Equal weights	1	1	1	1	86.50
Best combination	0.4	0.2	0.2	0.2	90.75%

Table-6.5 shows the first test results for 2D shapes with past metrics. (1) Represents predicted advice (0) represents none advice.

Table-6.5 the columns are representing the metrics, the weights and the advices.

The first column to the left presents the metrics, the second, third, fourth and the fifth columns display the weight settings for the shape features and the **(1)** represents the result of the metrics. These data were generated by the system evaluation to test the ShapeCBR system efficiency and performance. The last column shows overall advice the results of a human expert that have been compared with the first test and advice contains the Feeders, the Chills, the Advice and the Orientation.

6.5.2.2 Second Test

Mileman (2000) has discriminated that the size of components has been ignored and has not been answered. This research pointed out from the very beginning that the data dimensions for the shapes play an important role in increasing the perfectas for the shape retrieval. So the values of past componentisation have been changed with this research by adding the geometrical data for each shape to see how the idea affects and produce perfects shape retrieval by a given target. We thought and even the research expert advised that the number of feeders, chills and advice depends on the size of the shapes. The system runs for the second test same data with changes (data dimensional), by replacing the past component records with current records and the test results we thought and expected the perfect shape matching was retrieved by adding the properties of the shape. The second test is: testing 20 (3D) cases against 100 cases and using previous advice, because the second expert and first expert have the same experience and same methoding in casting design metal. See Table-6.6.

Metrics	weights	weights	weights	weights	The overall advice for: Feeders, Chills , Advice and Orientation
Component types and number	1	0	0	0	***83.45***
MCS	0	1	0	0	89.20
Leaves	0	0	1	0	74.375
Cycle	0	0	0	1	71.875
Equal weights	1	1	1	1	87.41124
Best combination	0.6	0.4	0.0	0.0	91.325

Table-6.6 show the second test results for 2D shapes with new metric component types.
(1) Represents advice given and (0) represents none advice.

Table-6.6 shows the columns representing the metric, the weights and the advice for feeders, chills, advice and orientation. The first column to the left presents the metrics. The second, third, fourth and the fifth columns display the weight settings for the features of the shape and the (1) represents the result of the metrics. These data are generated by the system evaluation to test ShapeCBR system efficiency.

The last column shows the results of a human expert that have been compared with the first test and advice contains the Feeders, the Chills, the Advice and the Orientation.

6.5.2.3 Third Test

Over 20 new views that have been derived from 3D shapes have been added into the large database knowledge for new case retrieval. In this test there were 20 first views from 3D where each view represents the first section from each individual 3D shape (the 3D shape could be have more than one views), which has been prepared for the first run to search for the weightings advice for each metrics such as component types, MCS, leaves and cycle) then in the same run the system was looking for the nearest similar case for individual metric and the best advice combination. Previous advice has been applied to this test and the reason for that as has been discussed at the beginning of this chapter, is that the current expert and past expert using the same engineering methodology in casting design. The results for the first 20 views divide by 20 cases from case base to produce the overall similarity for the first part. For more detail see chapter 5 on overall similarities. The second run for the second 20 views goes through the same process and final results will be divided by 20 second views to produce overall similarity for second test. Finally, the results for overall similarity between two 3D shapes are given in Table-6.7. The combination of the two tables produces the final results as give in Table below.

Metrics	Weights	Weights	Weights	Weights	The overall advice for the Feeders, Chills, Advice and Orientation Number (1) represents the predicted case.
Component types and number	1	0	0	0	**64.331**
MCS	0	1	0	0	**67.567**
Leaves	0	0	1	0	57.698
Cycle	0	0	0	1	52.451
Equal weights	1	1	1	1	62.124

Table-6.7 shows the third test results. (1) Represents given advice and (0) Represents None advice.

The first column to the left presents the metrics. The second, third, fourth and the fifth columns displays the weight settings for the shape features and the (1) represents the result of the metrics. The data in Table 6-6 have been generated by the system evaluation to test the ShapeCBR system for efficiency performance. The last column shows the results of a human expert that have been compared with the first test and advice contains the Feeders, the Chills, the Advice and the Orientation.

Table (6.5, 6.6 and 6.7 presents the outcome for the three tests and the results has been generated through the evaluation algorithm. The results have shown the best advice for the feeders, the chills, the chill advice and the orientation. The algorithm searches for the closest cases and the best combination of advice. The second test shows the optimal weight for individual features.

The overall evaluation tests have been shown that adding (new metrics types component) the dimensional data into the individual components gives better results.

We thought from the very beginning that aspect ratio calculation for size of component types plays an important role in quality of casting designs for shape retrieval.

6.5.2.4 Evaluation Procedures for Seeking the Best (X) % Target and the Top Five Cases of 100 Case Bases.

The previous three evaluations have shown only the top or the first best case target from 100 cases. In this section we investigate how we can improve the search procedure for X% by repeating the second test. This involves testing twenty 3D cases against one hundred 2D cases. To do this, we were able to use the pervious advice, because both the second and the first expert have the same experience and follow the same method in casting design. The results are shown in Table-6.8 and Figure 6.7.

Metrics	weights	weights	Weights	weights	Advice for the 5 %: Feeders, Chills , Advice and Orientation Number (1) represents the predicted case.
Component types	1	0	0	0	Orientation advice : .6,.4,0,0,88.9993 .8,0,0,.2,82.23546 .8,0,.2,0,82.9959 .8,.2,0,0,85.25321 1,0,0,0,**82.39706**
MCS	0	1	0	0	Feeder advice: 0,1,0,0,**86.08076** 2,0,0,.8,80.82803 .2,0,.2,.6,81.25401 .2,0,.4,.4,80.94425 .2,0,.6,.2,80.94425
Leaves	0	0	1	0	Chill advice : 0,0,1,0,**71.00282** 0,.2,0,.8,87.9368 0,.2,.2,.6,87.39653 0,.2,.4,.4,86.31297 0,.2,.6,.2,86.31297
Cycle	0	0	0	1	Advice : 0,0,0,1,**69.35841** 0,0,.2,.8,71.03661 0,0,.4,.6,71.03661 0,0,.6,.4,71.03661 0,0,.8,.2,71.03661
Overall best advice	1	1	1	1	1,1,1,1,90.29266

Table-6.8 shows the test results the top (X) percentage. (1 true) representing predict advice (0 false) represents no advice.

Tables (6.8 and 6.9) and the Figures (6.7) and (6.8) shows that the columns representing the metrics, the weights and the advices.

The first column to the left presents the metrics. The second, third, fourth and the fifth columns display the weight settings for shape features and the (1 true) represent the predicted result of the metrics. These data are generated by the system evaluation to test ShapeCBR systems efficiency. The last column shows the results of human expert that have been compared with the first test and advice contains, the Feeders, the Chills, the Advice and the Orientation.

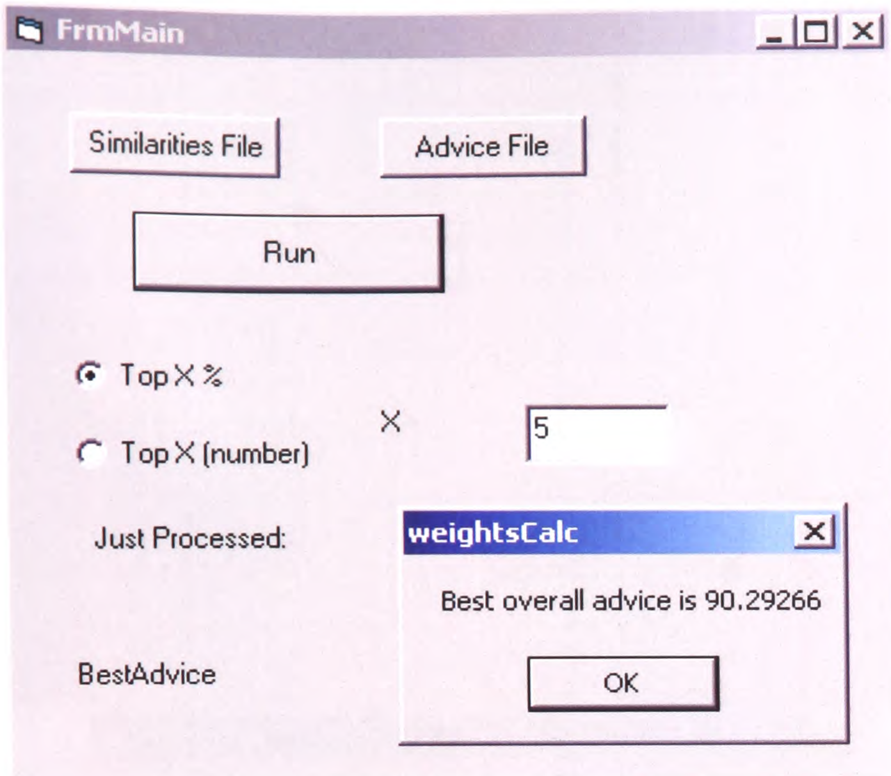


Fig.6.7 shows the test results for the top (X) percentage cases.

Metrics	weights	weights	weights	weights	Advice for five top cases Feeders, Chills , Advice and orientation
Component types	1	0	0	0	Orientation advice : .6,.4,0,0 ,85.88389 .8,0,0,.2,81.59202 .8,0,.2,0 ,82.40306 .8,.2,0,0 ,83.94219 1,0,0,0,**81.66008**
MCS	0	1	0	0	Feeder advice: 0,1,0,0,**86.08076** .2,0,0,.8,80.82803 .2,0,.2,.6,81.25401 .2,0,.4,.4,80.94425 .2,0,.6,.2,80.94425
Leaves	0	0	1	0	Chill advice : 0,0,1,0,**71.00282** 0,.2,0,.8,83.95559 0,.2,.2,.6,83.77381 0,.2,.4,.4,81.74648 0,.2,.6,.2,81.71477
Cycle	0	0	0	1	Advice : 0,0,0,1,**69.35841** 0,0,.2,.8,71.03661 0,0,.4,.6,71.03661 0,0,.6,.4,71.03661 0,0,.8,.2,71.03661
Best combination	1	1	1	1	Overall best advice : 87.35036

Table-6.9 shows the test results for the top five cases. (1 true) represents predicted advice. (0 false) represents no advice.

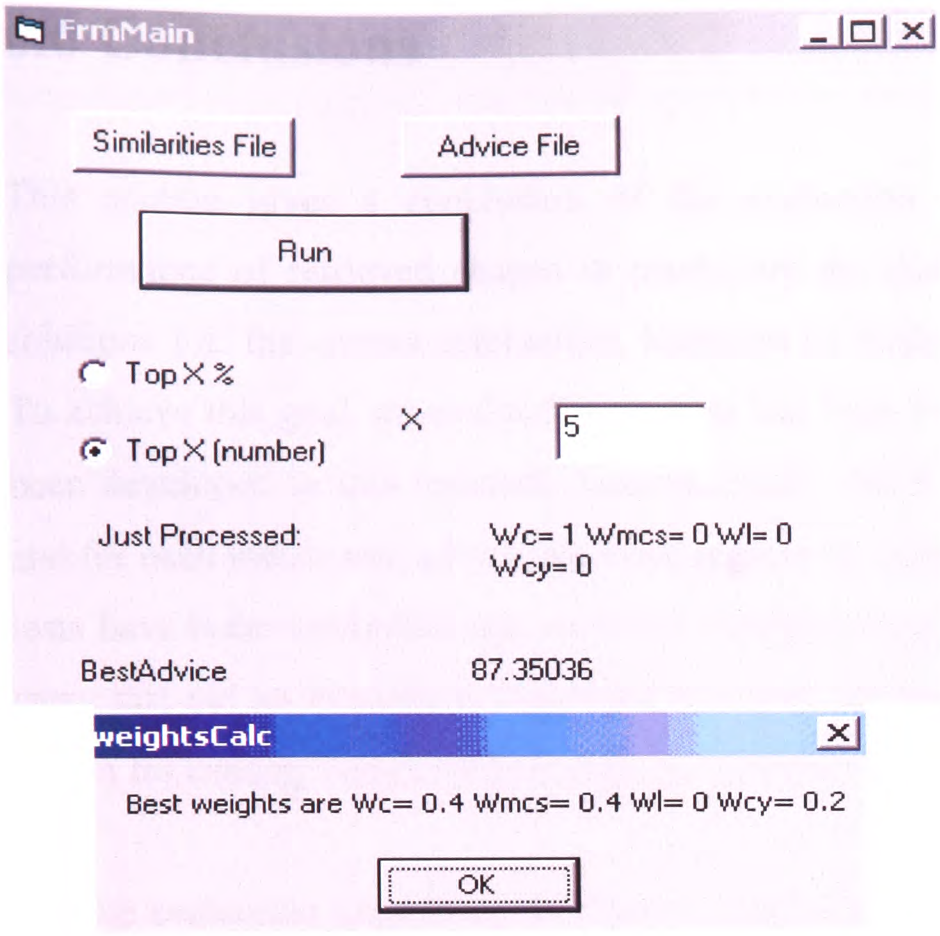


Fig.6.8shows the test results for the top (X) cases.

The first column in Table-6.9 shows weight settings for shape features: the component types, MSC, leaves and the cycle metric. The column 2, 3, 4 and 5 shows value (0 and 1) the ones represent the advice results of the (X) % for individual metric type. The column number 6 shows the final outcome on the advice for (X) % in terms of the Feeder advice, the Chill advice, the common advice and the orientation advice.

Finally, the test of overall of the testing evaluation of the performance of shape retrieval metrics against the human visual match (judgment) was examined. The main conclusion for these tests was that the size of the maximum common sub graph, and the component types with their number were the most strong and important metrics. The cycle metric was found to have little affected for shape matching and in shape retrieval and the leaves metric was found to be insignificant for shape retrieval.

6.6 Conclusions

This section gives a conclusion of the evaluation chapter and is focused on the performance of retrieved shapes in predicting the design of a given test case to give solutions for: the correct orientation, locations of feeders and chills, and general advice. To achieve this goal, an evaluation process has been set up to test all metrics which has been developed in this research; Leaves, cyclic, MCS and types of component metrics and for each metric and advice has been applied by human experts. For this purpose five tests have been conducted and each test compared against expert judgments. In order to carry this out an evaluation algorithm has been developed with number of equations to search for casting design advices such as; feeders, chills, orientations and general advice.

For the evaluation algorithm, four loops (loop inside loops) have been created and each loop identifies specific advice. The first loop identifies the best advice for cyclic metrics. The second equation is to find the first nearest neighbour case.
Another equation has been created for the specific task of identifying a bunch of closest cases or (X) percentage for target cases.

The evaluation system gives the best advice and nearest neighbour cases in the first, second and third tests. For example when a question is asked in Google the agent find the first case for you, but the human brain judgments works differently and suspiciously, and maybe finds other relevant cases underneath the first case. But we are discriminating that the top one might not be the only relevant case. Maybe there are more relevant cases; therefore two equations have been created to retrieve the five top target cases and the (X) % target cases from all source cases.

Retrieval of geometric shapes, the most challenging aspect of content-based shape retrieval has been carried out in this research. The method for evaluating a shape matching technique is in the first part: comparison against existing data that has been provided by (Mileman: 2000) and the second part replaces component metrics with current Component Type Metrics (CTM). In this part, aspect ratio has been added to-each component types to calculate the size of component types. The size components have been ignored by Mileman [Mileman: 2000] such as:

- Bar
- Taper
- L-shaped
- T-shaped
- X-shaped

The third test was comparing 3D shape against 100 cases and using the first expert knowledge and evaluating the 3D shapes through the ShapeCBR system and the second expert.

Leaves are important to be included, because leaves have own their role in shape creation, but (MCS) and (CTM) are of significant importance, and their degree of importance is higher than the cycles and the leave metrics. The reason is that the components determine the structure of the shape, and MCS is a combination between connectors and structures and therefore has a higher degree of importance than the other metrics.

The overall performance of the case base with a set of optimised weights was in the range of (86.10 %) to (100%). Following this, the performance for orientation, feeder, chill and general advice was examined. Orientation performance increases as the case base increases. Feeder performance improves as the case base increases in size, although from approximately sixty cases onwards there is no significant improvement in feeder performance; thus there is not much to be gained by adding more cases. The chill performance of (71.04 %) was an unsatisfactory, and the main reason for this low performance is associated with chill-feeder substitution, where feeders can be used instead of chills. The performance of advice showed that it increases as the case base increases; however there is not much of a performance increase after fifty cases and this suggests that it is more practical to enter key cases with the most significant advice.

Finally, the performance of shape retrieval metrics against the human visual match was examined. The main conclusion was that the size of the component types, , maximum common subgraph, and the number of components were the most important metrics. The cycle metric was found to have little effect on shape retrieval. Also, the leaf metric was found to be insignificant for the purposes of shape retrieval.

The final conclusion the top (5%) nearest cases out of 100 cases is show in Figure 6.9 and Table 6-10.

Predicted cases	MCS	Component	Leaves	Cycle	Best combination advice in terms of feeders, chills, orientation and general advice is: 87.35
First predicted case.	86.081	85.884	71.003	69.36	
Second predicted case.	80.828	81.592	83.956	71.04	
Third predicted case.	81.254	82.4	83.774	71.04	
Fourth predicted case	80.94	83.94	81.746	71.04	
Fifth predicted case.	80.94	81.66	81.715	71.04	

Table-6.10 shows the test results for the top (X) nearest cases.

Table-6.8 shows the test results for the top (X) nearest cases. The first column in the Table 6-8 is Predicted cases (first, second, third and fourth) the features of the shape in terms of the feeders, chills, orientation and general advice and the last column represent the best combination advice. The cells represent the advice results of the (X) % for individual metric type. The column number 6 shows the final outcome on the advice for the (X) % in terms of the Feeder advice, the Chill advice, the combination advice and the orientation advice.

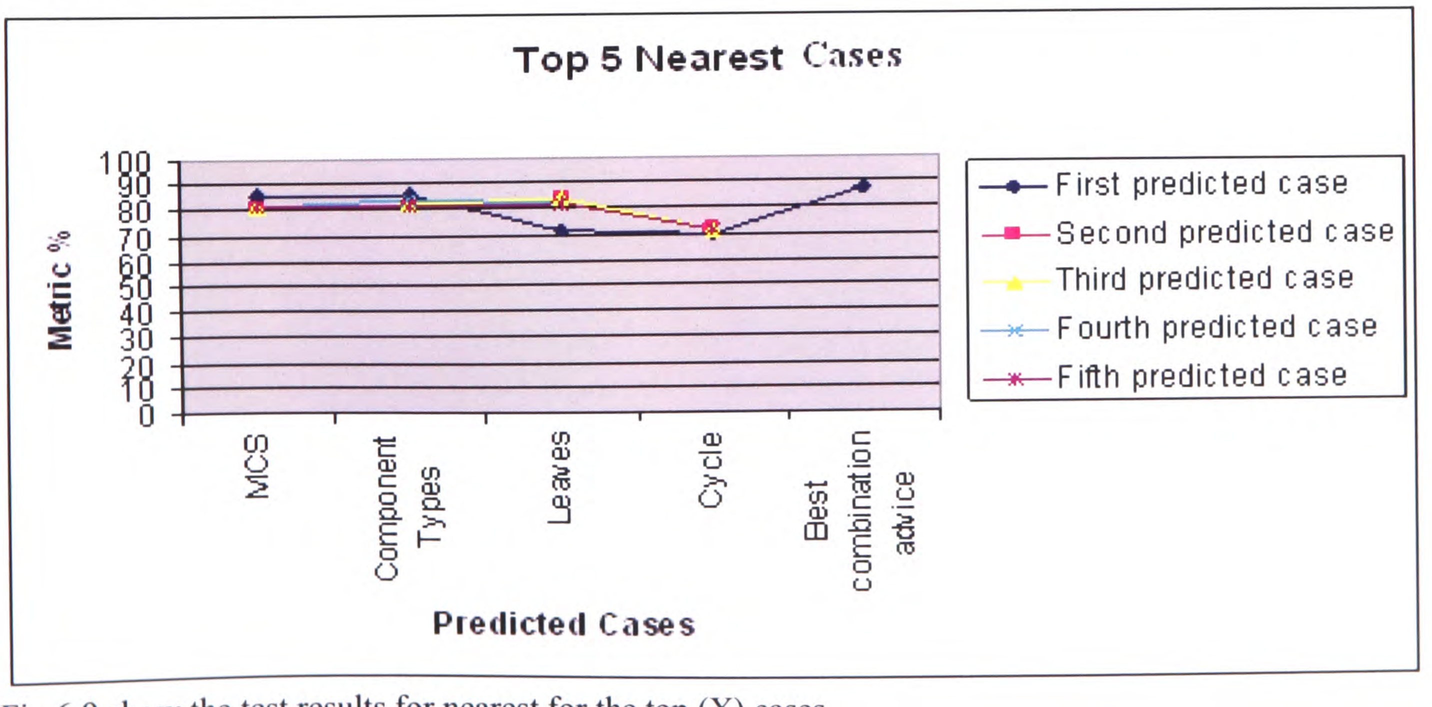


Fig.6.9 show the test results for nearest for the top (X) cases.

First column in the Figure 6.9 shows in above graph and each line coloured represent the advice result of the (X) % for individual type metrics. for the shape features: MCS, component types, leaves, cycle metric and the last black coloured represent the best combination advice. The next chapter presents thesis conclusions and future work.

Chapter 7

Conclusions and Future Work

This chapter presents the summary of this thesis, the contributions of this thesis and proposes some directions for further work that could investigate further ways to improve the efficiency of CBR systems for reuse of casting design advice.

7.0 Introduction

The discussion in this chapter begins with a brief review of each chapter findings and is followed by an appraisal of the research and the contributions. This focuses on the original aims, the achievements and failings, together with an assessment of what might has been done differently. Finally, there is a discussion of the extension of the methodology to the use of particular components such as "Taper Component Types" (TCT) taking into consideration the orientation of casting designs.

There are several complex reasoning processes that relate to human case-based casting designs. For years, casting design engineers have been looking for a simple and intuitive mechanisms to evaluate, assess and re-use designs. This research has shown that a CBR system is capable to provide an efficient mechanism for the effective re-use of casting design knowledge and that the associated CBR process can be automated.

The ShapeCBR system was implemented at University of Greenwich by the author to support the CBR process and provide an evaluation tool for the thesis of this research. It automates key processes such as componentisation process and shape matching and has shown to work efficiently when compared to a human expert. The system automatically decomposes the shapes into connected component types and classifies the shapes into the predefined component types and the resulted cases are stored as case-based knowledge for reuse during the shape retrieval process.

The conclusion drawn from this thesis, is that it is possible to automate the shape retrieval of past casting design knowledge, using efficient similarity metrics based on a decomposed graph representation of casting shapes and assist casting design practitioners during the early design stages of a casting design project.

For evaluation process a set of tests have been devised and conducted to evaluate the main approach, algorithms and metrics adopted by this research.

7.1 Summary

The thesis is made up of seven chapters, which address quite individual aspects of the problem. To summarise, the following was discussed and established in each of the main chapter.

Chapter 1 investigated the research questions and their objectives. The primary objective of this research was to prove the feasibility of developing an algorithm that would help to decompose 3D shapes into disjoint substantial components, with the resulted stored in case base for later retrieval for shape comparison.
In this chapter, 1 main question of similarity metrics between three dimensional shapes for shape retrieval problems have been posed in order to attempt answer this question was necessary to answer two subsidiary questions; componentisation question and similarity metric question. Both have been addressed in chapters 3, 4 and 5.

Chapter 2 extends the work chapter 1 and with the aim is to establish the domain of the research from the Knowledge Management, Computer Aided Design (CAD), Knowledge based design, Case-Based Reasoning (CBR) and each key was reviewed in details and reason to identify possible solution in casting design problem. As a result of this, the main idea was to investigate how to unify a solution to deal with research problems.

Chapter 3 investigated one of the primary objectives of the thesis, the componentisation question "automating decomposition process". In order to facilitate a possible solution for this problem, an algorithm has been designed to decompose 3D shapes into a number of 2D cross-sections or (views).The views represented 3D shapes. The elements of the decomposition process have been stored into the case-based knowledge to be prepare for the next process; the classification process.

Chapter 4 discusses the two algorithms that have been designed to automate the shape classification by retrieving the shape decomposition results from case-based database and classifying then into a set of generic components types.
In order to attempt to solve the classification process two algorithms have been designed and they are: **"Full scan" method** to identify the structure components and regions (L, T, X and tapers) and the second method is the "**Semi scan**" method to identify the elements such as bar types (A) and (B). The results of classification have been stored into the case-based knowledge for reusing in the future for casting design problems.

The fifth chapter deals with the similarity metrics for 3D shapes. Several functions have been created to achieve both individual similarity and population similarity (in this case looking for overall similarity between two 3D shapes) and also to describe the actual problem with their final solutions for shape matching. In this chapter a new metric "Component Type Metric" (CTM) has been created for retrieving useful and knowledgeable cases from the case base.

The fifth chapter deals with the similarity metrics for 3D shapes. Several equations have been created to achieve both individual similarity and population similarity (in this case looking for overall similarity between two 3D shapes) and also to describe the actual problem with their final solutions for shape matching in this chapter a new metric "Component Type Metric" (CTM) has been created for retrieving useful and knowledgeable cases from case base knowledge.

Chapter 6 is evaluates the ShapeCBR system for CBR itself in general This chapter is introduces the achievements of the research through testing a large number of cases to prove the efficacy and performance of ShapeCBR system. The research based on experimental results by testing over 100 3D shapes and 20 additional artificial shapes from 3D arbitrary types shape. In this chapter the first evaluation was based on re-evaluating previous research results from the Mileman thesis [Mileman, 2000] by examining his evaluation results with the second expert and for this aim a score of 1-5 has been sat up by the expert. And secondly evaluating 2D shapes by replacing the pervious recodes with a new metrics the "Component Type Metric" (CTM) to see the performance and efficiency for the new metric and the results compared with previous research [Mileman: 2000].

The third evaluation is the test for overall similarity metrics between 3D shapes (a number of views representing 3D shape) by slicing the 3D shape into different views (CAD application) and comparing the views for the two 3D shape in terms of the number of components with their types, number of cycles, number of leaves and orientation, then the results of this evaluation have been tested against human expert in terms of the chills number with their advice, the feeders with their advice, the best combinations advice and the overall advice

The fourth tests searches for a percentage or five nearest cases. For this test we reviewed second test 20 target cases examined against 100 cases from the database.
For this test an efficient algorithm was designed. The algorithm was made up three equations; the first one is searching for the percentage and the second one searching for the top five cases, and the other equation searching for the advice for individual case. The idea was that a human expert does not necessarily search for only the top case but some times the second case will also be relevant, or even the third may be relevant.

Chapter 7 presents the conclusion and future work of the thesis. It starts with a brief summary of the main findings (contributions), discussions about future enhancements and additional works followed by the thesis report, concluding with numbers of Appendices. These cover papers.

7.2 Research Outcomes

This section presents the research achievements based on the research problems defined. The section 7.2.1 presents a brief review of research goals and section 7.2.2 introduce the research achievements.

7.2.1 Review of the Goals

The main goal was to attempt to prove the feasibility of automatically shape componentisations into connected generic components that can help a casting designer to store the products of decomposition and classification into case-based knowledge, for retrieving automatically by a given target design. This goal was divided into four primary subsidiary aims:

(a)

The first aim was the attempt to automatically decompose 2D shapes into a set of connected generic components.

(b)

The second aim was the attempt to automatically classify 2D shapes into a set of connected generic components in identifiable component types and elements such as:

Bar.

L_shaped.

T_shape.

X_shaped.

Bar.

Taper.

(c)

The third aim was to attempt to automatically decompose 3D shapes into a set of substantially different 2D cross-sections or (views) that can be used to retrieve useful casting knowledge.

(d)

The fourth aim was to see if there is any possibility of achieving a useful casting knowledge about casting shapes to be retrieved automatically from a CBR system that stores the componentised views of the shapes.

The main goal was focussed on similarity metrics between 3D shapes. For this challenge several algorithms have been designed that could produce a competent and efficient way to retrieve useful casting design automatically from case-based knowledge from the ShapeCBR System.

7.2.2 Achievements

Contributions

This thesis has focussed on the study of similarity metrics between 3D shapes in casting designs problem based on case-based reasoning (CBR).

While investigating that, the main contribution was the concept and formulation of this automating shape decomposition, classification, shape matching process and its evaluation. We can define this result in a number of contributions; both in the specific field of similarity metric between 3D shapes for CBR and to CBR itself in general while investigating the questions posed.

7.2.2.1 The main contributions of the thesis work are:

1- Automating shape decomposition process

A new algorithm has been designed and tested to automate shape processing in a competent and efficient way for decomposing shapes into a set of rectangle and triangle shapes. The decomposition process method is based on identification of the **Hotspot** and horizontal and vertical **projection techniques** to identify the internal geometry of the shape.

2- Automating shape classification process

A new algorithm has been designed and tested to automate shape processing in a competent and efficient way for classification shapes into a set of connected generic components such as L-shaped, T-shaped, X-shaped, Taper types and element types such as (Bar Types (A) and (B). The classification process method based on search matching. Full-scan and Semi-scan techniques have been used to classify the components into generic, identifiable component types.

3- A new metric has been created, called "Component Type Metric" (CTM).

This takes into account by adding the geometrical features and properties of each single type component. The similarity metrics between components type have been extended using methods that take into account the geometrical features and proportions of each single shape component. The improved similarity metrics have been shown to give better results by matching and retrieving better expert casting advice.

4-An efficient equation to calculate the overall similarity metric for 3D shapes

Finally, an efficient equation has been created for overall similarity metrics for 3D rotational symmetric shapes using graphical representations to matching the shapes. Overall similarity metrics between arbitrary 3D shapes can be defined and used to retrieve relevant casting advice. These shapes can be treated from one view to the number of cross-sections (views). Often these shapes can provide valuable identifiers to enable accurate retrieval. Chapter 6 on evaluation discussed this in details.

The advantages of automated Componentisation process over manual process

In the first place, we believe that the contribution of automating the process of decomposition and classification 3D geometrical shapes using a graphical representation allow for the efficient retrieval of similar shapes for Case -Based Reasoning and thus improve the reuse of relevant casting design knowledge to gain some advantages, such as those shown Table-7.0.

Automated for decomposition process, classification process and shape matching process for 3D geometric shapes could offer a competitive advantage in terms of **reduced cost,**

reduced time to market, and **of specialised engineers for further design and development**. Table-7.0 briefly summaries the advantages of automated testing from the right-hand column over manual testing from the left column over automating testing.

Manual testing	Automating testing
- Time expensive. - Cost expensive. - Repetitive task. - Less "attractive" role for engineers. - None. - None. - None. - None - None.	- Reduced time to market. - Reduced cost, because less labour intensive. - Potential to increase quality by covering more. - "Freeing up" of specialised engineers for further design and development. - Anything from 1 week to 1 month of manual testing can be run in 1 night.

Table-7.0 shows the advantages of automated testing over manual testing.

The developed application ShapeCBR uses test case knowledge, containing shapes for various designs. The tests showed that even by using a minimum set of comparison indices, similarities in design could be identified. This allowed for similar designs to be ranked and selected and presented to the user for partial or full re-use in a new design problem. The retrieval process is much faster, more reliable and it has a higher accuracy rate than the one that would be used in a paper-based system.

7.2.2.2 Previous research contribution

Mileman [Mileman Dec. 2000] was identified in the knowledge elicitation stage. A 'nature' shape componentisation scheme of 2D section (one view) slices into basic components and for the classification of shapes six component types, identified from knowledge elicitation have been used for componentization. This process was done manually.

Mileman did not focus on arbitrary 3D shapes which were included in his thesis as **future work.** Mileman discussed that similarity measurement techniques for all bar elements are similar to other bars. Yet it is clearly seen that there are differences from one bar to another bars by looking at the locations, roles and types of the bar. The contribution in the similarity part was very strongly achieved to tackle these types of problems by creating an individual formula (consider aspect ratio for each type of component) for components to be compared against each other.

7.3 Further work

Although the contributions to knowledge of this research are significant, another real contribution of this work is the future work that it motivates. The main goal of this is to further advance the efficiency of shape retrieval using a number of research directions:.

1- More investigation on **taper components**, which represent a challenge to automating complex 3D shapes, could help to add to the current library new shapes. For this research only two types of taper have been considered type (e) and type (f) (see on Fig 4.5 shows the different Types of Taper). The reason was the complexity with other types of taper and this part has been explained and suggested in chapter four and illustrated by examples. Although these do not occur frequently in casting shapes, it would be beneficial to investigate the further inclusion of such components.

2- **Orientation of casting designs**

The second expert used in the evaluation argued that orientation advice is important but until now we try to avoid delving deep into the issue of rotating the moulds to achieve better casting design. This issue can be investigated deeper in further research.

3- **Integration with physical modelling systems** is another challenge. There are many physical modelling systems used in shape design such as the CAD application, which is excellent for designing multi-dimensional shapes.

However, their inherent fault lies in not being able to give valuable re-design advice. A design may be sound, but a cheaper way to make a similar or identical design may have been designed several years ago. If ShapeCBR could be integrated with such physical modelling systems such as CAD, the ShapeCBR System would not only give the advice, but valuable design analysis could also take place. This would be a valuable aid for the shape designing industry.

4- **Finally,** if ShapeCBR were integrated with network support, it would be possible for multiple users collectively to interact with the application at the same time, using separate interaction elements, and thereby promoting group work. This could include access to casting design advice case base coming from different foundries and experts that could decide to pool their knowledge.

7.4 Conclusions

There are several complex processes that relate to human case-based reasoning in casting design. For years casting designers have looked for simple and intuitive mechanisms to evaluate, assess and re-use designs.

The ShapeCBR system was designed and built to assist this research at all research stages, to test the research ideas that have been thought out and planned to achieve the final goal of retrieving similar cases or a high percentage of closest cases from the case-based knowledge, by giving a target one which gain a successful result.

The ShapeCBR system could be used together with standard CAD system features to automate shape componentization problems, particularly in industrial areas. For example, assuming it is one of the CAD packages: the ShapeCBR system allows the foundry engineer to quickly bridge the gap between the designs and development. Optimisation or better efficiency during the developed cycle leads to substantial time and cost savings.

However the ShapeCBR system has been of significant assistance in this research and succeeded in providing an evaluation for the research problems such as the shape designing process, the decomposition process, the classification process, the shape matching process and finally the evaluation system and associated tests have been created to test the efficacy and performance of ShapeCBR system. The feedback from the results was extremely successful.

All these processes have been presented within the various chapters in detail to meet the main objectives and their sub objectives. The outcomes have been evaluated for improvement against human experts and the results and shown that the experts past and current had the same knowledge in the same times was useful to exchange the knowledge between them have been presented in chapter evaluation.

Finally the conclusion on the achievements prove that it is possible for a shape similarity problem to manage to retrieve useful casting information efficiently and automatically from a "similar" existing three-dimensional casting design to a given target shape.

References

Anandan S. and Summers D. J.:2006 "Similarity Metric Applied to Graph Bas ed Design Model Authoring". Computer-Aide Design & Applications, Vol. 3, Nos. 1-4, 2006, pp 2 7-306.

[**AutoCAD**]. Autodesk Inc. USA.

Aziz M.:2004 Slemani Metal Casting Design Expert from Kurdistan Government Region Iraq.

Aziz M.:2003 Slemani Metal Casting Design Expert from Kurdistan Government Region Iraq.

Biederman I., "Recognition-by Components: A Theory of Human Image Understanding", "Psychological Review", vol. 94(2), pp.115-147, 1987.

Bacik, R. "Structure of Graph Homomorphisms" Ph. D. Thesis, Simon Fraser University, 1997.

Binford T. O., "Visual Perception by Computer", "IEEE conference on Systems and Control", December 1971.

Blust, G. and Mehta, D. P. 1993. Corner stitching for L-shaped tiles. In Proceedings of the Third Great Lakes Symposium on VLSI (Kalamazoo, MI, March 5–6), 67–68.

Borgefors G., di Baja G. S., and Svensson S., "Decomposition Digital 3D Shape Using a Multiresolution Structure", in Bertrand G., Couprie L., and Perroton L., eds., "DGCI'99, LNCS 1568", Springer Verlag, pp. 19-30.

Burbeck C. A. and Pizer S. M., 1995 " object representation by cores".Vision Resarch, 35:13: 1917-1930 and 1995.

Bunke, H. and Messmer, B. T. (1994) "similarity measure for structured representations", in (eds) Burkhard, H. D. and Lenz, M., Topics in Case-Bsaed Resoning. Bunke H.:1997" On a relation between graphs edit distance and maximum common subgraph". Pattern Recognition Letters, 18(8):689–694, Aug 1997.

Bunke, H. and Guenter, S,: *Validation indices for graph clustering*, in J.-M. Jolion, W. Kropatsch, M. Vento (eds.): Proc. 3rd IAPR- TC15 Workshop on Graph-based Representations in Pattern Recognition, 2001, 229 – 238.

Bunke, H. and Shearer, K.: *A graph distance metric based on the maximal common subgraph*, Pattern Recognition Letters, Vol 19, Nos 3 - 4, 1998, 255 – 259.

Campbell, J. 1991b"Castings", Buttterworth-Heinemann.

Cai, Y. and Wong, D. F. 1993. On minimizing the number of L-shaped channels in 93 building-block layout. IEEE Trans. Comput. Aided Des. 12, 6, 757–769.

Carbonell and Mitchell (Eds) Morgan Kaufmann, Boston(1995) Software environment to support early phases in building design. Approach, vol. II.

Cleary et al.: 2003"A new way of Modelling of high Pressure die casting" Third International Conference on CFD in the process Industries CSIRO Melbourne Australia 10-12 Des. 2003.

Corbett, C.F. December, 1989 Solstar "Right first time...with solstar", Foundry Practice, Number 218.

Coulon C.H.:1995 `Automatic indexing, retrieval and reuse of topologies in architectural layouts'. The Global Design Studio-Proceedings of the 6th International Conference on Computer-Aided Architectural Design Futures, 577-586, Singapore, (1995).

Dai, W. M., Asano, T., and Kuh, E. 1985. Routing region definition and ordering scheme forbuilding-block layout. IEEE Trans. Comput. Aided Des. 4, 189–197.

Debenham (1998:1) "states that a unified KE methodology treats data". Chapter 3: Review of literature.

Dreyfus, H. (1993). *What computers still can't do: A critique of artificial reason.* Cambridge, MA: MIT Press.

Eroh2 Linda1 and Schultz2 Michelle1: 1998 "Matching Graphs" Journal of Graph Theory, 1998 - doi.wiley.com, Received December 11, 1996; accepted March 13, 1998.

Eshera M. A. and Fu K.-S.: 1968" An image understanding system using attributed symbolic representation and inexact graph matching". IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(5):604–618, 1986.

Fernández, L. M. and Valiente, G. "A Graph Distance Metric Combining Maximum Common Subgraph and Minimum Common Supergraph," *Pattern Recognition Letters,* vol. 22, nos. 6-7, pp. 753-758, 2001.

Fischer G. (1994) Domain-Oriented Design Environments. In Automated Software Engineering, Kluwer Academic Publishers, Boston, MA, pp. 177-203.

Flemming, U. and Woodbury, R. (1995) Software environment to support early phases in building design: Overview. Journal of Similarity metrics used in the CBRSP Engineering **4**:1:147-152.

Foggia P., Sansone C., Tortorella F., and Vento M.: 1999" Definition and validation of a distance measure between structural primitives". Pattern Analysis and Applications, 2(3):215–227, 1999.

Gagvani N. and Silver D., "Shape-based Volumetric Collision Detection", "IEEE Volume Visualization and Graphics Symposium", 2000.

Gao, S. and J. Shah, J" Automatic recognition of interacting machining features based on minimal condition subgraph", Computer Aided Des 30 (1998)

Gebhardt, F. 1997 "Survey on structure-based case retrieval", The Knowledge Engineering Review, Vol.12: 1, pp: 41-58.

Hennessy, D; Hinkle, D. October (1992). Applying case-based reasoning to autoclave loading. IEEE Expert 7(5), pp. 21-26. October 1992. Keane-88.

Hinrichs, T. R. (1992). *Problem Solving in Open Worlds: A Case Study in Design.* Hillsdale, NJ.: Lawrence Erlbaum Associates, Pubs.

Hinrichs, T.R. and Kolodner, J.L. (1991) The Roles of Adaptation in Case-Based Design. In *Proceedings of the 1991 DARPA Workshop on Case-Based Reasoning,* pages 121-132, Washington, D.C., May 1991.

Hoffman D. and Richards W. , "Parts of recognition," Cognition, vol. 18, pp. 65 96, 1984.

Holvio 1993 Creative computer aided architectural design. An internal approach to the design process".

Huet, B., and E. R. Hancock, "*Inexact graph retrieval*," In Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries, pp. 40-44. 1999. Colorado. USA.

Jolly, M. 1996 "Overview and appraisal of numerical analysis software for the simulation of casting processes", BICTA bulletin 3, 1996

Kiviniemi, A. & Penttilä, H.:1995" The Role of CAD In Creative Architectural Sketching". Rakennus-CAD. Helsinki,Rakennustietosäätiö148 s.

Knight, B; Cowell, D.; Preddy, K. 1995 "An object-oriented support tool for the design of casting procedures", In Engineering Applications of Artificial Intelligence, Volume.8, Number 5, pp: 561-567.

Kolodner J. (1993) Case-Based Reasoning. Morgan Kaufmann, San Mateo.
21. Liebich T. (2000) XML schema language binding of EXPRESS for ifcXML, MSG-01-001 (Rev 3), International Alliance for Interoperability.

Kolodner, J., ed., Proceed- ings of the DARPA Case-Based Reasoning Workshop, 425434. San Mateo, CA: Morgan Kaufmann. Veloso, M., and Carbonell, J. 1994.
Lingas, A., Pinter, R., Rivest, R., and Shamir, A. 1982. Minimum edge length partitioning of rectilinear polygons. In Proceedings of the 20th Allerton Conference on Commun. Control Comput., 53–63.

Lipson, H. and Shpitalni, M., 1996, "Decomposition of a 2D polygon into a minimal set of disjoint primitives", *CSG96 Conference on Set-Theoretic Solid Modeling*, Winchester, UK, April 1996, pp. 65-82.

Maher, M. L., Balachandran, M. B. and Zhang, D. M. (1995) Case-Based Reasoning in Design. Lawrence Erlbaum Assoc., NJ. Publishers, Mahwah, New Jersey, 1995

Maher, M.L. (1997) SAM: A multimedia case library of structural designs. In Y-T Liu, J-Y Tsou, and J-H Hou (eds) CAADRIA '97, pp 5-14.

Mangan A. and Whitaker R., "Partitioning 3d surface meshes using watershed segmentation," IEEE Trans. on Visualization and Computer Graphics, vol. 5, no. 4, pp. 308–321, 1999.

Mario, A. Lopez and Mehta, D.:1994 "Efficient Decomposition of Polygons into L-Shapes with Applications to VLSI Layouts". Volume 1, Issue 3 (July 1996) Pages: 371 - 395 Year of Publication: 1996 .ISSN:1084-4309.

Marir, F; Watson, I. 1994 "Case-based reasoning: a categorized bibliography", The Knowledge Engineering Review, Vol.9: 4, pp: 355-381, 1994. See p. 374.

Medland AJ, Mullineux G. (1988). CAD: A Course Book.

Mehta, D. P., Shanbhag, A., AND SHERWANI, N. A. 1995. A new approach for floorplanning in MBC designs. Tech. Rep. 95-02, Univ. of Tennessee Space Institute.

Messmer, BT, Bunke, H., 1999. A decision tree approach to graph and subgraph isomorphism detection. Pattern Recogn. 32, 1979-1998.

Mileman T., Petridis M., Knight B 2000 (An Investigation of the Feasibility of a CBR Approach to the Design of Sand Castings in the Foundry Industry).

Mitchell T.M. (1997) Machine Learning. McGraw-Hill.

Moore C.J. (1993) Complementary Innovative Computer Systems for Bridge Design. EG-SEA-AI Workshop: Application of Artificial Intelligence. In Structural Engineering, Lausanne, Switzerland.

Naaranoja, Marja and Polytechnic, Vaasa1997" Information Technology Strategies and Adaptation of Knowledge a Conceptual Analysis in the Construction Industry".

Nahar, S. and Sahni, S. 1988. A fast algorithm for polygon decomposition. IEEE Trans. Comput. Aided Des. 7 (April), 478–483.

Nonaka I., and Konno, N.:1998" The Concept of Building a Foundation for Knowledge Creation". *Educational Technology & Society 5 (3) 2002 ISSN 1436-4522.*

NovaCast: Sillen, R. December, 1991 "Using artificial intelligence in the foundry", Modern Casting,

Oxman, R. (1994). "Remembrance of things past: design precedents in libraries." Automation Based Creative Design, A. Tzonis and I. White, eds., Elsevier Science, Amsterdam, 55-68.

Oxman R and Ann, H.:(1991) "A Case with a View Towards an Integration of Visual and Case-Based Reasoning in Design"

Pegler :1993" Case-based reasoning in the melting pot" International Journal of Applied Expert Systems, volume 1(2), 1993.

Pentland A., "Part segmentation for object recognition," Neural Computation, vol. 1, pp. 82–91, 1981.

Price, C.J; Peglar, I.S.; Ratcliffe, M.B; McManus, A. 1997 "From troubleshooting to process design: closing the manufacturing loop", Lecture Notes in Artificial Intelligence. 1266, Case-based reasoning research and development. Springer-Verlag. 1997.

Rivard H., Fenves S.J. (2000) A Representation for Conceptual Design of Buildings. Journal of Computing in Civil Engineering, vol. 14, no. 3, pp. 151-159.

Ravi, B. January, 1999"Computer-Aided Casting Design - Past, Present and Future", Special Issue of Indian Foundry Journal.

Rom H. and . Medioni G, "Part decomposition and description of 3d shapes," in Proc. Int'l Conf. Pattern Recognition, 1994, pp. 629–632.

Schenker, A., Last, M., Bunke, H., Kandel, A.: 2003 "Comparison of Distance Measures for Graph-based Clustering of Documents". In: Hancock, E. and Vento, M. (eds.): Lecture Notes in Computer Science, Vol. 2726 (2003) 202–213.

Schank R.C. (1982) Dynamic Memory: A Theory of Learning in Computers and People, Cambridge University Press, Cambridge.

Schank R.C. (1999) Dynamic Memory Revisited. Cambridge University Press, Cambridge.

Schulz S.:1999 "CBR-Works - A State-of-the-Art Shell for Case-Based Application Building". Proceedings of the 7 th German Workshop on Case-Based Reasoning, GWCBR'99, Wrzburg, Germany, March 3-5, 1999.

(SEED) Griffith ED, Hicks DK and McGraw KD: 1999 "Towards Model Based Design A Case Study": The Modular Design System - Eight International Conference on Durability of Building Journal of Architectural Engineering, Vol. 1 No. 4, ASCE, pp. 147-152.

Shanbhag , A. G., Danda S. R., and Sherwani, N. 1994. Floor planning for mixed macro block and standard cell designs. In Proceedings of the Fourth Great Lakes Symposium on VLSI (Notre Dame, IN, March 4–5), 26–29.

Shokoufandeh A. and Dickinson S.: 2001" A Unified Framework for Indexing and Matching Hierarchical Shape Structures". In *Proceedings, 4th International Workshop on Visual Form*, Capri, Italy, May 28–30 2001.

Shokoufandeh A. and Dickinson S.: 1999 "Applications of Bipar-tite Matching to Problems in Object Recognition". In Pro-ceedings, ICCV Workshop on Graph Algorithms and Com-puter Vision, September 1999.

Shokoufandeh A, Dickinson SJ, Siddiqi K, Zucker SW: 1999 "Hierarchical decomposition "Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference, Vol. 2 (1999).

Simmons M. and Sequin H. C.: 1998"2D Shape Decomposition And The Automatic Generation Of Hierarchical Representations".

Sowa, J. F. (1984) Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, Mass.

Sundar, H., Silver, N. D., Gagvani and Diskinson. "Selection based shape matching and retrieval". In Shape Modelling and Applications Conference SMI 2003, Seoul, Korea, May 2003.

Tan T. S., Chong K. F., and Low K. L., "Computing Bounding Volume Hierarchy using Simplified Models", "Proceedings of ACM Symposium on Interactive 3D Graphics",pp. 63-69, 1999.

Tammer, E.C; Steinhofel, K; Schoner, S; Matuschek, D. 1995 "Anwendung des Konzepts der strutrellen Ahnlichkeit zum Fallvergleich mitterls Term- und Graph-representationen", Fabel report, 38, GMD, Sankt Augustin, 1995.

Taylor,N. R., N. Medvidovic, K.M. Anderson, E.J. Whitehead Jr., J.E. Robbins, K.A. Nies, P. Oreizy, and D.L. Dubrow, 1996 "A Component-Based Architectural Style for GUI Software," *IEEE Trans. Software Eng.,* vol. 22, no. 6, June 1996.

Wallis, W.D., Shoubridge P., Kraetz, M., Ray, D.: 2001" Graph Distances using Graph Union". *Pattern Recognition Letters* 22 (2001) 701–704.

Waltz, D. L., (1988) Is indexing used for retrieval? Proc. DARPA Case-Based Reasoning Workshop, Clear Water Beach, FL. 41-44.

Wang Y., Ishii N.:1997 "Method of Similarity Metrics for Structured".

Wang, T. and Wong , D. F. 1990. An optimal algorithm for floorplan area optimization. In Proceedings of the 27th Design Automation Conference, 180–186.

Watson, I. 1994 " Cased-based reasoning : a categorised bibliography.

Wlodawer R., 1967, Directional Solidification of Steel Castings, Pergamon.

Wilson and Hancock: 1996 "A Bayesian Compatibility Model for Graph Matching". Pattern Recognition Letters, vol. 17, pp. 263-276, 1996.

Wu K. and Levine M. D., "3d part segmentation using simulated electrical charge distributions," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, pp. 1223–1235, 1997.

Wu, S. and Sahni, S. 1991. Fast algorithms to partition simple rectilinear polygons. Int. J. Computer. Aided VLSI Des. 3, 241–270.

Wu, S. and Sahni, S 1990. Covering rectilinear polygons by rectangles. IEEE Trans. Computer. Aided Des. 9 (April), 377–388.

Yeap, K. and Sarrafzadeh, M. 1993. Floorplanning by graph dualization: 2-Concave rectilinear modules. SIAM J. Comput. 22 (June), 500–526. Decompositions of Polygons into L-Shapes • 395Representations".

Xuetao Li, Tong Wing Woon, Tiow Seng Tan and Zhiyong Huang,2001" Decomposing Polygon Meshes for Interactive Applications".

Online:

(Online): http://people.hofstra.edu/geotrans/eng/ch2en/meth2en/ch2m1en.html

(Online): http://www.cfd-online.com/Resources

(Online): **Robert** Osada, Thomas Funkhouser, Bernard Chazelle, David Dobkin , Princeton University.

(Online): http://www.diaonisticstrategies.com/papers/similarity.htm

(Online): http://Alloy characteristics and Casting Design.htm

(Online): http://Steel Founders Society Of America (2006)
The Casting Design Handbook, 1962, American Society for Metals.

Software:

CRUSADER software, SCRATA, Sheffield.

SOLSTAR, FOSECO Ltd, Tamworth

FEEDERCALC, FOSECO Ltd, Tamworth

Appendix A

Published Paper I

S Saaed, M Petridis, , B Knight and T J Mileman 'Automating Case Creation and Selection for Case-Base of Rotationally Symmetric Shapes for the Design of Metal Castings'. 7th UKCBR Workshop-10th December 2002-Peterhouse, Cambridge.

Abstract

In this paper, we discuss the problem of automating the process of creation and selection of cases to populate a CBR system for retrieval of rotationally symmetric shapes to assist with the design of metal castings. The special feature of this system is that similarity is derived primarily from graph matching algorithms. The particular problem of such a system is that it does not operate on search indices that may be derived from single cases and then used for visualisation and principal component analyses. Rather, the system is built on a similarity metric defined directly over pairs of cases. An overview of previous research in this area is presented. This demonstrated the feasibility of a CBR approach to the design of metal castings. The architecture of the *ShapeCBR* system is presented. Performance measures for the CBR system are given, and the results of trials of the system are presented. This paper describes further research into the use of the traditional componentisation as used in method engineering to provide a shape representation suitable for efficient retrieval of design knowledge. Finally, this paper presents current work aiming mainly at enhancing the efficiency and accuracy of the similarity metrics used in the *ShapeCBR* system.

Keywords

Case-Based Reasoning, Spatial reasoning, Shape recognition, Casting design, Knowledge Based Systems, 3-D Shapes, Casting, Foundry.

1 Introduction

This paper reports on current research at the University of Greenwich that aims at automating the process of retrieving and reusing design knowledge involved in the design of sand castings. In the heart of this problem lies the problem of retrieval of rotationally symmetric shapes from a case base designed as a design assistant to be used in the metal casting industry. In a previous paper [7] a support tool was described, based upon a traditional componentisation of 3D shapes, with components of known cooling modulus. An initial evaluation of the tool through its application to a family of rotationally symmetrical casting shapes has shown the feasibility of a CBR approach to assist the design of sand castings [25],[26].
Out of a range of solids processing methods for the mass production of components; for example, casting, forging and machining, casting is the generally the cheapest. However, the problem with casting is one of quality, which depends on the existence of casting design knowledge. The advantages of a CBR system, capable of containing detailed information on the design process for products, devolve from its ability to realise casting know-how as a valuable asset. The knowledge of how to cast a product soundly within tight cost constraints is the result of a huge investment on the part of industries, universities and government over many years. Although the value of design knowledge is widely recognised throughout the industry, the management of design knowledge is often ad hoc in some respects. Design histories are often lost, or banished to paper files that are difficult to search. Also, design engineers retire [23], or move away leaving inadequate design records.

There are many problems faced by a casting design engineer, centering on the physical freezing processes. Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness [1]. Other practical problems arise during pattern making and subsequent machining of the cast part. Many software tools have been developed to assist the designer. Jolly [2] found in his survey that the foundry industry is looking for software that can not only predict problems that occur during metal solidification (such as shrinkage porosity) but also, having predicted these problems, to propose intelligent solutions to problems found. Current commercial casting software can be classified into two broad areas: intelligent knowledge-based systems (IKBS)[22],[24], and numerical simulations based on physical process models[3-5].

IKBS systems attempt to support an earlier stage in the design process. Numerous software tools such as those discussed in [7] have clearly demonstrated the usefulness of knowledge-based and other advanced heuristic-based programs for designing castings. Some of the commercial software packages available can calculate the position of feeders (NOVACAST [8]) and also analyse geometric properties to give suggestions to improve the design further (AutoCast [9]).

Although many prototype tools have demonstrated the efficacy of CBR in the domain of engineering and design [10-15], there is a scarcity of research for its use in the foundry industry. CBR can play an important role in intelligent casting software. One commercial CBR system [16] called Wayland, is used for the setting of parameters in pressure die-casting. This research has demonstrated that CBR has an exciting future in casting software.

The main problem for a CBR system is how to retrieve cases, where the retrieval must be based on shape. Although there are other possible search indices, for example the type of casting alloy, weight and general description of part (wheel, sea-gland, valve, engine bearing cap, etc.), these descriptions are too general for accurate retrieval. General classifications of shape components have been proposed; for example, Biederman's *geons* [17]. However, during this research, it became apparent during knowledge elicitation that a decomposition of shapes specific to the casting industry already existed in practice, [7, 18]. The research described here uses a graphical representation of shapes based on this decomposition as a foundation for shape retrieval. This paper is mainly concerned with the process of automating the process of decomposing the geometry of a real 3D geometrical casting shape to the graphical representation needed to allow for the efficient retrieval of similar shapes and thus reuse relevant casting design knowledge.

In section 2 of this paper, the graphical representation and the similarity measures used for retrieval are explained. Section 3 discusses the automatic process of encoding real 3D casting shape into the case base to allow efficient retrieval of similar shapes and reuse of casting knowledge. Section 4 gives an evaluation of the current system based on experimental results from a trial domain of rotationally symmetric objects.

2 Graphical representation and similarity metrics

In [7] a decomposition of a shape into a set of joined components was described. The decomposition is a natural one, used over many years by casting design engineers. It is based on a set of component types of significance in casting design. There are 8 main component types including Bar, L, T, X, Taper, Flange, Bespoke-Taper, and Bespoke-T. The componentisation process distinguishes two sets of component type: those that define the structure (L's, T's and X's) and those that join the first set together (bars and tapers). Using this classification, we may abstract a graphical representation of the structure of any shape S where the nodes are elements of either set, and the arcs represent interfaces between components.

As an illustration, consider the rotationally symmetric shape shown in cross section in figure 1. A graph representation of this figure is given in Figure 2.

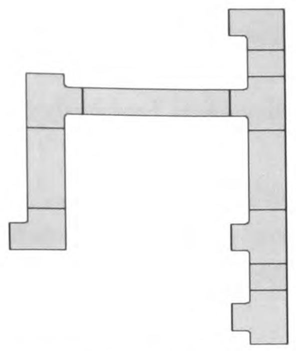


Figure 1. A casting, made of 3 component types (Bar, L, T)

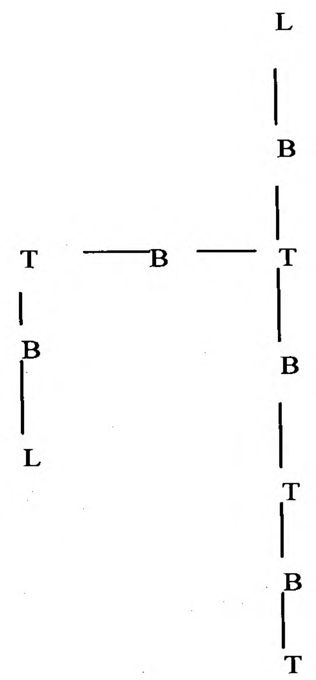


Figure 2. Representation of figure 1 as a graph

Retrieval of shapes for casting design is an example of structure based case retrieval, as defined by Gebhardt [15]. For these systems, attributes representing complex structures are difficult to define, and similarity must be derived from structure directly. For the sub-class of graphical structures, Gebhart reviews several retrieval systems. These include clique detection as in the Fabel component Topo [19], largest common subgraph [20] and hamming distance [21].
In the research described here, we have used similarity measures based on features extracted from the structural graphs. Perfect similarity between shapes S1 and S2 is obtained when they have identical structural graphs. However for graphs that do not match completely, there are a number of features that can be extracted and compared. Each feature gives rise to a different similarity measure, representing a different case retrieval.
Correspondingly, there are a number of different problems associated with casting a shape, each connected with a different structural feature. Porosity tends to depend on specific local features, whereas machining problems tend to depend on global structure. The approach of this research has been to construct a retrieval tool to investigate the efficacy of the various metrics with respect to different casting design problems. The tool employs a generalised similarity measure $\sigma(S1,S2)$ between shapes S1 and S2, representing a weighted sum of the similarity measures based on different features extracted from the graphs of S1 and S2:

$$\sigma\,(S1,S2) = w_{comp}\sigma_{comp} + w_{mcs}\sigma_{mcs} + w_{cycle}\sigma_{cycle} + w_{leaf}\sigma_{leaf} \qquad (1)$$

Variation of the weights in this formula allows a general test of retrieval against any given casting problem.

The individual similarity metrics in (1) are defined as follows:

- $\sigma_{comp}(S1,S2)$ is a measure based on the number of component types that are common to the two graphs. If two graphs are nearly identical, σ_{comp} will be close to 1. The length equation is defined as length(S) = number of components in S, and the value of this metric is given by:

$$\sigma_{comp}(S1,S2) = \sum_{comp} \frac{\mathrm{length}(\mathrm{S}'_{\mathrm{comp}})^2}{length(S1)length(S2)} \qquad (2)$$

where S'_{comp} is the maximal number of common components of a particular type to graphs S1 and S2. Nevertheless, this metric does not take into account of the graph's topology. This is taken care of by the following metric:

- σ_{mcs} (S1,S2) is a measure based on the length of the maximum matching subgraph. If two graphs are nearly identical, σ_{mcs} will also be close to 1. This similarity metric is given by:

$$\sigma_{mcs}(S1,S2) = \sum_{comp} \frac{\mathrm{length}(\mathrm{S}')^2}{length(S1)length(S2)} \qquad (3)$$

where S' is the maximal common subgraph of S1 and S2, i.e. the largest graph which is a subgraph of both S1 and S2. The problem of finding S' is related to that of the well-known graph isomorphism problem. For small graphs of up to 10 arcs, a search based on direct comparison of all subgraphs of S1 with those of S2 is possible. For larger graphs a strategy based on a preliminary comparison of node types and degree can help to reduce the search time.

- σ_{ncycle} (S1,S2) is based on a count of elementary graph cycles:

$$\sigma_{cycles}(S1,S2) = 1 - \frac{|\mathrm{ncycles(S1) - ncycles(S2)}|}{\mathrm{max(ncycles(S1), ncycles(S2))}} \qquad (2)$$

- σ_{nleaf} is based on a count of leaf nodes, and gives the number of branches to a tree:

$$\sigma_{leaves}(S1,S2) = 1 - \frac{|\mathrm{nleaves(S1) - nleaves(S2)}|}{\mathrm{max(nleaves(S1), nleaves(S2))}} \qquad (3)$$

3 Automating the decomposition process

The *ShapeCBR* software system has been developed at Greenwich to automate the process of matching a given target shape to a case in the Case Base. The case base is populated with cases containing information relevant to real metal casting experience. The information contained in each case relates to both a geometrical description of real shape and domain specific information about the way that the shape was actually cast. Additionally, some cases may contain general expert advice relevant to casting the shape in a textual form. The system allows the user to retrieve a shape from the case base to match a target case, according to a match on the four contributing features as described in the previous section. The user to attach varying importance to each of the similarity measures can apply weighing factors.

Figure 3 shows an example exercise of matching a target case to a retrieved case from the case base. Advice on positions of feeders and chills is annotated on the picture of the retrieved case.

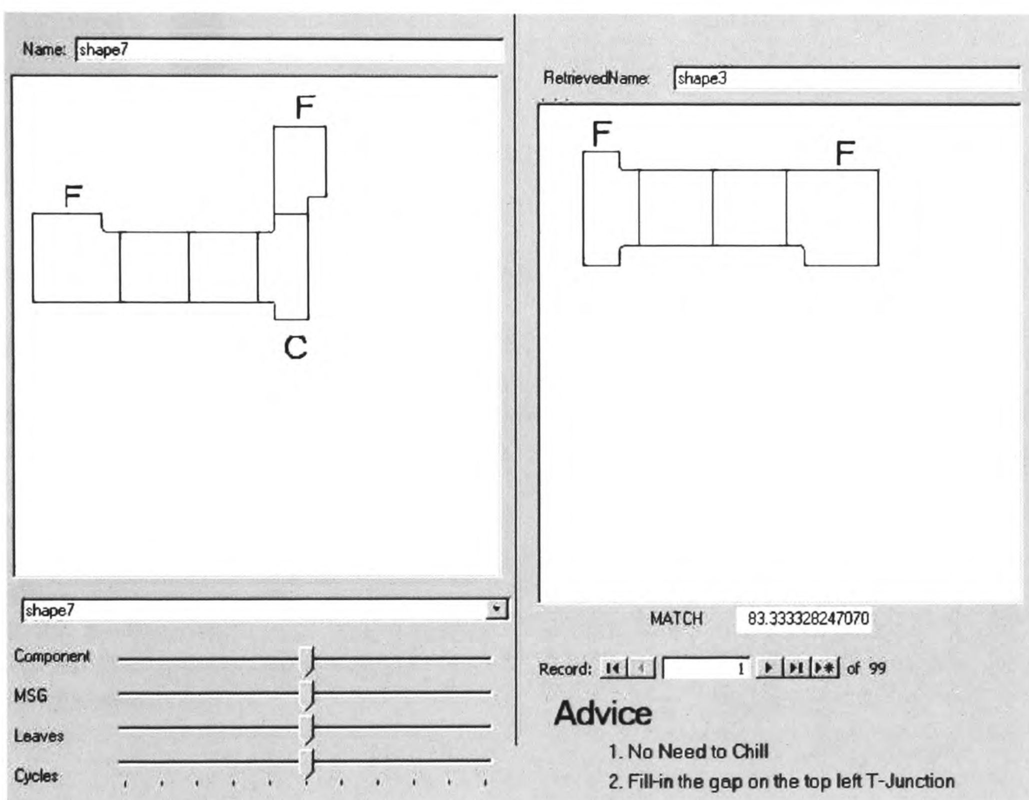


Figure 3. Matching a case to a target in the *ShapeCBR* system

Early feedback from the use of the system has been promising[25]. However, the first generation of the *ShapeCBR* system as described by Mileman [26] relied on the manual decomposition of shapes to the generic components identified. This was an inefficient and cumbersome process that could hamper the practical use of a commercial system. Additionally, only the type and not the actual geometrical dimensions of a component were stored. This prevented us from increasing the sensitivity of the similarity criteria to take into account a similarity measure between components of the same type. For example, it makes sense that the aspect ratio of a Bar component would affect its similarity to another Bar component for purposes of casting. The positioning of feeders and chills can be affected, so that the knowledge associated with a shape may be contingent not only on the types, but on actual geometrical features of the constituent components.

Typically, casting shapes are stored as files produced by CAD packages such as AutoCAD[6]. These files contain all geometrical information and most CAD packages provide facilities for providing 2D sections through the 3D shape. The case base in the first system contained only one 2D section through each shape. However, in many cases two or more substantially dissimilar 2D sections could provide a more accurate description of a 3D shape. These would need to be taken into account for a more efficient retrieval of 3D shapes. The selection of dissimilar 2D sections can be achieved with the use of a similarity threshold to define substantially dissimilar sections.

In this case, the overall measure of similarity between two 3D shapes S1 and S2 needs to be redefined as:

$$\sigma_{3D}\,(S1,S2) = \Sigma\ w_i\sigma(S1_{2Dsection\ n}, S2_{2Dsection\ m}) \qquad (4)$$

where each 2D section (n,m) of each 3D session is used only one so that the above measure is maximised.

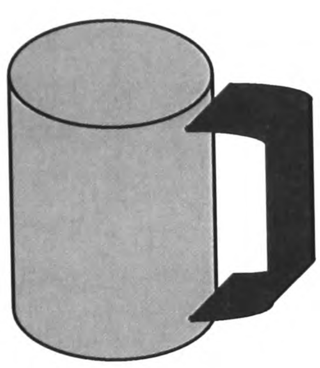


Fig. 4. Arbitrary 3-D Shape (mug)

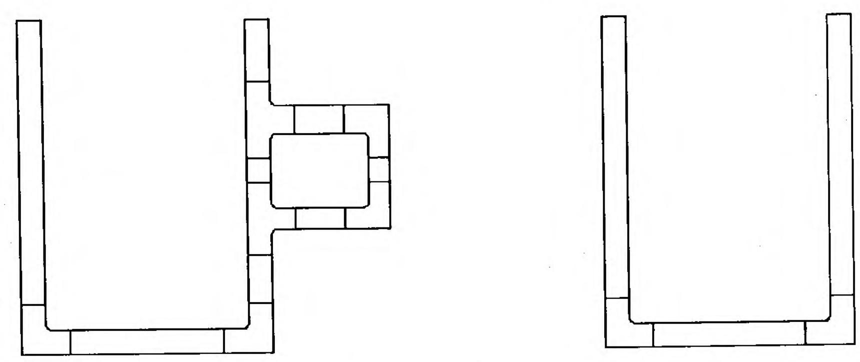


Fig. 5. View 1 of Mug (of Figure 4.10) made of Bars, L-junctions and T-Junctions.

An algorithm was devised to provide for the automatic decomposition of shapes into the generic components used in this research. This algorithm starts by projecting each vertex to any sides that are directly opposite it. This provides a decomposition of the area of the shape into a set of rectangles and triangles (fig. 6 second shape). These are then reconciled and merged if sides defined by internal points only connect them. For example two consecutive Bar components can be merged into one longer one. A set of rules then identifies each element as one of the generic components needed for the componentisation of the shape. For example, a rectangular component that has two opposite internal sides is resolved to be a Bar. Finally, the components are created by adding "stems" where appropriate (typically to joins, such as L, T and X). Figure 6 shows an example of such a decomposition. Notice the top left L component. In the middle figure, the algorithm has identified there a rectangle. The rule that identifies this as an L component fires on the fact that this rectangle has two adjacent sides (right and bottom) that are internal lines. This identifies the component as an L.

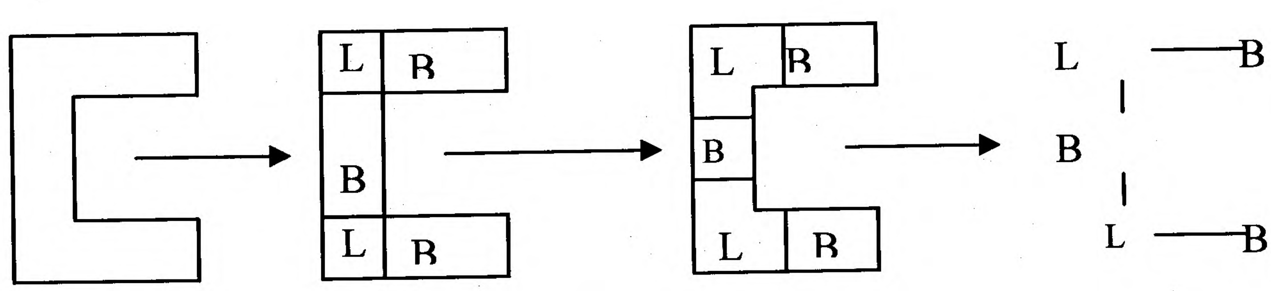


Fig. 6. The decomposition process

An additional advantage of automating the decomposition into components is that the output of this process is not only the graph of connected components representing the structure of the shape. Each component is now associated with geometrical information describing the exact dimensions of the component. This allows us to extend the definition of similarity between shapes taking into consideration the actual geometry in addition to just the layout of the components in each shape.
Figure 7 shows the architecture of the system and an overview of the process of importing cases into the case base.

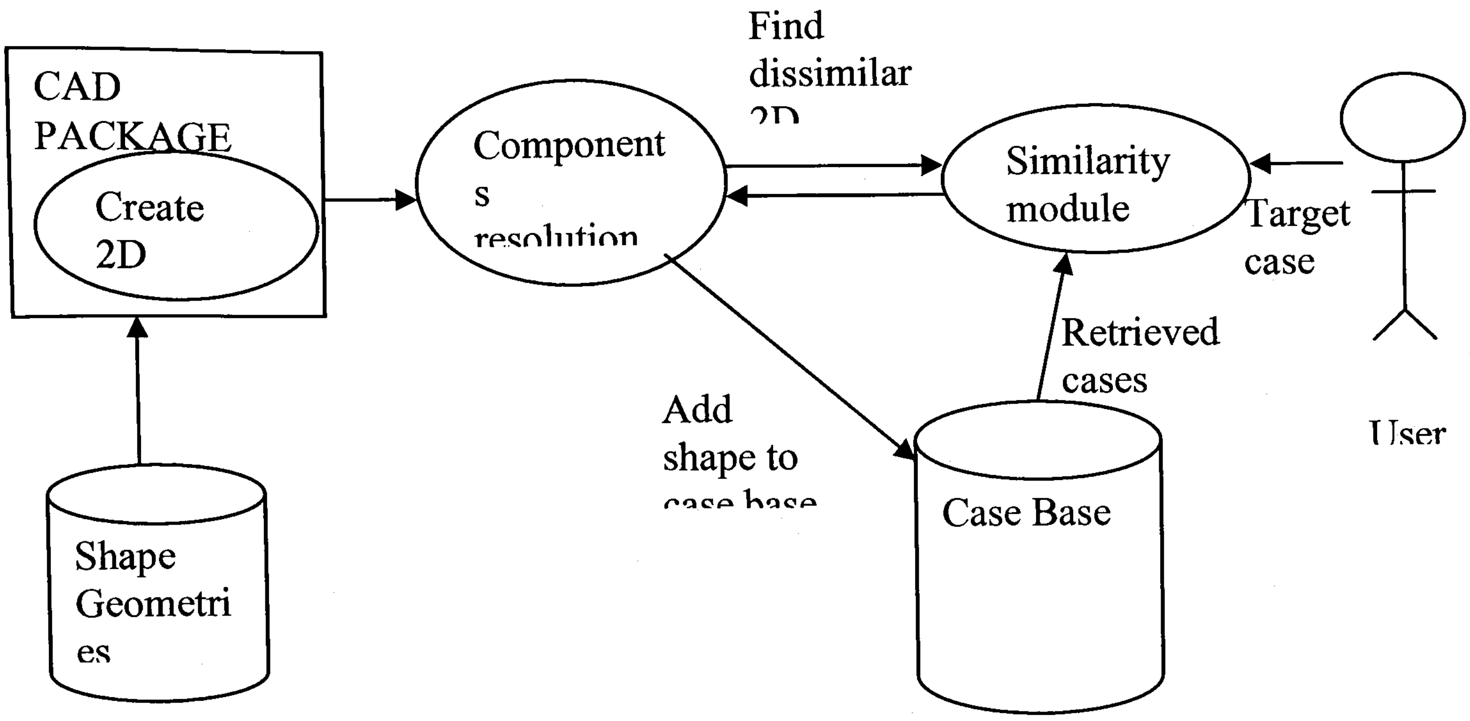


Fig. 7. Overview of the *ShapeCBR* system and process

4 Evaluation of the system

The integrated system with the new algorithm that automates the componentisation of the shapes into cases in the case base was tested against the 100 manually generated cases used to evaluate the previous system. At the current state, the algorithm does not support extraction of taper components, so shapes including such components were excluded from the test. It was then found that the componentisation produced by the system was identical to the one produced manually.

The automation process has thus not affected the overall performance of the system. The performance measures for the whole system were based on the trial case base consisting of 100 cases and tested against 20 new problems. For this trial, we took the domain of shapes with rotational symmetry: wheels, armatures, cylinders, etc. This domain is coherent from a practical point of view, so that we can attempt to cover it with a limited case base. It is however sufficiently varied to encompass a wide range of casting problems.

Performance of the case base was assessed on several different measures. For a given target the retrieved set should provide the solution to (I) correct orientation, (II) the number and positions of feeders (III) the position of possible chills, (IV) the need for chills. (V) special problems encountered with this shape. For each of these problems, we can score how well the retrieved case presents the answer. In situations where no obvious visual match may be made with the nearest case, we can widen the search to retrieve more cases, and leave the user to select the one with the best visual match. In such a mode of operation, the user is allowed to browse the nearest matches to look for the best advice. A full discussion of this trial can be found in the paper by Knight et al [25] and in the thesis by Mileman [26].

Table 1 summarises the results of this trial using either equal weights for all similarity components or two types of optimisation of the weights (first only involving 20 cases, the second optimisation using all 100 cases tested around the first set of optimised data).

	Equal weights	First optimisation	Final optimisation
Orientation	75.5%	81.5%	75%
Feeders	74.28%	76.8%	84%
chill positions	57.28%	56.27%	79%
chill advice	82.5%	84%	84%

Table 1. Performance for whole case base, as weights are optimised

These performance figures can serve as a benchmark for future maintenance of the case base. We can require that future versions of the case base should perform at least as well on the original 100 cases as the performance figures in Table 1 show.

6 Conclusion

In this paper we have described recent work on a case based system for the design of metal casting procedures. The key problem addressed by the work is the retrieval of rotationally symmetric shapes. The method proposed is based on a shape componentisation, which is particular to the domain of casting problems. The shape componentisation gives rise to a graphical representation of shapes, from which similarity metrics may be abstracted.

We presented an overview of previous research that showed the feasibility of a CBR approach to assist the design of metal castings in the foundry industry. Current research work on automating the process of eliciting the cases from real CAD drawings of 3D shapes has been discussed here. Preliminary tests show that it is possible to automate this process and produce a componentisation similar to the one produced by domain experts. This componentisation can then be used by the CBR system to determine the similarity between shapes and thus retrieve competent solutions for a given target case shape.

Additionally, the inclusion of this missing link into the proposed process allows more information relating to the shape its components to be linked directly to the cases in the case base. In particular, the inclusion of actual geometrical data describing each component in the graph representation of a shape will allow an extension of the similarity metrics that may increase the competence of the system. Additionally, it is now possible to automate the process of eliciting distinct 2D sections through 3D rotationally symmetric shapes. This allows the use of the extended similarity metrics between 3D shapes that draw on the similarity between numbers of 2D sections from each shape. There is currently work under way pursuing these two lines of research.

Finally, it is now possible to link more relevant information to cases in the database. For example documents such as test documents, photographs and blueprints can be associated with a retrieved case, providing richer contextual knowledge and thus improving the usefulness and relevance of the advice given by the system.

Future work is also planned to extend the trials to wider domains, including general 3D systems. Work is also being planned for the integration of the system with physical modelling systems, such as SOLSTAR, to prototype the casting.

References

1. Campbell, J. "Castings", Buttterworth-Heinemann, 1991

2. Jolly, M. "Overview and appraisal of numerical analysis software for the simulation of casting processes", BICTA bulletin 3, 1996

3. Mavis, Diana. Alphacast software Ltd, England

4. Simular. Aluminium Pechiney, Aluval, BP27 38340 Voreppe, France

5. SOLSTAR. Corbett, C.F. "Right first time...with Solstar", Foundry Practice, Number 218, December, 1989

6. AutoCAD. Autodesk Inc. USA.

7. Knight, B; Cowell, D.; Preddy, K. "An object-oriented support tool for the design of casting procedures", In Engineering Applications of Artificial Intelligence, Volume.8, Number 5, pp: 561-567, 1995

8. NovaCast: Sillen, R. "Using artificial intelligence in the foundry", Modern Casting, December, 1991

9. Ravi, B. "Computer-Aided Casting Design - Past, Present and Future", Special Issue of Indian Foundry Journal, January, 1999

10. See p. 374, Marir, F; Watson, I. "Case-based reasoning: a categorized bibliography", The Knowledge Engineering Review, Vol.9: 4, pp: 355-381, 1994

11. Raphael, B; Kumar, B. "Indexing and retrieval of cases in a case-based design system", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10, pp: 47-63, 1996

12. Hennessy, D; Hinkle, D. "Applying case-based reasoning to autoclave loading", IEEE Expert, pp: 21-26, October 1992

13. Kumar H.S; Krishnamoorthy, C.S. "Case-based reasoning in bridge design", pp: 197-205, Advances in Computational Structures Technology, Civil-Comp Press, Edinburgh, 1996

14. Yeh, I. "Case-Based approaches for preliminary design of steel building frames", Microcomputers in Civil Engineering, 12, pp: 327-337, 1997

15. Gebhardt, F. "Survey on structure-based case retrieval", The Knowledge Engineering Review, Vol.12: 1, pp: 41-58, 1997

16. Price, C.J; Peglar, I.S.; Ratcliffe, M.B; McManus, A. "From troubleshooting to process design: closing the manufacturing loop", Lecture Notes in Artificial Intelligence. 1266, Case-based reasoning research and development. Springer-Verlag, 1997

17. Biederman, I.; Hummel, J.E.; Cooper, E.E.; Gerhardstein, P.C. "From image edges to geons to viewpoint invariant object models: A neural net implementation", Applications of Artificial Intelligence X: Machine vision and robotics, 1992

18. Wlodawer, R "Directional solidification of steel castings", Pergamon, Oxford, 1967

19. Coulon, C.H. "General geometric and topological retrieval and adaptation (Topo)". In Borno, K, editor, Modulus for design support, Vol. 35 of Fabel report, pp: 33-445, GMD, Sankt Augustin, 1995

20. Tammer, E.C; Steinhofel, K; Schoner, S; Matuschek, D. "Anwendung des Konzepts der strutrellen Ahnlichkeit zum Fallvergleich mitterls Term- und Graph-representationen", Fabel report, 38, GMD, Sankt Augustin, 1995

21. Bunke, H; Messmer, B.T. "Similarity measures for structured representations". In Burkhard, H.D and Lenz, M, editors, Topics in Case-Based reasoning: First European workshop, EWCBR-93, selected papers; Lecture notes in Artificial intelligence, 837, pp: 106-118, Springer-Verlag, 1994

22. Price, C.J; Peglar, I.S "Deciding parameter values with case-based reasoning", Progress in case-based reasoning (ed. Ian D. Watson), Springer-Verlag, pp: 121-133, 1995

23. See page 123. Price, C.J; Peglar, I.S; Bell, F. Case based reasoning in the melting pot, Int. Journal of Applied Expert Systems, vol.1, no.2, pp: 120-133, 1993

24. Hennessy, D; Hinkle, D. "Applying case-based reasoning to autoclave loading", IEEE Expert, pp: 21- 26, October 1992

25. B Knight, M Petridis (2000) "Maintenance of a case base for the retrieval of rotationally symmetric shapes for the design of metal castings" in LNAI: Lecture Notes in Artificial Intelligence 1898, Advances in Case Based Reasoning , Eds: E Blanzieri and L Portinale pp 418 -430, 2000, Springer

26. Mileman T, "An Investigation of the feasibility of a CBR approach to the design of Sand Castings in the foundry industry", PhD Thesis, University of Greenwich, 2001.

Appendix B

Similarity metrics equations:

1- Cycle Metric Equation

```
Private Equation GetCyclicMetric(ByRef SourceShape As clsShape, ByRef TargetShape
As clsShape) As Double

'calculates cyclic metric

If SourceShape.IsCyclic And TargetShape.IsCyclic Then
   GetCyclicMetric = 1
ElseIf Not SourceShape.IsCyclic And Not TargetShape.IsCyclic Then
   GetCyclicMetric = 1
Else
   GetCyclicMetric = 0
End If

End Equation
```

2- MCS Metric Equation

```
Private Equation GetMCSRMetric(ByRef SourceShape As clsShape, ByRef TargetShape
As clsShape) As Double

'calculates MCSR metric

Dim lngSLComponent As Long
Dim lngSTComponent As Long
Dim lngSXComponent As Long
Dim lngSTpComponent As Long

Dim lngTLComponent As Long
Dim lngTTComponent As Long
Dim lngTXComponent As Long
Dim lngTTpComponent As Long

Dim dblLMetric As Double
Dim dblTMetric As Double
Dim dblXMetric As Double
Dim dblTpMetric As Double
```

```
Dim lngCompenentTypes As Long

With SourceShape
   lngSLComponent = .NoOfLComponents
   lngSTComponent = .NoOfTComponents
   lngSXComponent = .NoOfXComponents
   lngSTpComponent = .NoOfTpComponents
End With

With TargetShape
   lngTLComponent = .NoOfLComponents
   lngTTComponent = .NoOfTComponents
   lngTXComponent = .NoOfXComponents
   lngTTpComponent = .NoOfTpComponents
End With
```

3- Leaves Metric Equation

```
Private Equation GetNoOfLeavesMetric(ByRef SourceShape As clsShape, ByRef
TargetShape As clsShape) As Double
'calculates number of leaves metric

Dim lngSSatelliteBarB As Long
Dim lngSWings As Long
Dim lngSLeaves As Long

Dim lngTSatelliteBarB As Long
Dim lngTWings As Long
Dim lngTLeaves As Long

Dim dblSatelliteBarMetric As Double
Dim dblWingMetric As Double
Dim dblMetric As Double

With SourceShape
   lngSSatelliteBarB = .NoOfSatelliteBarsTypeB
   lngSWings = .NoOfWings
   lngSLeaves = lngSSatelliteBarB + lngSWings
End With

With TargetShape
   lngTSatelliteBarB = .NoOfSatelliteBarsTypeB
   lngTWings = .NoOfWings
   lngTLeaves = lngTSatelliteBarB + lngTLeaves
End With
```

```
'calculate metrics

dblMetric = Abs(lngSLeaves - lngTLeaves) / GetMaxValue(lngSLeaves, lngTLeaves)
GetNoOfLeavesMetric = 1 - dblMetric

End Equation
```

4- Component Type Metric (CTM) Equation

```
Private Equation GetTypeOfComponentsMetric(ByRef SourceShape As clsShape, ByRef
TargetShape As clsShape) As Double

'calculates number of components metric

Dim lngSLComponent As Long
Dim lngSTComponent As Long
Dim lngSXComponent As Long
Dim lngSTpComponent As Long
Dim lngSSatelitteBarTypeA As Long
Dim lngSSatelitteBarTypeB As Long

Dim lngTLComponent As Long
Dim lngTTComponent As Long
Dim lngTXComponent As Long
Dim lngTTpComponent As Long
Dim lngTSatelitteBarTypeA As Long
Dim lngTSatelitteBarTypeB As Long

Dim dblLMetric As Double
Dim dblTMetric As Double
Dim dblXMetric As Double
Dim dblTpMetric As Double
Dim dblSatelitteTypeAMetric As Double
Dim dblSatelitteTypeBMetric As Double

Dim lngCompenentTypes As Long

With SourceShape
    lngSLComponent = .NoOfLComponents
    lngSTComponent = .NoOfTComponents
    lngSXComponent = .NoOfXComponents
    lngSTpComponent = .NoOfTpComponents
    lngSSatelitteBarTypeA = .NoOfSatelliteBarsTypeA
    lngSSatelitteBarTypeB = .NoOfSatelliteBarsTypeB
End With
```

```
With TargetShape
   lngTLComponent = .NoOfLComponents
   lngTTComponent = .NoOfTComponents
   lngTXComponent = .NoOfXComponents
   lngTTpComponent = .NoOfTpComponents
   lngTSatelitteBarTypeA = .NoOfSatelliteBarsTypeA
   lngTSatelitteBarTypeB = .NoOfSatelliteBarsTypeB
End With

lngCompenentTypes = 0

'calculate L, T, X, Tp metrics of components contributed to both shapes

'find L metric

If lngTLComponent > 0 Or lngSLComponent > 0 Then
   dblLMetric = GetMetricValue(SourceShape, TargetShape, itypCoreBarL)
   lngCompenentTypes = lngCompenentTypes + 1
End If

'find T metric

If lngTTComponent > 0 Or lngSTComponent > 0 Then
   dblTMetric = GetMetricValue(SourceShape, TargetShape, itypCoreBarT)
   lngCompenentTypes = lngCompenentTypes + 1
End If

'find X metric

If lngTXComponent > 0 Or lngSXComponent > 0 Then
   dblXMetric = GetMetricValue(SourceShape, TargetShape, itypCoreBarX)
   lngCompenentTypes = lngCompenentTypes + 1
End If

'find Tp metric

If lngTTpComponent > 0 Or lngSTpComponent > 0 Then
   dblTpMetric = GetMetricValue(SourceShape, TargetShape, itypCoreBarTp)
   lngCompenentTypes = lngCompenentTypes + 1
End If

'find  bar type A metric

If lngTSatelitteBarTypeA > 0 Or lngSSatelitteBarTypeA > 0 Then
   dblSatelitteTypeAMetric = GetMetricValue(SourceShape, TargetShape,
itypSatelliteBarTypeA)
```

```
    lngCompenentTypes = lngCompenentTypes + 1
End If

'Find bar type B metric

If lngTSatelitteBarTypeB > 0 Or lngSSatelitteBarTypeB > 0 Then
    dblSatelitteTypeBMetric = GetMetricValue(SourceShape, TargetShape,
itypSatelliteBarTypeB)
    lngCompenentTypes = lngCompenentTypes + 1
End If

GetTypeOfComponentsMetric = (dblLMetric + dblTMetric + dblXMetric + dblTpMetric
+ dblSatelitteTypeAMetric + dblSatelitteTypeBMetric) / lngCompenentTypes

End Equation
```

Get Similarity Metric Value

```
Public Equation GetSimilarity(ByRef SourceShape As clsShape, ByRef TargetShape As
clsShape, _
                        ByVal lngCyclicWeight As Long, _
                        ByVal lngNoOfComponentsWeight As Long, _
                        ByVal lngTypesOfComponentsWeight As Long, _
                        ByVal lngMCSRWeight As Long, _
                        ByVal lngSatelliteBarsTypeAWeight As Long, _
                        ByVal lngNoOfLeavesWeight As Long) As Double

'returns similarity between two shapes

Dim dblCyclic As Double
Dim dblNoOfComponents As Double
Dim dblTypeOfcomponents As Double
Dim dblMCSR As Double
Dim dblSatelliteBarsA As Double
Dim dblNoOfLeaves As Double
Dim lngTotalWeight As Long
Dim dblTotalValue As Double

lngTotalWeight = lngCyclicWeight + lngNoOfComponentsWeight +
lngTypesOfComponentsWeight + lngMCSRWeight + lngSatelliteBarsTypeAWeight +
lngNoOfLeavesWeight

If lngCyclicWeight > 0 Then
    dblCyclic = GetCyclicMetric(SourceShape, TargetShape)
```

```
Else
   dblCyclic = 0
End If

If lngNoOfComponentsWeight > 0 Then
   dblNoOfComponents = GetNoOfComponentsMetric(SourceShape, TargetShape)
Else
   dblNoOfComponents = 0
End If
If lngNoOfLeavesWeight > 0 Then
   dblNoOfLeaves = GetNoOfLeavesMetric(SourceShape, TargetShape)
Else
   dblNoOfLeaves = 0
End If

If lngMCSRWeight > 0 Then
   dblMCSR = GetMCSRMetric(SourceShape, TargetShape)
Else
   dblMCSR = 0
End If

If lngTypesOfComponentsWeight > 0 Then
   dblTypeOfcomponents = GetTypeOfComponentsMetric(SourceShape, TargetShape)
Else
   dblTypeOfcomponents = 0
End If

If lngSatelliteBarsTypeAWeight > 0 Then
   dblSatelliteBarsA = GetSatelliteBarsTypeAMetric(SourceShape, TargetShape)
Else
   dblSatelliteBarsA = 0
End If

'calculate similarity

dblTotalValue = dblCyclic * lngCyclicWeight + dblNoOfComponents *
lngNoOfComponentsWeight + dblTypeOfcomponents * lngTypesOfComponentsWeight
+ dblMCSR * lngMCSRWeight + dblSatelliteBarsA * lngSatelliteBarsTypeAWeight +
dblNoOfLeaves * lngNoOfLeavesWeight
GetSimilarity = (dblTotalValue / lngTotalWeight) * 100

End Equation
```

'Returns similarity between two shapes

```
Dim dblCyclic As Double
Dim dblNoOfComponents As Double
Dim dblTypeOfcomponents As Double
Dim dblMCSR As Double
Dim dblSatelliteBarsA As Double
Dim dblNoOfLeaves As Double
Dim lngTotalWeight As Long
Dim dblTotalValue As Double

lngTotalWeight = lngCyclicWeight + lngNoOfComponentsWeight +
lngTypesOfComponentsWeight + lngMCSRWeight + lngSatelliteBarsTypeAWeight +
lngNoOfLeavesWeight

If lngCyclicWeight > 0 Then
   dblCyclic = GetCyclicMetric(SourceShape, TargetShape)
Else
   dblCyclic = 0
End If

If lngNoOfComponentsWeight > 0 Then
   dblNoOfComponents = GetNoOfComponentsMetric(SourceShape, TargetShape)
Else
   dblNoOfComponents = 0
End If

If lngNoOfLeavesWeight > 0 Then
   dblNoOfLeaves = GetNoOfLeavesMetric(SourceShape, TargetShape)
Else
   dblNoOfLeaves = 0
End If

If lngMCSRWeight > 0 Then
   dblMCSR = GetMCSRMetric(SourceShape, TargetShape)
Else
   dblMCSR = 0
End If

If lngTypesOfComponentsWeight > 0 Then
   dblTypeOfcomponents = GetTypeOfComponentsMetric(SourceShape, TargetShape)
Else
   dblTypeOfcomponents = 0
End If

If lngSatelliteBarsTypeAWeight > 0 Then
```

```
   dblSatelliteBarsA = GetSatelliteBarsTypeAMetric(SourceShape, TargetShape)
Else
   dblSatelliteBarsA = 0
End If

'calculate similarity

dblTotalValue = dblCyclic * lngCyclicWeight + dblNoOfComponents *
lngNoOfComponentsWeight + dblTypeOfcomponents * lngTypesOfComponentsWeight
+ dblMCSR * lngMCSRWeight + dblSatelliteBarsA * lngSatelliteBarsTypeAWeight +
dblNoOfLeaves * lngNoOfLeavesWeight
GetSimilarity = (dblTotalValue / lngTotalWeight) * 100

End Equation
```

Appendix C Software Operations

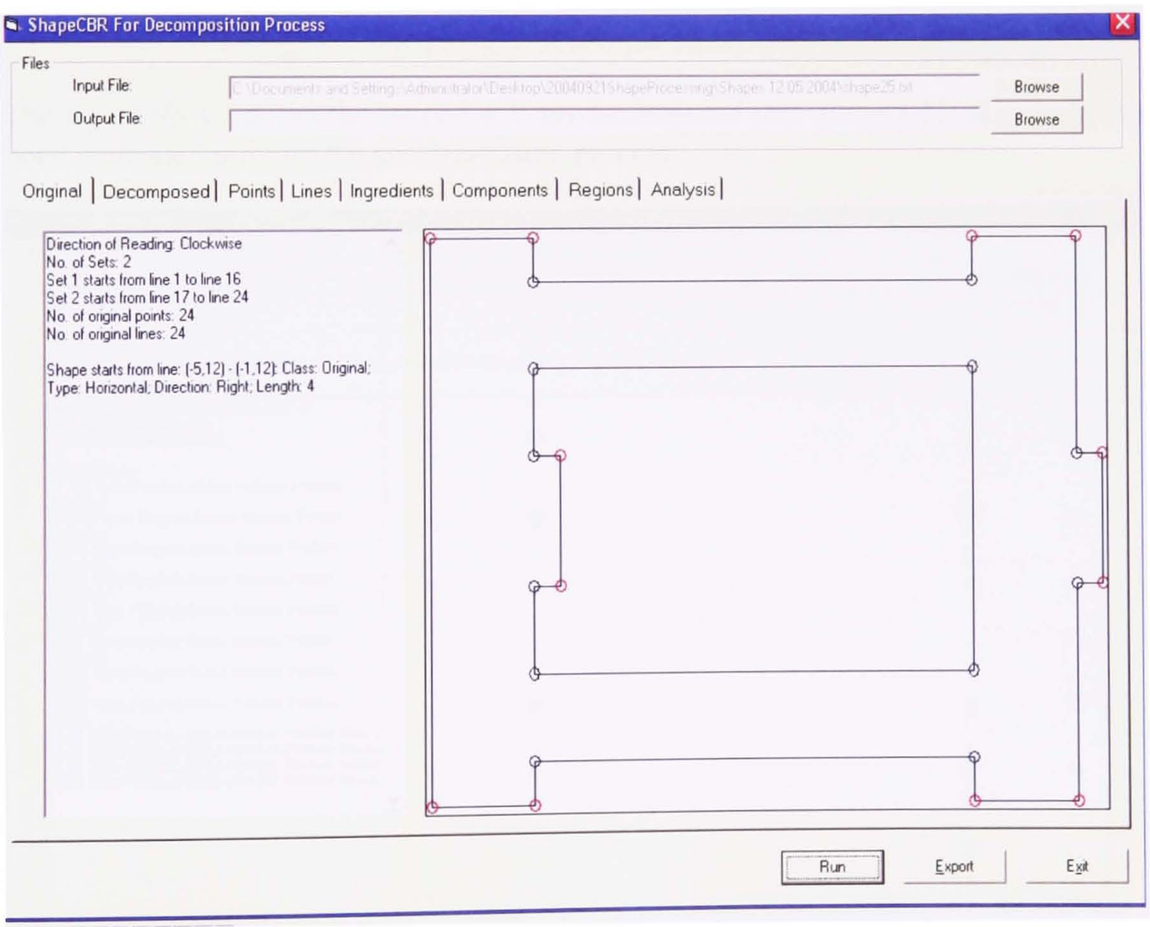
ShapeCBR- Decomposition and Classification Process

ShapeCBR have been developed in visual basic 6 programming language at the Greenwich University. The system has been used to support research objectives.

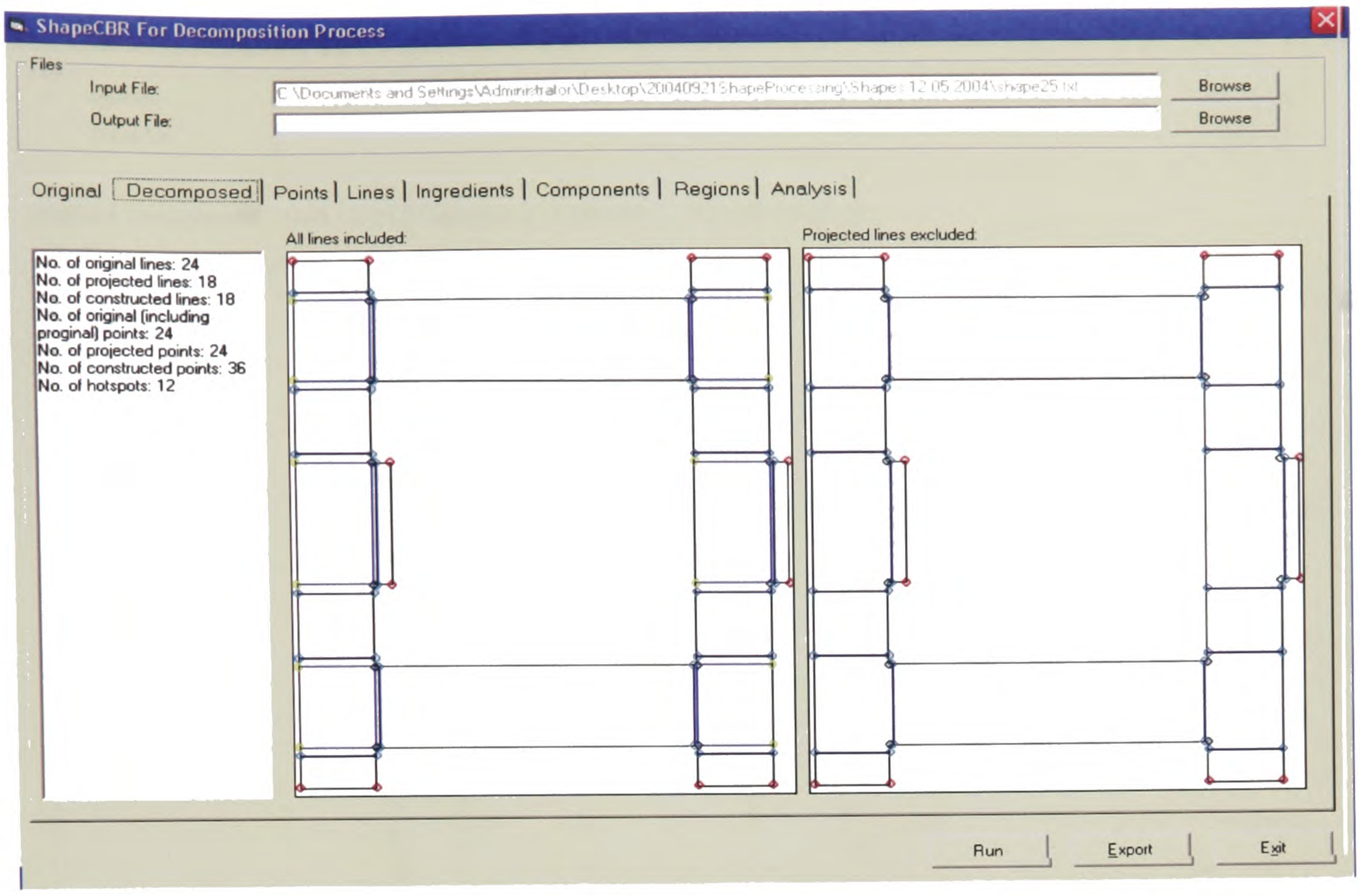
Appendix C presents the decomposition process by illustrating an example view for shape-Id 99.

3D casting design have been drawn and sliced into dissimilar views by CAD application.

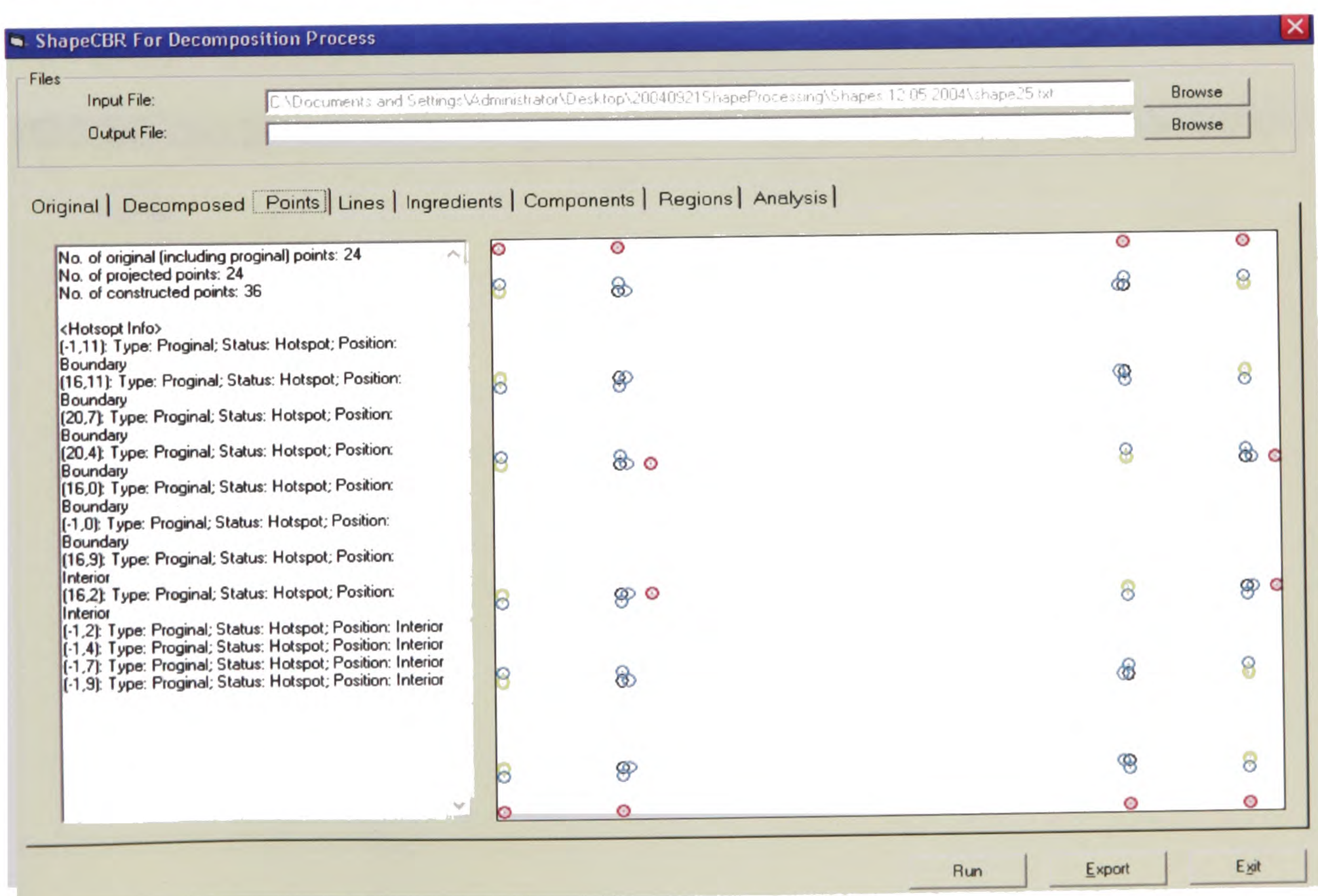
1- The user insert a new case (view 99) into the system to find the most similar case from case base knowledge .The ShapeCBR system task is to do the following processing:



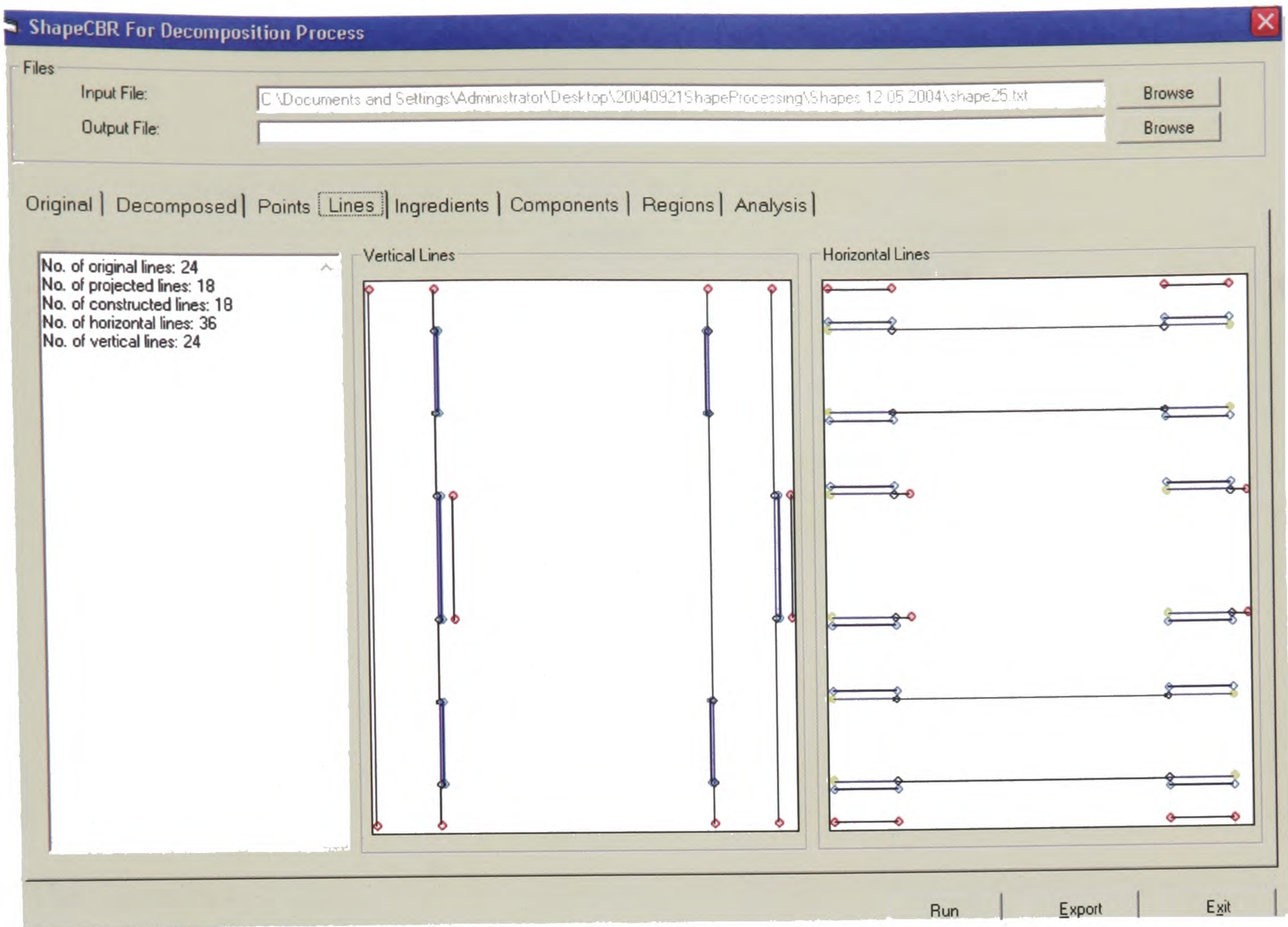
The figure above show the shape in the right side of the ShapeCBR system interface and it is ready to go through the decomposition process.



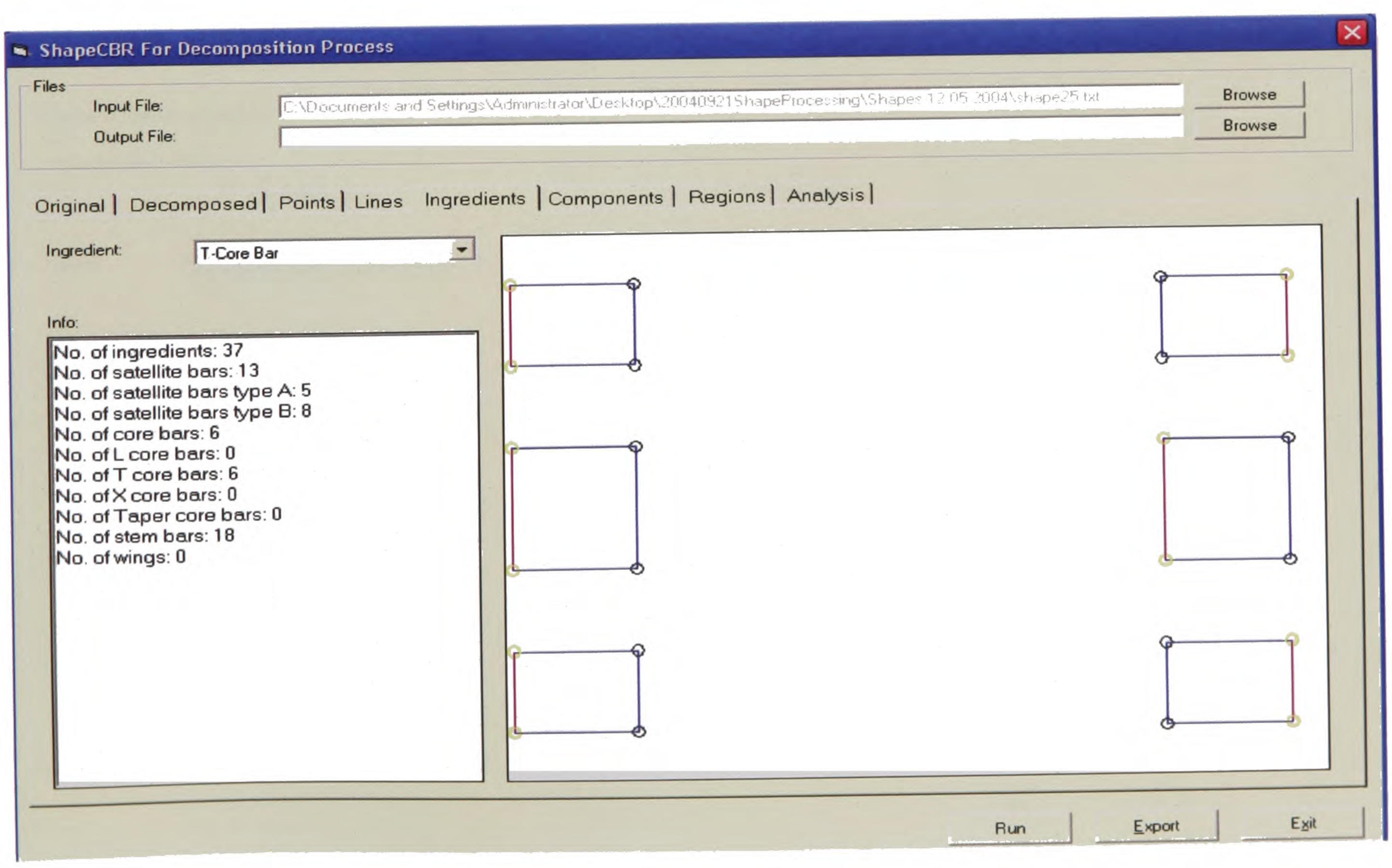
The figure above shows the shape has been decomposed into rectangular shapes that have been generated through the decomposition process.



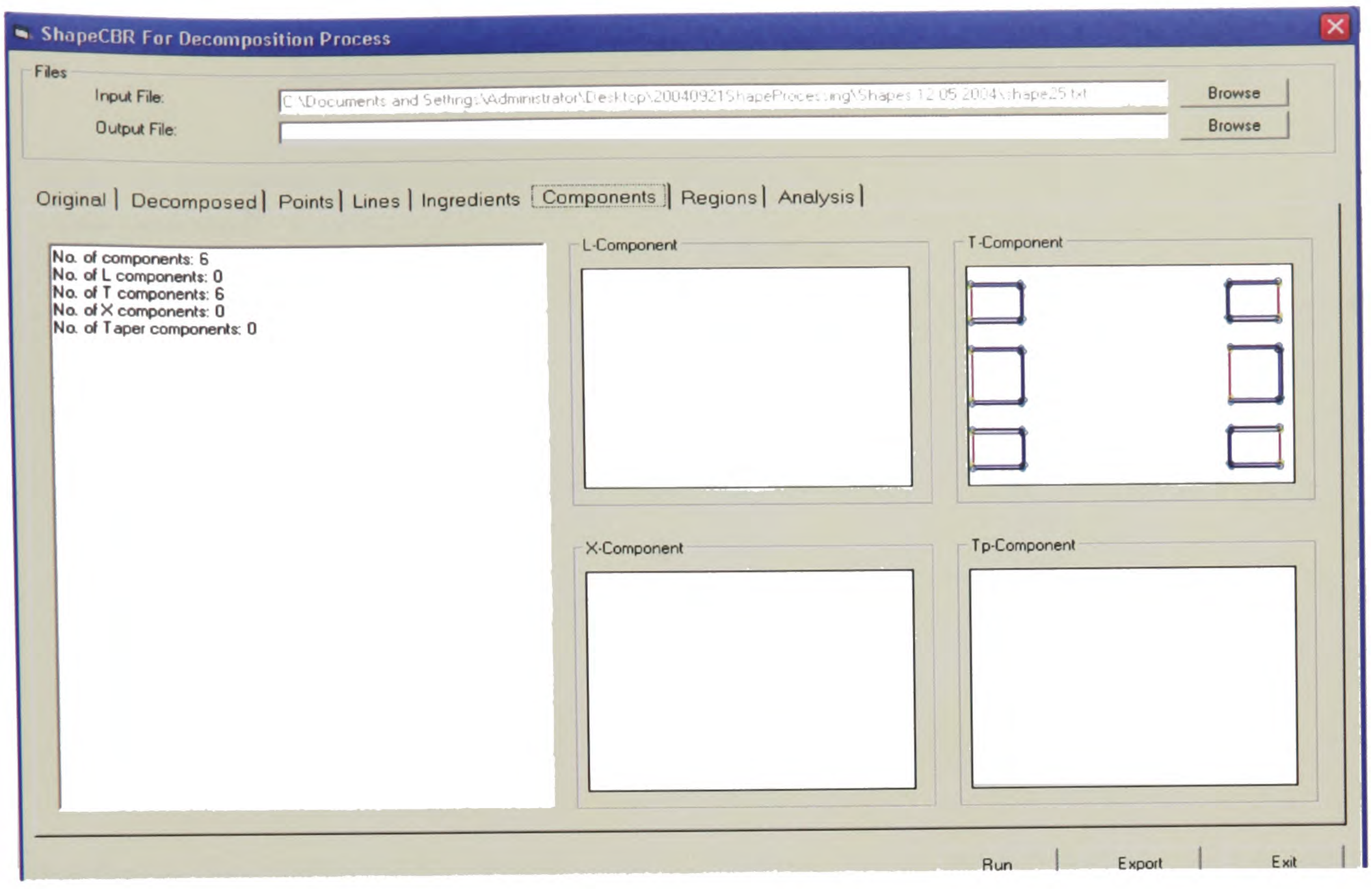
The figure above shows the point analysis for the shape that has been generated through the decomposition process.



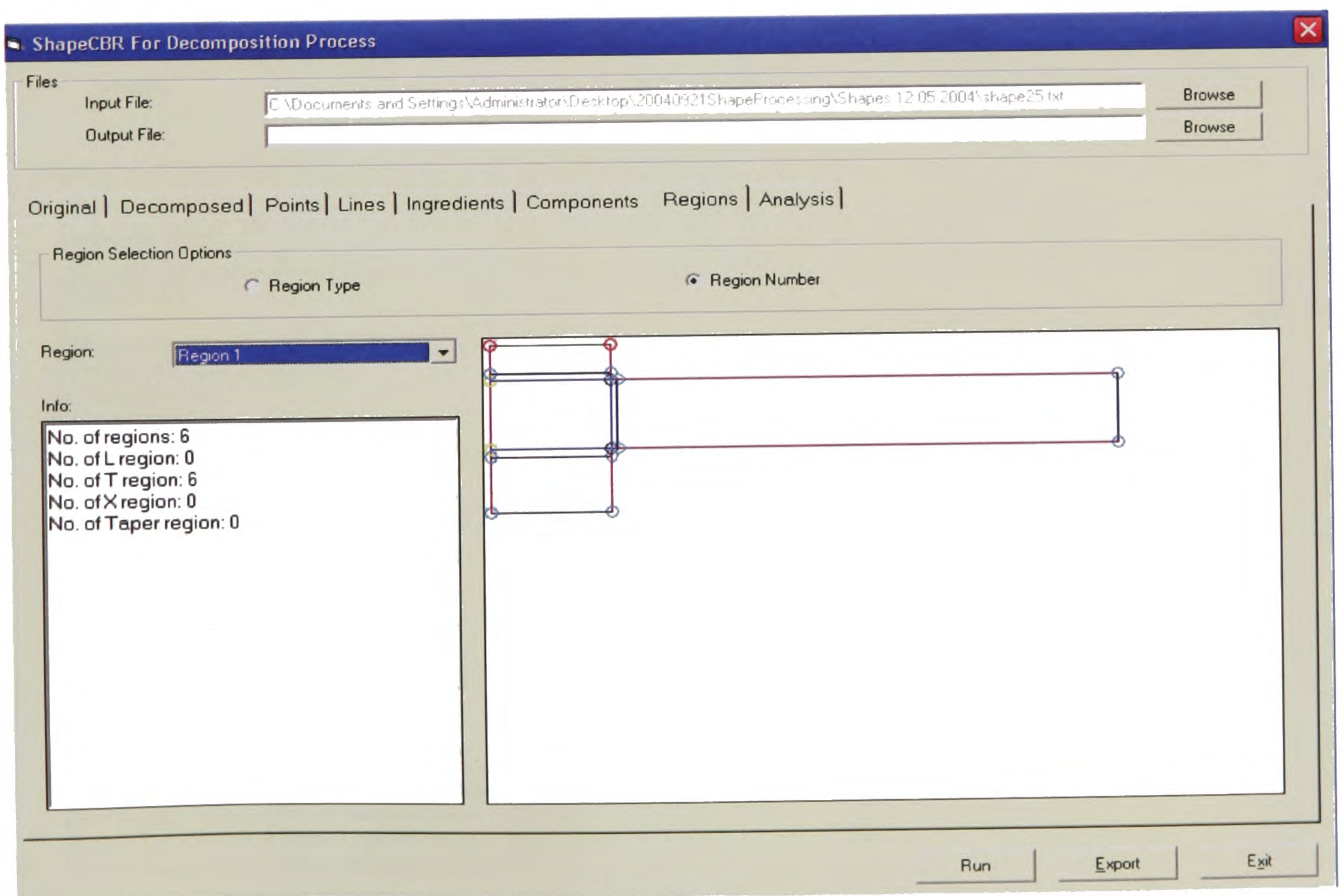
The figure above shows the vertical lines and horizontal lines type analysis that have been generated through the decomposition process.



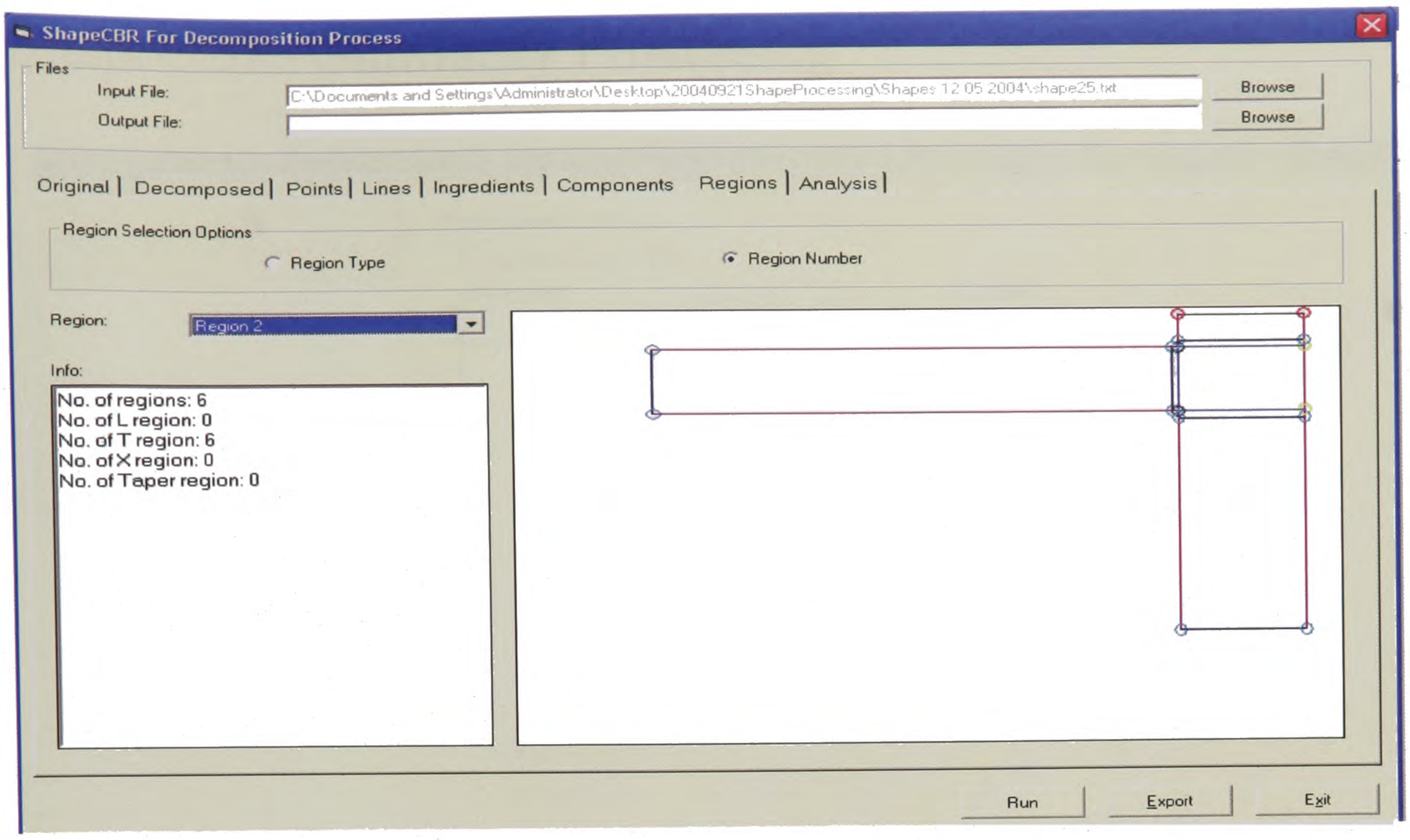
The Figure above shows the T-core-bars that have been generated through the decomposition process.



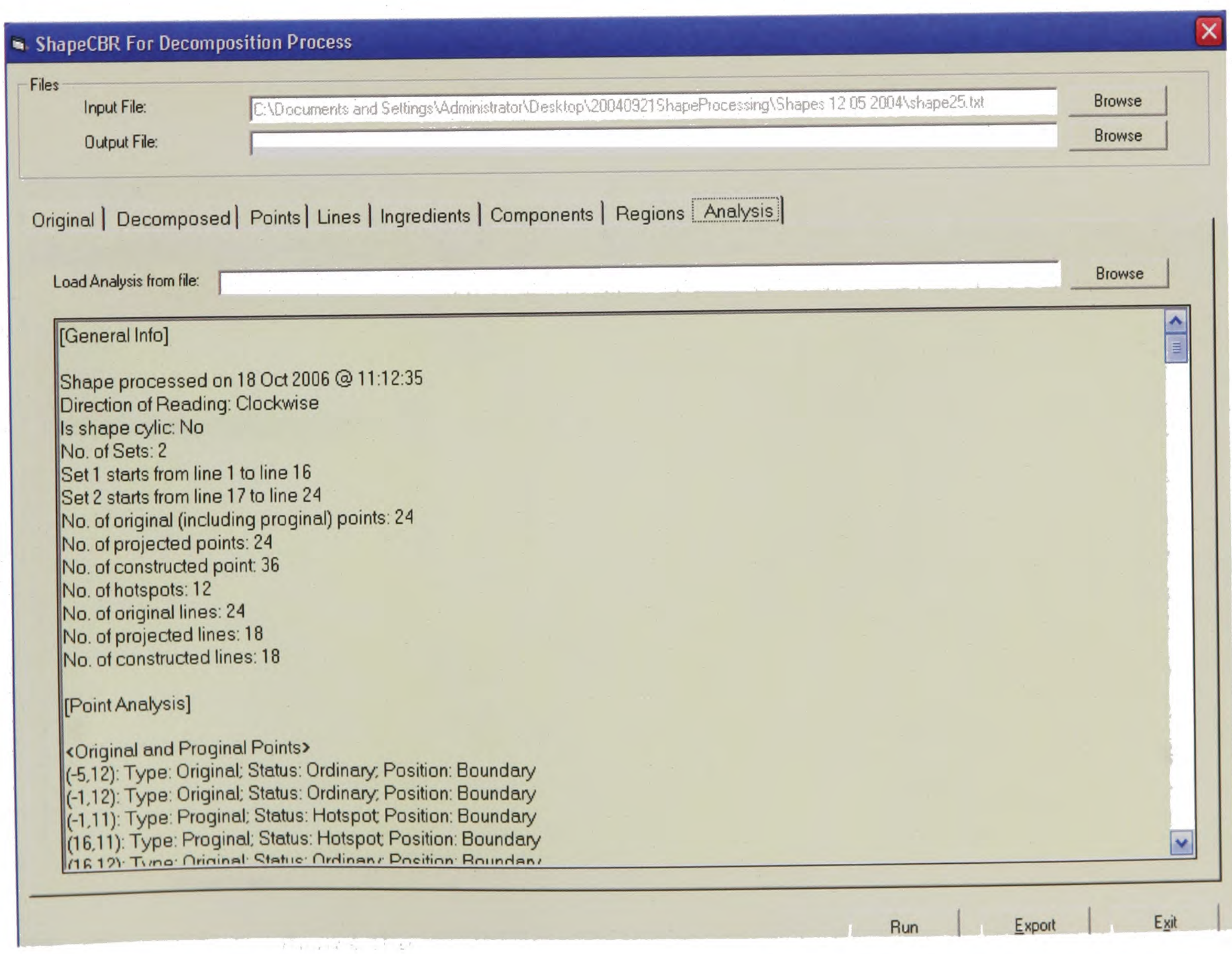
The figure below shows the T-component types that have been generated through the decomposition process.



The figure above shows the first region for the T-components which is made up from 3bars and one T-component.

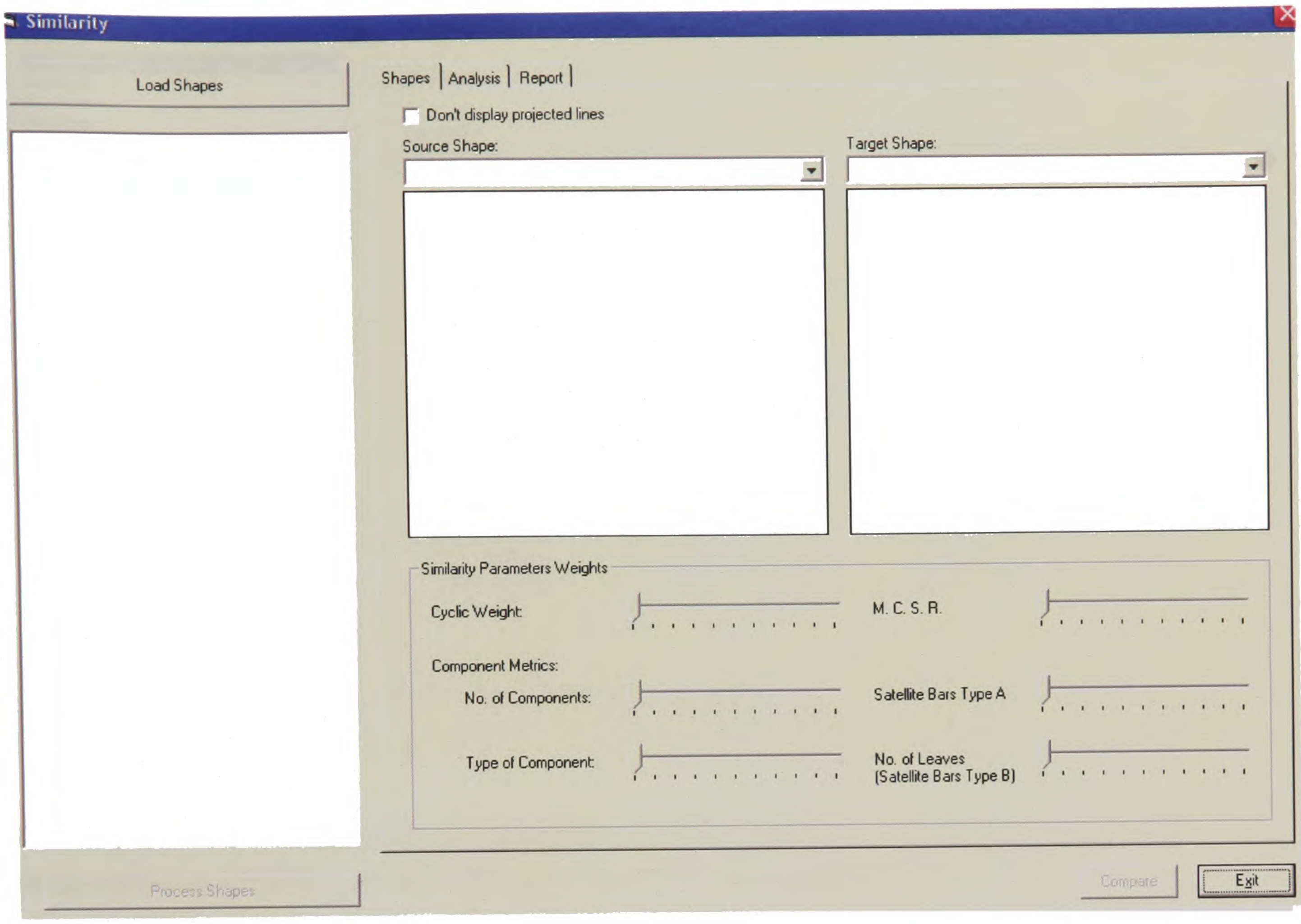


The figure above shows the second region of the T-components which is made up from 3bars and one T-component

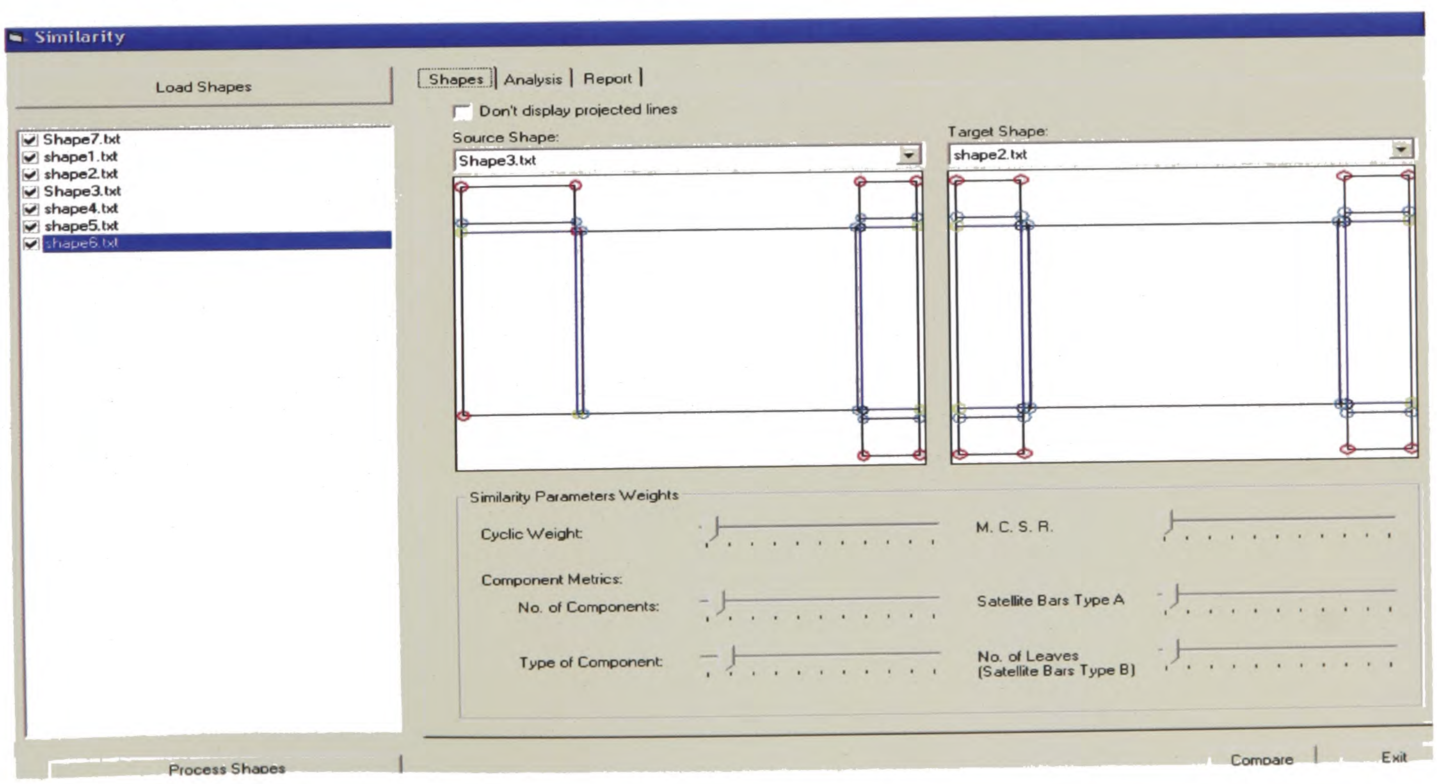


The figure above shows the final analysis report in shape decomposition process.

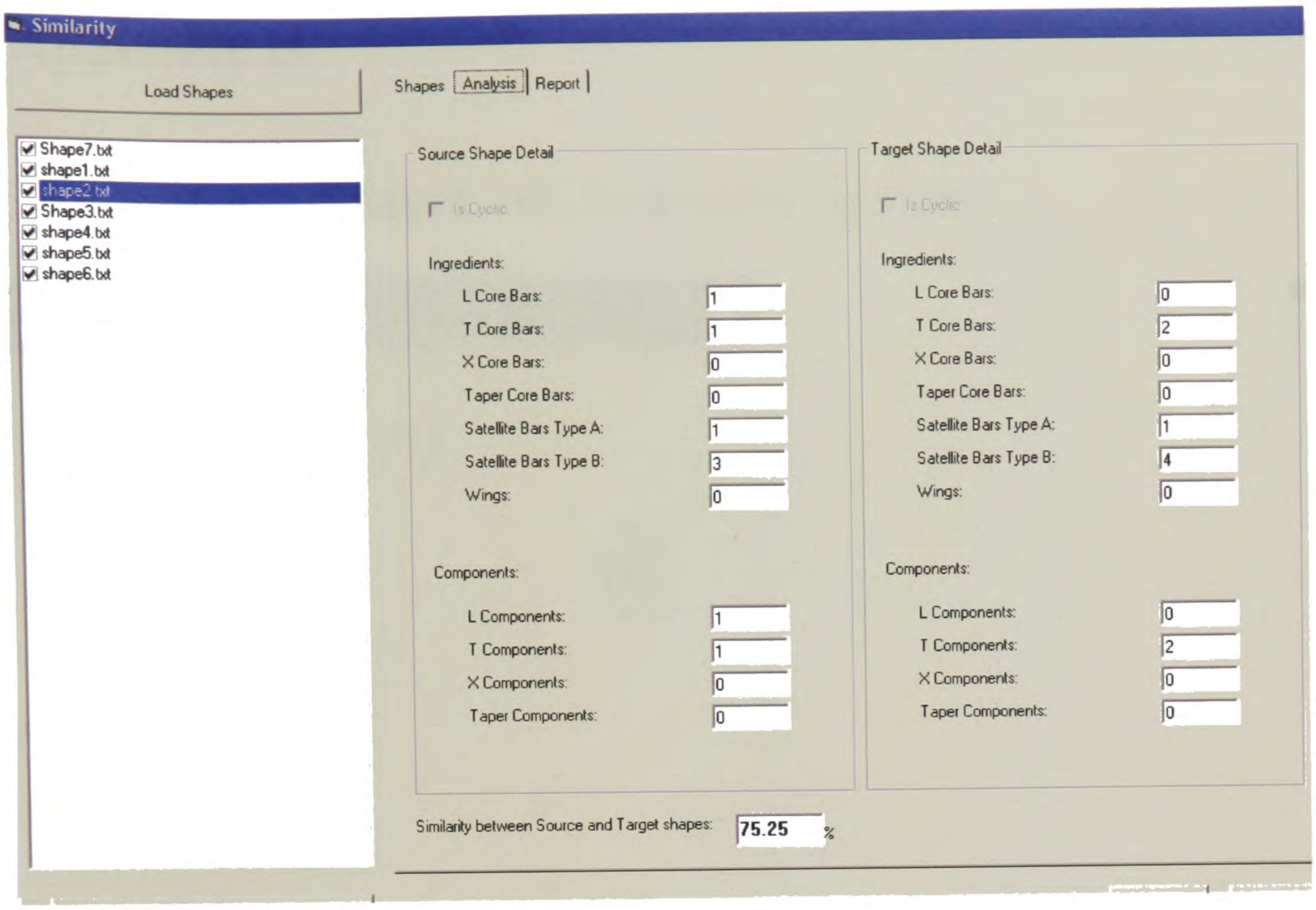
ShapeCBR- Similarity Processing



The Figure above shows the interface of similarity process.



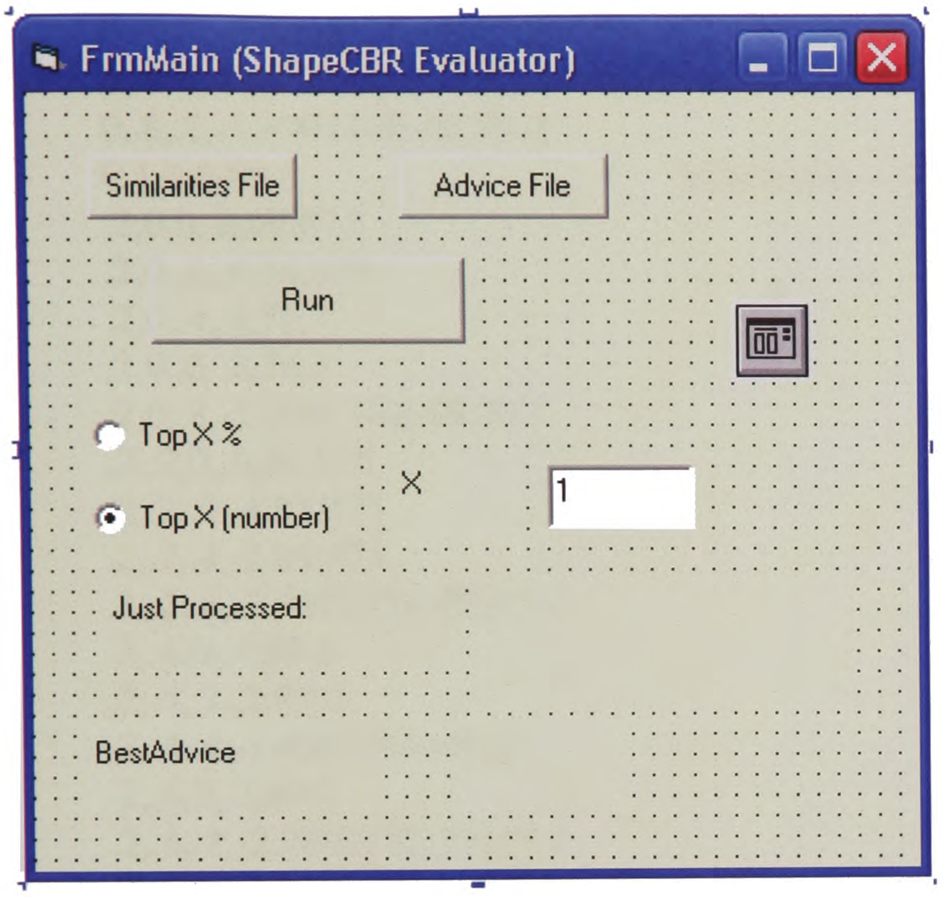
The figure above shows the target case 2 is given and the most similar one was case3 and the result can be found in the next figure below.



This figure shows the result of similarity 75.25% when we compared the target components with the sources without size of the shapes and it is 75%.

Appendix D

ShapeCBR Evaluator System



The figure above shows ShapeCBR Evaluator Interface.

The results below for the first test(1) comparing 20 target cases with 100 source cases and shows only the top one case gives the advice for individual metric in terms the feeders, the chills, the orientation and general advice if it needs.

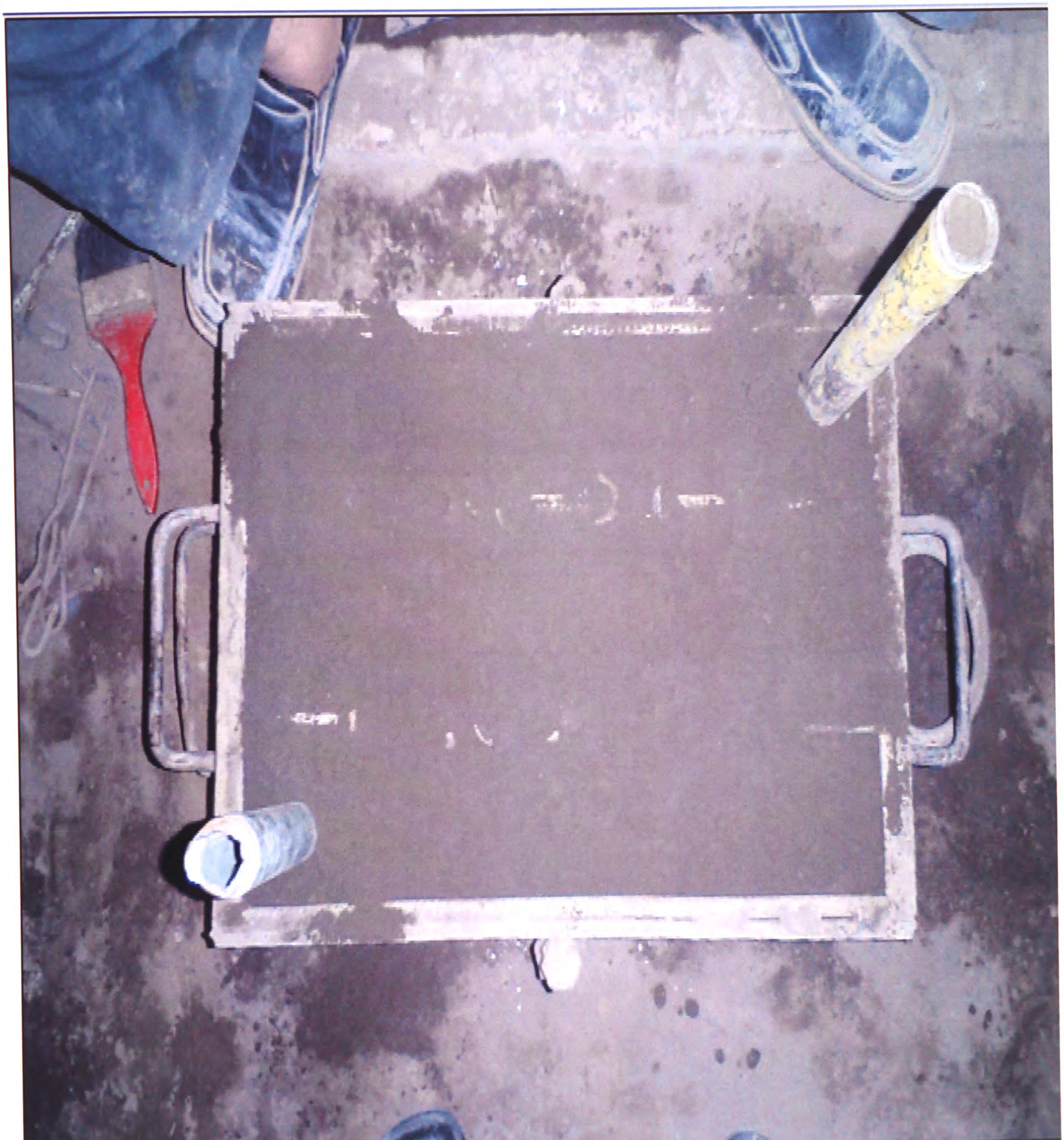
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
#1899-12-30 13:34:32#
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
0,0,0,1,74.375 Leaves metric
0,0,.2,.8,72.3125
0,0,.4,.6,72.3125
0,0,.6,.4,72.3125
0,0,.8,.2,72.3125
0,0,1,0,71.875 Cycle metric
0,.2,0,.8,89.2
0,.2,.2,.6,89.2
0,.2,.4,.4,86.5375
0,.2,.6,.2,86.5375
0,.2,.8,-1.490116E-08,86.5375

0,.4,0,.6,89.2
0,.4,.2,.4,89.2
0,.4,.4,.2,89.2
0,.4,.6,-2.980232E-08,86.5375
0,.6,0,.4,89.2
0,.6,.2,.2,89.2
0,.6,.4,-2.980232E-08,89.2
0,.8,0,.2,89.2
0,.8,.2,-1.490116E-08,89.2
0,1,0,0,89.2 Maximum Common Subgraph (MCS)
.2,0,0,.8,69.875
.2,0,.2,.6,74.625
.2,0,.4,.4,76.7
.2,0,.6,.2,76.7
.2,0,.8,-1.490116E-08,76.7
.2,.2,0,.6,86.075
.2,.2,.2,.4,85.875
.2,.2,.4,.2,85.875
.2,.2,.6,-2.980232E-08,83.2
.2,.4,0,.4,89.2
.2,.4,.2,.2,89.2
.2,.4,.4,-1.490116E-08,89
.2,.6,0,.2,89.2
.2,.6,.2,-2.980232E-08,89.2
.2,.8,0,-1.490116E-08,89.2
.4,0,0,.6,69.875
.4,0,.2,.4,79.375
.4,0,.4,.2,74.625
.4,0,.6,-2.980232E-08,74.625
.4,.2,0,.4,86.5
.4,.2,.2,.2,86.5
.4,.2,.4,-1.490116E-08,86.5
.4,.4,0,.2,86.075
.4,.4,.2,-1.490116E-08,85.875
.4,.6,0,-2.980232E-08,89.2
.6,0,0,.4,69.875
.6,0,.2,.2,79.375
.6,0,.4,-2.980232E-08,79.375
.6,.2,0,.2,85.25
.6,.2,.2,-2.980232E-08,85.25
.6,.4,0,-2.980232E-08,86.5
.8,0,0,.2,69.875
.8,0,.2,-1.490116E-08,79.375
.8,.2,0,-1.490116E-08,80.875
1,0,0,0,69.875 Component Type Metric (CTM)

For the test2 only the "Component Type Metric "CTM value have been changes, because the geometrical for the features have been added to the component types of the shapes and the result for CTM is: 83.45. This result is improvement if you compared with previous research result that the geometrical information of the shapes is important and has to be considered.

Appendix E

The two pictures below (in page 187 and page 188) show the metal casting processing. The first picture shows the two feeders sticking out and the sand covers the mould to cool down the metal during solidification process.

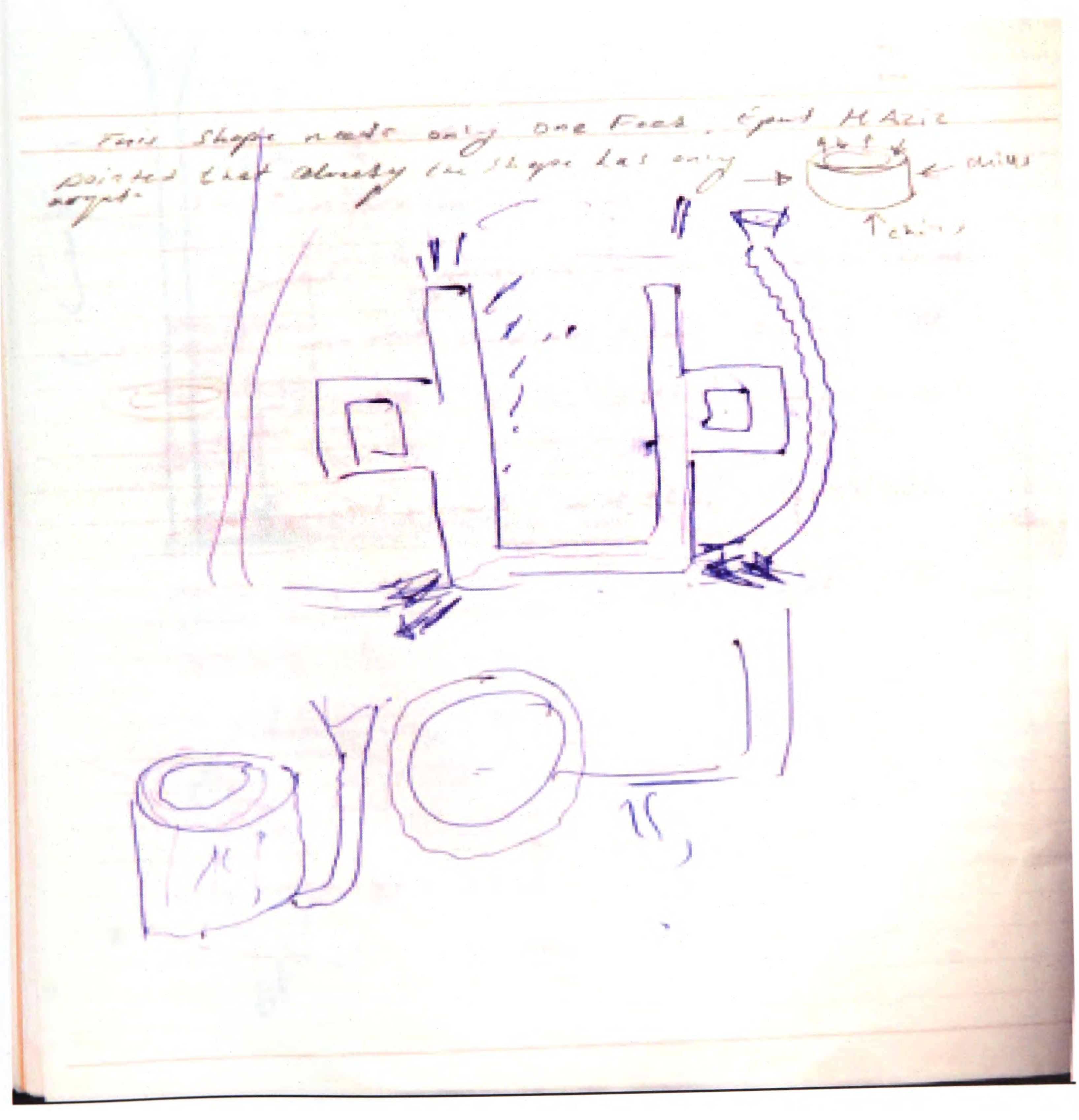


This picture shows the part has been solidified.

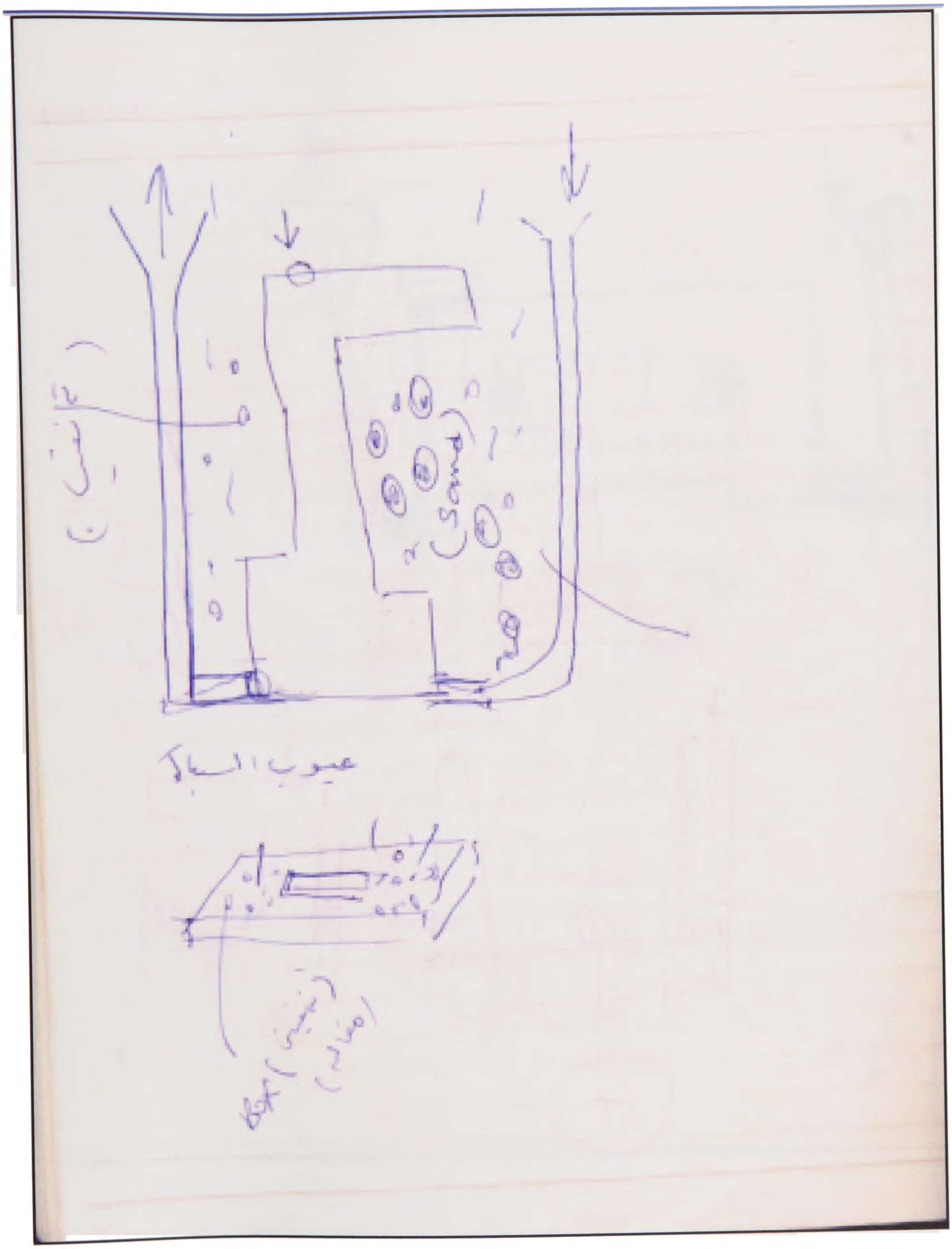
Some Hand Sketch for Metal Casting Processing by the expert [Aziz M. 2004].

The sketches below (1, 2, 3 and 4) shows examples of shapes annotated with metal casting advice such as number of the feeder, number of the chills and general advice by expert in casting design “Aziz” [Aziz, M. 2003].

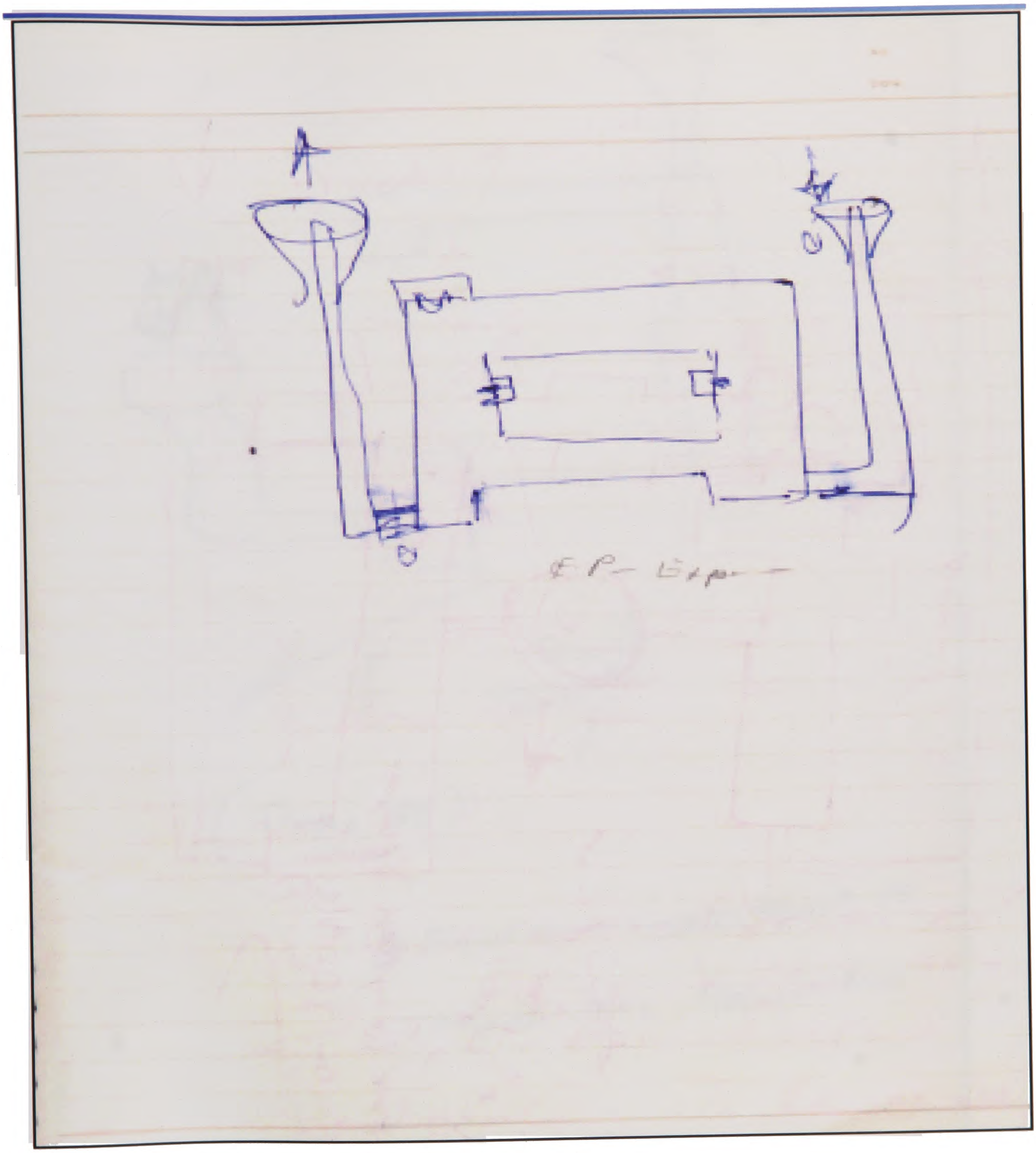
Sketch (1)



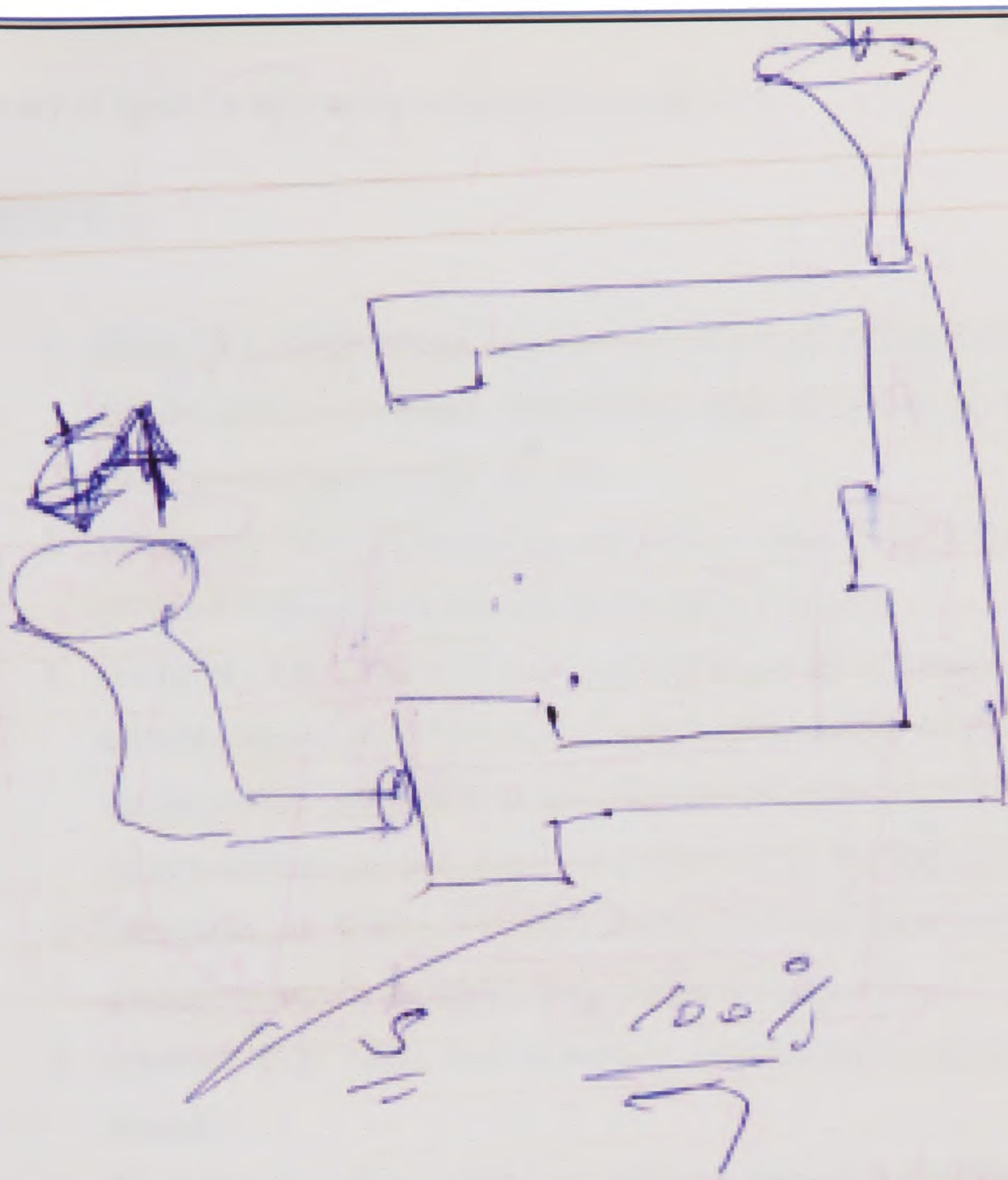
Sketch (2)



Sketch (3)



Sketch (4)



(Shape 76)

3 Feeder → equal to

expert Mr Mohamed Anwar

Appendix F Research "Citations"

Summary of notes on references organised by chapters:

Chapter 1

1. [Pegler C.J.1993] Design histories are often lost, or banished to paper files that are difficult to search. Also, design engineers retire, or move away leaving inadequate design records.
2. [Campbell, 1991] Foremost among these is shrinkage in the mould, which can give rise to porosity and areas of structural weakness.
3. [Jolly, M. 1996] found in his survey that the foundry industry is looking for software applications that can not only predict problems that occur during metal solidification (such as shrinkage porosity) but also, having predicted these problems, propose intelligent solutions for the problems found.
4. [Hennessy, D. Hinkle, D 1992] Intelligent knowledge-based systems (IKBS) attempt to support an earlier stage in the design process.
5. [Corbett, C.F. 1989] And numerical simulations based on physical process models.
6. There is some of the commercial software packages available in market can calculate the position of feeders such as:
   - NOVACAST [NovaCast: Sillen, R.1991] analyse geometric properties.
   - AutoCast [Ravi, B.1999] give suggestions to further improve the design.
7. [Marir, F., Watson 1994-15] Although many prototype tools have demonstrated the efficacy of CBR in the domain of engineering and design, there is an insufficiency of research for its use in the foundry industry.
8. [Price, C.J. et al. 1997] One commercial CBR system called Wayland is used for the setting of parameters in pressure die-casting.
9. [Biederman et al 1992] General classifications of shape components have been proposed; for example, Biederman's *geons*.

10. [Knight B. et al, 1995, and Wlodawer, R: 1967] a decomposition of shapes specific to the casting industry already existed in practice.
11. [Mileman: 2000] proved that useful casting advice from the ShapeCBR system could be retrieved from similar designs

Chapter 2

1. [Moore: 1993] This is also the case when designing where over 50% of the work on a day-to-day basis is routine design that consists of modifying past solutions.
2. [Kolodner: 1993] Making use of past experience in the form of cases is commonly known as Case-Based Reasoning (CBR).
3. [Maher, et al 1997, Rivard and Fenves, 2000] The application of CBR in design, known as Case-Based Reasoning Design, is still in its infancy even though several CBR systems focusing on various domains have been developed.
4. [Schank, 1982, 1999] Case-Based Reasoning (CBR) originates from the cognitive observation that humans often rely on past experience to solve new problems.
5. [Kolodner, 1993] The premise of dynamic memory is that remembering; understanding, experiencing and learning cannot be separated from each other.
6. [Flemming and Woodbury 1995] It should also be pointed out that unless the CBR process becomes more or less automatic, the designers would be reluctant to add potentially useful cases to the case-base or to try to reuse old cases.
7. (Aziz: 2004)Solid shrinkage changes the dimension of the casting from those in the mould to those dictated by the rate of solid shrinkage for the shapes.
8. [Mileman: 2000] This is an advantage allowing CBR systems to realise casting know-how as a valuable asset.

9. Computer Flow Dynamic (CFD) [Cleary et al.: 2003], it is another tool using numerical methods representing the fundamental physical processes occurring during many researchers used this method for casting design problem and many applications have been developed for this problem.
10. [Kolodner et al: 1994] As such, this approach comes with a cost and great difficulty for the user, and the recent one, case-based reasoning (CBR) discovered early 90's and was raised by researches for the first time to solve this type of problem.
11. [Moore 1993] CBR is the cheap and easy way to run. More than half of the daily work done by designers is routine design that consists of modifying past work.
12. CYCLOPS [Navinchandra 1991] which supports landscape layout.
13. JANUS ([Fischer and Nakakoji 1991], kitchen design.
14. FABEL [Consortium 1993] and a case base of construction component.
15. SEED is a system environment which aims at providing computational support for the early phases in building design.
16. (Flemming and Woodbury, 1995) The tasks supported at the present time are architectural programming, schematic layout design and the generation of a fully 3-dimensional configuration of physical building components.
17. [Oxman: 1994] recognises four cognitive approaches for modelling design case knowledge:

a) **Generic models:** Knowledge is used to define classes of designs called generic designs.

b) **Associative models:** The associative mechanism is another key principle of cognition, which is present in design thinking.

c) **Exemplar models:** In this approach it is attempted to re-use past knowledge rather than to generate new designs.

d) The design precedent: the selection process of relevant ideas from prior designs in current design situations has been termed precedent-based design.

18. Several software tools may be used to assist the methoding process:
    a) SOLSTAR [SOL], which support the initial design stages,
    b) SIMULOR [SIM], which support the simulation stages
    c) CRUSADER give numerical support on such aspects as feeder sizes and feeder-feeder distances, but do not attempt to give experiential advice on such elements as re-design for casting, or mould orientation.
    d) [Cleary et al.: 2003], more advanced numerical software (SPH), using computational fluid dynamics techniques which, support the simulation stages of die filling predictions is very high and the last locations to fill correlate well with porosity void age observations made by manufactures of these components.
19. Insufficient capabilities shift attention from design to equipment and programs and the work itself suffers [Heikkonen 1995].
20. Also the wrong basis for CAD investments has led to poor results and caused a negative attitude towards information technology on a wider scale [Naaranoja 1997].
21. Building specifications and other text documents are, however, produced with separate computer applications, at least so far [Kiviniemi & Penttilä 1995].
22. The space model describes the real object [Davies *et al.* 1991, Holvio 1993, and Medland 1988].
23. Rendering means producing colored and shaded pictures. Color, brightness, material and transparent features, lights and shadows are added into space models [Kiviniemi and Penttilä 1995].
24. Nonaka [Nonaka: 1998] states that in an economy where the only certainty is uncertainty, the one source of lasting economical advantage is knowledge. Knowledge management (KM), as defined by the Gartner Group (www.gartnergroup.com), is a discipline with new processes and technologies that differentiate it from information management
25. Knowledge is reasoning about information and data to actively enable

performance problem solving, and decision-making, learning and teaching [Beckman: 1999].

26. KE) has limited use for the range and complexity of design tasks. Debenham [Debenham 1998:1] states that a unified KE methodology treats data, information and knowledge in a standardized mode.
27. [Oxman: 1994]. An expert system is a system in which knowledge is represented as it is, possibly in the same form that it was extracted from an expert.
28. [Debenham 1998] defines a Knowledge-Based System as a system that represents an application containing a significant amount of real knowledge and has been designed, implemented and possibly maintained with due regard for the structure of the data, information and knowledge.
29. [Debenham 1998:23] identifies differences between Knowledge-Based Systems and expert systems where as Expert Systems perform in the way of a particular trained expert. A knowledge-based system is not constrained in this way.
30. Due to the complexity of design, systems for design have often defined the task with artificial narrowness [Hinrichs 1991:3].

31. To make the systems tractable the following typical four approaches were used [Hinrichs 1991:3]:

    a) Selection. Select components to instantiate a skeletal design.

    b) Configuration. Arrange a given set of components.

    c) Parametric. Fix numeric parameters.

    d) Constructive. Build up designs from components.

32. Hinrichs summarises some of these approaches as:

    a) ***Pure synthesis*:** construct designs from the bottom up.

    b) ***Hierarchical refinement*:** refine skeletal designs from the top down.

    c) ***Transformational approach*:** mapping from equation to structure.

d) ***Case-Based Design***: the case-based and analogical approaches assume that the problem being solved is probably similar to one that was seen before.

33.[Kolodner, 1993] suggested CBR depends on the method of parameter adjustment for interpolating values in a new solution based on those from an old shape.

34. [Mileman: thesis 2000] research has demonstrated the feasibility of a CBR system to assist the casting design process, but the work in that research did not show how the process can be automated and also had some limitations such as efficiency of retrieval and not dealing with 3D shapes.

Chapter 3

1. A vast number of researchers (Mileman: 2000, Kotschi and Plutshack 1981) have simplified the evaluation of 3D shapes by using a slicing technique to simulate 3D shapes as 2D slices.
2. While a significant amount of research for shape segmentation or decomposition of 2D shapes has been conducted over the last two decades [Hoffman et al 2000], little effort has been made on shape segmentation of 3D models [Rom H. and Medioi G.:1994] Rom proposed *"a framework consisting of decomposing 3D objects into single components and then describing those parts by higher-level primitives, such as generalised cylinders"*.
3. Wu [Wu K.:1997] presented a physics-based part segmentation approach.
4. [Mangan A. et al 1999], alternatively a surface segmentation decomposing method based on either planar surfaces or arbitrary shapes.
5. [Wu K.:1997] The disadvantage of this method is the limited usefulness of surfaces compared to equational parts in high-level tasks such as object recognition.
6. [Mileman: 2000] demonstrated the feasibility of this approach, but he used a manual approach to do this decomposition process.
7. [Mileman 2000] assumed that one 2D cross-section or view representing 3D shapes is enough to represent 3D shape.

8. [Kotchi and Plutshack 1981] 2D cross-sections or views are easier to display than 3D shapes.
9. [Kotchi and Plutshack 1981] Also 2D cross section shapes are easier to display than 3D shapes.
10. [Kotchi and Plutshack 1981] have already investigated that one objective; a way to define the geometrical complexity of a 3D shape is to ask how many different cross sections are required to describe the shape.
11. [Xuetao Li and Tong Wing Woon 2001], developed an efficient framework to decompose polygon meshes into components that adopts the idea *of edge contraction a space sweeping* to decompose into objects automatically.
12. The generalised cylinders method [Binford T.O. 1971], Geon's [Biederman I. 1987], Super-quadric [Hertel S. et al, 1984] and their extensions were used in 2D images and range data.
13. work on volumetric objects was presented by Gayvani [Gayvani and Silver 2000].
14. Decomposition of 3D (volume) digital shapes is based on a hierarchical decomposition method developed by [G. Borgefors et al, 1999].
15. Lopes A. M. and Metha P.M.: 1994] used a method that it is quite closely related to our current decomposition shapes, but only horizontal projection has been used to partition a polygon into rectangles and L-shapes and the decomposition process done manually.
16. [Tan T. S. et al 1999] argued that he achieved good results in decomposing objects through the use of vertex-based simplifications.
17. [Simmons M. and Sequin H. C.: 1998] developed an automatic system to generate a hierarchical 2D object representation especially for geometric tasks.
18. Their approach is based on the axial generation module that could be replaced by an alternate construction, like that used in producing cores [Burbeck A.C. and Pizer M.S.: 1995].

19. [Mario A. Lopes and Dineshp P. Methat 1994], presented two practical algorithms for partitioning circuit components, represented by rectilinear polygons, so that they can be stored using the L-shaped corner stitching data structure.
20. Mehta [Blust and Mehta 1993] that the data structure stores L-shaped tiles (hexagons) in addition to rectangular tiles.
21. . [Shanbhag et al. 1994; Mehta et al. 1995] This L-shaped variant of corner stitching was motivated by a need for a data structure that could store rectilinear shapes more general than rectangles
22. L-shaped objects, in particular, have been studied in the context of floor planning [Wang and Wong 1990; Yeap and Sarrafzadeh 1993] and routing [Dai et al. 1985; Cai and Wong 1993].

Chapter 4

1. Six generic components identified in previous research [Mileman: 2000] and General classifications of shape components have been proposed; for example, Biederman's *geons* [Biederman et al 1992].

Chapter 5

1. [Anandan S. and Summers D. J.:2006] Anandan proposed four distinct similarity metrics for shape retrieval in an interactive modeling environment; entity similarity, relationship similarity, attributes similarity, and structural similarity.
2. [Gebhardt, F: 1997] argues that for retrieval systems, features representing complex structures are difficult to define, and similarity must be derived from structure directly. Gebhart reviews several retrieval systems:
    a) These contain group detection as in the Fabel component Topo [Coulon, C.H.:1995],
    b) largest common subgraph [Tammer, E.C at el. 1995]
    c) and hamming distance [Bunke, and Messmer: 1994].

3. The formalism of conceptual graphs allowed the introduction of graph matching between pairs of graphs [Sowa, 1984],
4. from either a theoretical or a practical view point, in combination with matching graphs
5. [Eroh and Schultz, 1998], minimal condition subgraphs
6. [Gao and Shah, 1998], finite graphs [Bacik R: 1997],
7. Weighted mean of a pair of graphs [Bunke and Gunter, 2001].
8. Messmer and Bunke present a new graph structure which is better suited for representing parameterised image features.
9. Bunke [Bunke, 2001] Different ways of representing patterns have been analysed in terms of symbolic data structures such as strings, trees, and graphs.
10. Schemer [Schemer et al: 2003] a graph G = (V, E) in its basic form is composed of vertices and edges.
11. Graphs have been proved as an effective way of representing objects [Eshera and Fu, 1986].
12. [Fernandez and Valiente, 2001] proposes a way of representing attributed relational graphs, the maximum common subgraph and the minimum common supergraph of two graphs by means of simple constructions, which allow to obtain the maximum common subgraph from the minimum common supergraph, and vice versa.
13. A distance measure between pairs of circular edges and relations among them is introduced in [Foggia et al., 1999].
14. In [Bunke, 1997] the relation between graph edit distance and the maximum common subgraph is analysed, showing that under a metric equation for MCS graph distance computation is equivalent to solving the maximum common subgraph problem.
15. Several graph distance techniques rely on finding the "Maximum Common Subgraph" (MCS) [Schenker et al: 2003].
16. Sundra [Sundra, H.: 2003], used graph matching as a method for searching and comparing 3D objects.

17. Wang [Wang and Ishii, 1997] case-based reasoning can be applied to many areas, such as the chemical field, the bio-technical field, the multimedia field, the businesses field and the heavy industries field.
18. Gebhardt [Gebhardt, F: 1997] argue for retrieval systems representing complex structures, features are difficult to define, and similarity must be derived from structure directly.
19. (G1) and G2, which has the maximum number of nodes as compared to all the possible subgraphs of (G1) an (G2) [Anandan S. and Summers D. J 2006].
20. And the general similarity ratio is the sum of similarity ratios of features multiplied by their correspondent constancy of ratio of importance (wi). See **equation of Watson** (Watson 1996).
21. Messmer [Messmer and Bunke, 1999]. Several references can be found on performing efficiently graph matching to all the models.
22. [Williams et al., 1997] describes the development of a Bayesian framework for multiple graph matching.
23. Wilson and Hancock [Wilson and Hancock, 1996] which is generalised from *matching graph pairs to multiple graphs.*
24. [Huet and Hancock, 1999] a graph-matching technique for recognising line-pattern shapes in large image databases is described matching algorithm that uses edge-consistency and vertex attribute similarity.
25. Multiple graph matching has also been applied to many other problems such as the comparison of saliency map graphs [Shokoufandeh et al., 1999 and 2000].

Chapter 6

1. There is a body of research that looks into the evaluation of CBR systems and tools [Althoff K. D.: 1995].
2. if the weight is over 20kg, their designed casting needs to utilise one to two feeders (Aziz expert in casting design).

3. Case base has been defined by Taylor (Taylor: 1997: pp: 136) as "The memory of past experience".
4. Mileman [2000] created three equations to calculate the feeders and chill numbers between two cases.
5. These roles have been set up by a previous expert [Preddy K: 1999] and an intermediate expert (Mileman, thesis 2000).

$$\frac{2 * (\text{Matching Feeders in Target and Retrieved})}{(\text{number of Target Feeders} + \text{Retrieved Feeders})}$$