An Analogy Based Costing System For Injection Molds Based Upon Geometry Similarity With Wavelets

Cyrus Clinton Hillsman
University of Central Florida

AN ANALOGY BASED COSTING SYSTEM FOR INJECTION MOLDS BASED UPON GEOMETRY SIMILARITY WITH WAVELETS

By

CYRUS CLINTON HILLSMAN
A.A.S. Mold Making, Southeast Community College, 1987
A.A.S. Tool and Die, Southeast Community College, 1989
B.S. Finance, University of Central Florida, 2000
B.S. Management Information Systems, University of Central Florida, 2001
Graduate Certificate, Data Mining, University of Central Florida, 2004
M.S. Industrial Engineering, University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctorate of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall 2009

Major Professor: Yan Wang

Science is facts; just as houses are made of stones, so is science made of facts; but a pile of stones is not a house and a collection of facts is not necessarily science.

--Henri Poincare

A classic is something that everybody wants to have read and nobody wants to read.

--Mark Twain The Disappearance of Literature speech, 1900

So it goes when writing a dissertation. Your task is to combine what you have learned and contributed yourself into a coherent thought process that remains readable so that it might inspire others to do the same. We hope we have achieved this goal when writing this dissertation. I am solely responsible for the contents of the dissertation and any unintentional failings or inaccuracies it may contain. What is good about it could not have been achieved without the advice, consent, counsel, and motivation of many others.

I would like to thank my committee members for their inspiration, ideas, encouragement, pressure and advice.

Dr. Robert Hoekstra, Dr. Dima Nazzal, Dr. Charles Reilly, Dr. Marshall Tappen

Special thanks to my major professor Dr. Yan Wang for his countless and tireless hours spent helping me bring this dissertation to fruition.

ABSTRACT

The injection molding industry is large and diversified. However there is no universally accepted way to bid molds, despite the fact that the mold and related design comprise 50% of the total cost of an injection-molded part over its lifetime. This is due to both the structure of the industry and technical difficulties in developing an automated and practical cost estimation system. The technical challenges include lack of a common data format for both parts and molds; the comprehensive consideration of the data about a wide variety of mold types, designs, complexities, number of cavities and other factors that directly affect cost; and the robustness of estimation due to variations of build time and cost. In this research, we propose a new mold cost estimation approach based upon clustered features of parts. Geometry similarity is used to estimate the complexity of a mold from a 2D image with one orthographic view of the injection-molded part. Wavelet descriptors of boundaries as well as other inherent shape properties such as size, number of boundaries, etc. are used to describe the complexity of the part. Regression models are then built to predict costs. In addition to mean estimates, prediction intervals are calculated to support risk management.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: PROBLEM STATEMENT

Section 1.1 Definition of the problem

The injection molding industry is large and diversified. However there is no universally accepted way to bid molds. The reasons are manifold. Some of the reasons have to do with the structure of the industry. Many companies operate as small job shops and may not have the resources to develop a systematic way to bid the molds. This is one of the primary reasons why the most commonly used bidding method in actual use is ad-hoc. Any ad-hoc procedure carries significant risk and is dependant on the competency of the person doing the bid. To minimize such risks a systematic method for automated cost estimation of injection molds is needed.

However there are some significant challenges in developing a practical working system that will work well for all molders. There are two classes of problems encountered. The first class is associated with data. The historical data of parts, molds, and costs is not structured in such a way to facilitate their use for bidding. The second class of problems has to do with bidding risk handling. Some types of parts and molds have more variations in build time and cost than others. This higher variation translates into a higher risk of a missed bid.

The first data related problem is lack of a common information format for parts and molds. Studies have shown that the average molder has over 700 unique parts to mold [1]. However there may be no common format for all 700 parts and molds. A recent study by Christman [2] showed that designs are received in many formats for bidding. The most common data format was 2D CAD received in 59.2% of cases. The next most common format was 3D CAD used in 20.6% of cases followed by paper blueprints 13.0%, direct database access 6.4%, and prototype 0.7%. According to an earlier study by Fallbohmer et al. [3], full 3D CAD models are only available in less than 30% of bidding cases. There are several reasons for the different data formats. Blueprints are easier to access on the shop floor and contain not only geometry but also non-geometric information. This could be information on block tolerances, geometric dimensioning and tolerancing, surface finishes, and/or flash or gating requirements. 2D CAD could be viewed as an electronic extension of blueprints and currently has the widest user base. 3D CAD has the advantage of being able to view the part and mold from many perspectives. Given that many formats are used at the same time, data must be converted to a common format prior to the use for automated cost estimation.

The second data related problem is how to store and retrieve design and cost data automatically. The cost of molds can range from $1,000 to $250,000 due to very different sizes, mold designs, mold types, part complexities, number of mold cavities, etc. Given the great variety of molds, the data of the parts and molds must be categorized based on mold types, mold designs, materials, complexities, tolerances, and other factors that directly determine the mold cost. The key question is how to group and retrieve the information in an automated or semi-automated way so that it helps the person preparing the bid to estimate the cost quickly and accurately.

The third problem is how to improve the robustness of bidding given the variation in costs of similar molds. Since the data is categorized into similar jobs prior to estimation, the estimate may be based upon a very small sample. This small sample could have high variation. Some job types such as very special molds have inherently higher variations in actual build time, and therefore cost, than those regularly built parts and molds. This variation translates into risk or uncertainty of the cost estimate. Most of the existing mold estimation techniques use only point estimates. They are not accompanied by confidence and prediction intervals to evaluate the robustness of the estimate.

In this research, we will try to tackle the issues above and develop enabling methodology to facilitate automated cost estimation for the molding industry.

Section 1.2 Importance of the problem

Here we show the importance of this work by two complementary propositions. The first proposition is that the injection molding industry is large and important to the U.S. manufacturing sector. The second proposition is that bidding is crucial to the success of this industry.

1.2.1 Size of the injection molding industry

The industry is large and diverse. The 2005 census of manufacturing defines North American Industry Classification System (NAICS) 3261 as Plastics Product Manufacturing. This NAICS code accounted for 163 billion dollars in 2005. Of this amount NAICS code 326199 All Other Plastics Product Manufacturing accounted for 78 billion dollars [4]. This industry is thought to use the most injection molds. In 2002 NAICS code 326199 employed 488 thousand workers [5].

The molds or dies for this industry have significant revenue themselves. The census of manufacturers NAICS code 333511 is defined as establishments primarily engaged in manufacturing industrial molds for casting metals or forming other materials such as plastics, glass, or rubber. This industry accounted for 5.4 billion dollars in 2005. This does not account for those who are primarily engaged in other industries but may produce molds in order to make their primary product.

1.2.2 Importance of an accurate bid

The cost estimation of a new mold requires molders to consider the historical data of previous jobs. It is estimated that 70% of the total cost of a product during its lifecycle is fixed at design time [6]. In the injection molding domain the mold and related design account for 50% of the total cost of manufacturing the plastic part over its lifetime [7]. Therefore, getting a good idea of the mold costs at the early design phase facilitates design choices that result in lowest total cost.

The importance of getting the bid right cannot be overstated as many of these companies operate as job shops competitively bidding for the job. A bid that is too high could result in not receiving the order and a bid too low would put the potential profits of the company at risk. The risk is compounded by the fact that molders consistently have low after tax margins historically and sometimes as low as 2.1% [8]. Worldwide 60% of the mold shops are classified as small with the average shop in the United States producing between 25-50 jobs per year [3]. Therefore one missed bid could put profits at risk, as these companies are neither large enough nor profitable enough to absorb missed bids.

Because of the risk associated with a missed bid we would like to develop a systematic way to use the past knowledge and designs to estimate mold costs.

Section 1.3 Research Objective

The objective of this research is to develop a methodology for cost estimation of injection molds by automatic or semiautomatic means that can be used for bidding a wide variety of molds with the consideration of mold type, mold design, part complexity and bid risk.

Section 1.4 Introduction to Methodology

Our overall process is to reduce the dataset for all molds and parts to a smaller dataset of only those most relevant molds. A flowchart is given for these processes (see Figure 1). The functional modules of the proposed automated cost estimation system are shown in Figure 2, which may be developed from our methodology for a true industrial implementation. However, the scope of this research is developing the methodology to implement the system not a software program itself.

First we convert all data to a common neutral format thereby making maximum use of historical data. We convert all part data to a 2D image with multiple orthographic views of the part prior to further processing. A typical 2D image is shown in Figure 3. This is a data-prepossessing step that is done offline depending on the format of the data received from the customer.

Second we select mold type depending on the needs of the customer. This is shown in the first yellow block in the process overview flowchart (See Figure 1). We consider cost, material, part geometry, estimated annual volumes and the estimated lifecycle of the part. The mold type for the part being bid is selected as Conventional, MUD, or Modular. The database of prior jobs is queried to match the mold type selected for the new job. The user does this selection process from the 2D image of the part. Although we use mold type as part of our grouping criteria our methodology does not select the mold type for the user.

Next the mold design is selected based upon the geometry of the part whether Straight, Ejector, or CAM Action. This is shown in the second yellow box in the process overview flowchart see (Figure 1). The design of the mold for the new part is selected and the database of previous jobs is queried to match the design selected for the new bid. Mold design is used as part of our grouping criteria however our methodology does not select the mold design for the user.

Third, the part complexity of the new job being bid is estimated and matched against those with similar complexity in the part database. This is shown in the series of blue boxes in the process overview flowchart see (Figure 1). This is a several step process. A typical 2D image of a part is shown to help visualize the process see (Figure 3). Notice that in Figure 3 one feature namely the blue through hole has been filled in. The outside of the blue filled in area in each view is known as a boundary. A formal definition of a boundary can be found in Gonzalez [9]. This will help to explain the shape signature identification process as follows.

(1) Each enclosed boundary of the part on the 2D image is found for each orthographic view. An example of a boundary would be a profile or outline for the blue through hole in one view of the part see (Figure 3). (2) The boundary type is recorded whether the boundary is on the outside of the part or an internal feature. In Figure 3 the

blue through hole would be considered one feature on the part. (3) Each boundary in each view is described using a wavelet descriptor and other descriptors. The descriptors for one feature are combined from the descriptors from each view of the part. Each feature will have an wavelet descriptor and other descriptors such as size, perimeter length, etc for each view of the part. (4) Once each feature is described, the overall part complexity can be matched to previous parts of similar complexity. A vector describing the part complexity will be developed to include the number of boundaries, the boundary type, and wavelet descriptors, along with other descriptors.

At the end of this process we have decomposed the database of all molds and parts into only those relevant molds for cost estimation. Once we have a small sample of relevant molds for cost estimation, the next stage is to estimate the variation in build time and cost of those molds.

1.4.1 Example 1

Here we provide an example of how the methodology will work see (Figure 4). The overall process is to reduce the dataset to relevant molds for cost estimation. (1) The first step is to select mold type. In this example Conventional Mold was selected. Only Conventional Molds are in the reduced dataset before the next step. (2) In the second step the mold design was selected as Straight from the reduced dataset for Conventional Molds. Therefore the third step only considers mold type Conventional and design type Straight. (3) In the third step a clustering algorithm is used on the part descriptors to match parts with similar complexity. In this example the part being bid was a washer.

This washer had a similar part complexity and therefore mold cost to previously built molds for washers of mold type Conventional and mold design Straight. (4) To give more information about costs, we calculate mean, standard deviation and other statistics on only the washer molds. This provides a way to evaluate the robustness of the estimates.

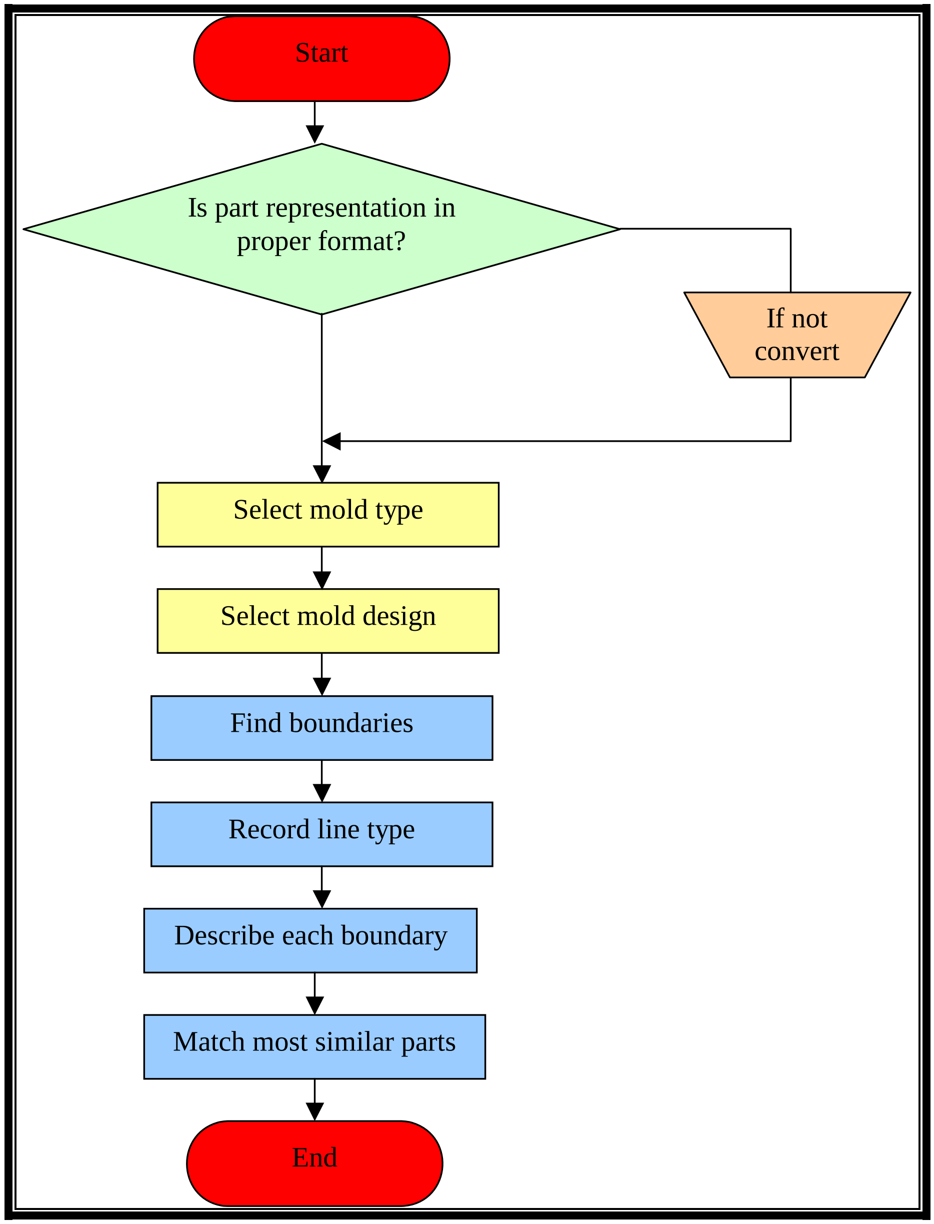

Figure 1 Process Overview Flowchart

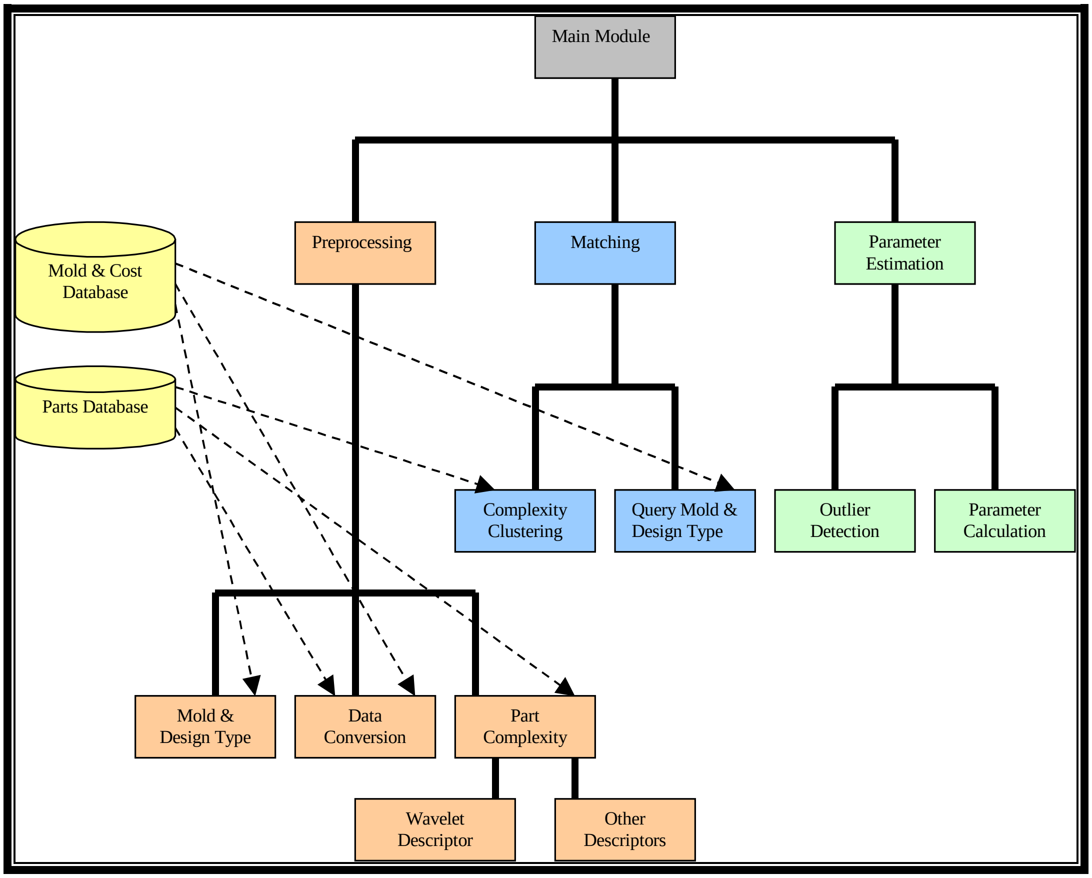

Figure 2 Functional Modules needed for Industrial Implementation of Methodology

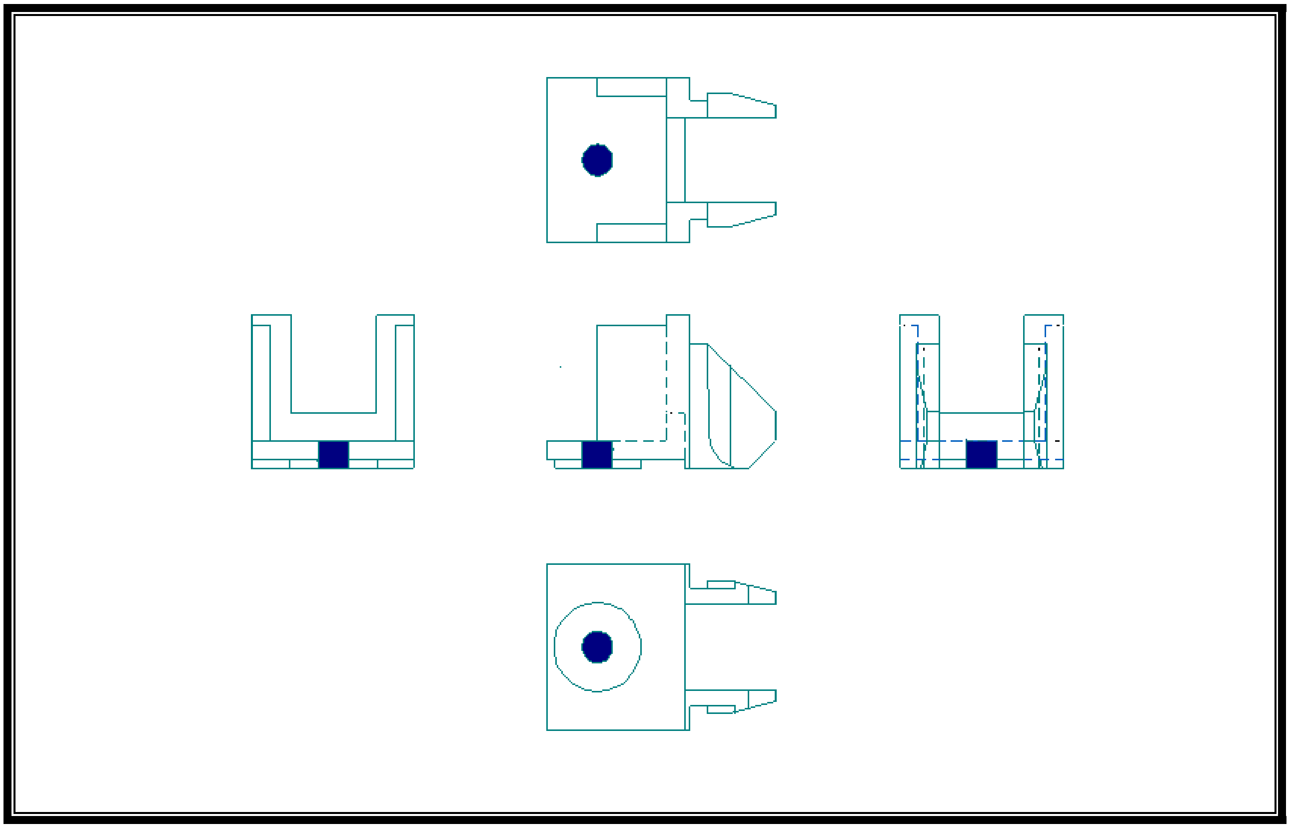

Figure 3 2D orthographic views; the blue filled in area is one feature of the part

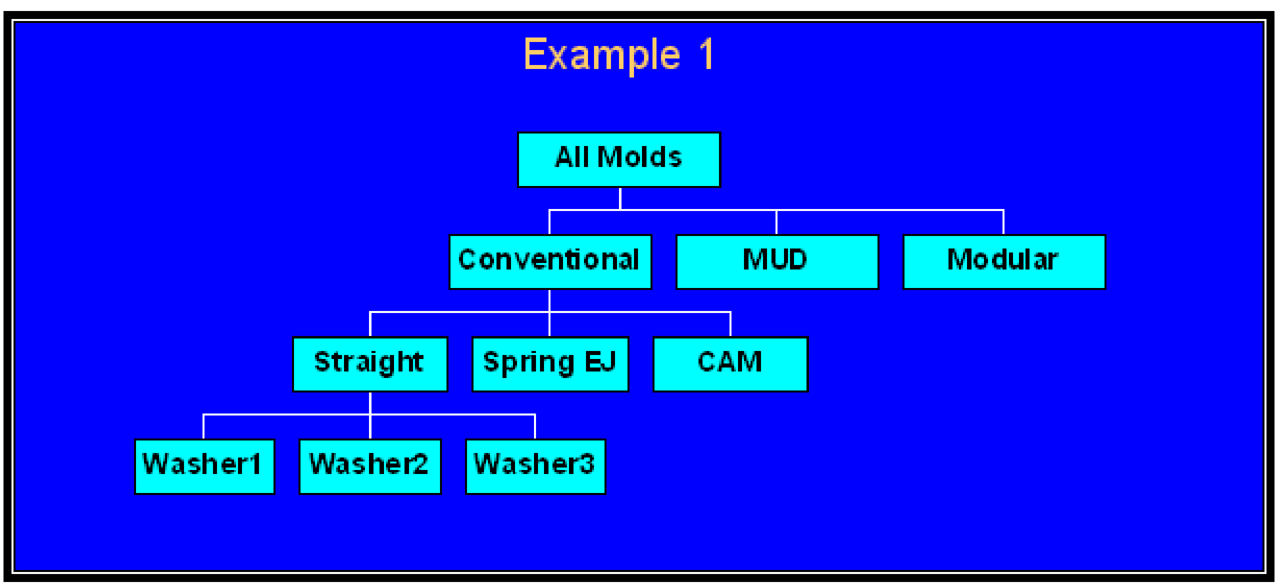

Figure 4 Example 1 of proposed Methodology

Section 1.5 How this work is unique

Previous researchers have tried to address the problem of injection mold cost estimation using both mathematical and analogy-based methods. Mathematical approaches are based upon estimating costs from parametric models. Analogy based methods make the assumption that similar jobs yield similar costs and therefore group the data. Our methodology combines the ability of some mathematical methods to estimate variation along with the ability to group molds similar to analogy based costing. We believe our method has some distinct advantages over previous efforts for several reasons.

First, the previous methods only work on one data format and cannot make use of all the historical data available. This is true for all mathematical and analogy based systems we encountered in our research. The most commonly used part and mold representations are blueprints, 2D CAD, and 3D CAD. The uniqueness of our method is that we convert all data to a neutral 2D image format. Paper blueprints can be converted to a 2D image by scanning documents into a bitmap or other compatible format. 2D CAD programs either directly export to bitmap or can be converted by a simple screen capture to a 2D image of the part. Almost all 3D CAD programs have a drafting mode. In this mode, the representation is similar to 2D CAD and could be converted either by export or by direct screen capture.

Second, other researchers only partially separate the mold and part data into homogeneous groups of mold types, designs, and part complexities before estimation. This could lead to estimates that are either biased or have a high variance. Most mathematical approaches do not separate the data at all. Previous analogy based methods group by mold type or design. However they relate the complexity of the part to the cost of the injection mold only in a rudimentary and predominately manual way. The previous methods can only distinguish simple shapes such as circles, triangles, or other primitives. The importance of this is that non-standard shapes for part features are frequently encountered in practice. Our methodology describes part features exactly using wavelet descriptors and can describe any shape.

Also unique to our work is the use internal features. Most previous methods only focus on the outline or profile of the overall part and ignore similarities not on the profile. By recording the line type as internal or external we can describe the part in greater detail. This ability to represent each feature on the part provides a finer degree of granularity for shape matching.

Third, most of the previous methods do not evaluate the variations of mold costs on a subset of only relevant molds. Therefore they cannot assess the risk or uncertainty of the bid. In most previous methods that do consider the variance, it is calculated based on the population of all molds and therefore not specific to relevant jobs. Since we group by mold type, mold design, and complexity the variance can be reduced and a more robust and accurate estimator for cost can be obtained.

Section 1.6 Assumptions

The first assumption is that a subject matter expert in mold cost estimation is available for bid preparation. We do not consider some influential non-geometric factors such as tolerances, geometric dimensioning and tolerancing, surfaces finishes, etc., which could be relevant to cost in an automated way. These factors are part of our methodology only in mold type and mold design selection indirectly. The subject matter expert would have to consider these factors and include them in the cost of the bid.

The second assumption is that the 2D images have been preprocessed to eliminate lines not related to the geometry of the part. For example, we manually delete dimension lines and ignore partial and sectional views. We also ignore isometric views or those that are not orthographic to each other. The title block of the drawing may also be removed.

The third assumption is that the database of previous molds has costing information. Each part in the part database should be matched to exactly one mold in the mold database. Therefore there is a direct link from the part to the mold and to the cost of the mold.

CHAPTER TWO: LITERATURE REVIEW

In this chapter, background and relevant work will be reviewed. Each section outlined will provide the background needed to understand the next section and the methodology chosen.

In Section 2.1, injection molding and associated tooling types will be briefly introduced. In the injection molding industry there are three primary mold types, specifically Conventional, Master Unit Die (MUD) and Modular. Each of these mold types has their own cost structure. It would not be appropriate to combine them in most cases even for similar parts or designs. For this monograph design types were limited to Straight, Spring Loaded Ejector, and CAM Action. Each of the designs types also has their own cost structure. Therefore it would not be appropriate to lump them into one group. Mold type refers to the system used to tool the part whereas design type refers to the design or construction of the mold. Many types and designs and even combinations of these are possible. However our approach is to use the most common types and designs encountered in industry.

In Section 2.2, we give a review of systematic methods used to estimate the costs of injection molds. Two basic methods are possible those being parametric or mathematical approaches and analogy or nonparametric methods. Each of the methods has their advantages and disadvantages. However, they both use history as a guide to estimate the cost of the new part under consideration.

In Section 2.3 we compare mathematical and analogical methods based upon the research. This comparison of the two approaches provides a basis for the methodology chosen.

In Section 2.4 Part Similarity is reviewed, it is important to restate that for this dissertation part similarity, geometry, and complexity are synonymous and considered to be related to cost. The tacit assumption made is that more complex geometries or shapes would take more hours to produce and would result in a higher mold cost. Section 2.5 is a review of wavelets and wavelet descriptors.

Section 2.1 Injection Molding General Discussion

In this section we review some background that will help the reader to understand the methodology chosen. In Section 2.1.1 we review the injection molding process. In Section 2.1.2 we show three mold types. Specifically we show the Conventional, MUD, and Modular mold types. These are three common systems used in the injection molding industry and each one has a unique cost structure. In Section 2.1.3 three mold designs namely Straight, Spring Ejector, and Cam Action are examined. Each mold design is fundamentally different from the others and carries different costs.

2.1.1 Injection Molding Process

Injection molding is a process where solid plastic pellets approximately the size of a grain of rice are heated until they are liquefied and injected into the mold under very high pressures on the order of 5000-15000 pounds per square inch see Figure 5. After the plastic is injected into the mold the plastic cools to a solid state and is ejected from the mold in preparation for the next cycle.

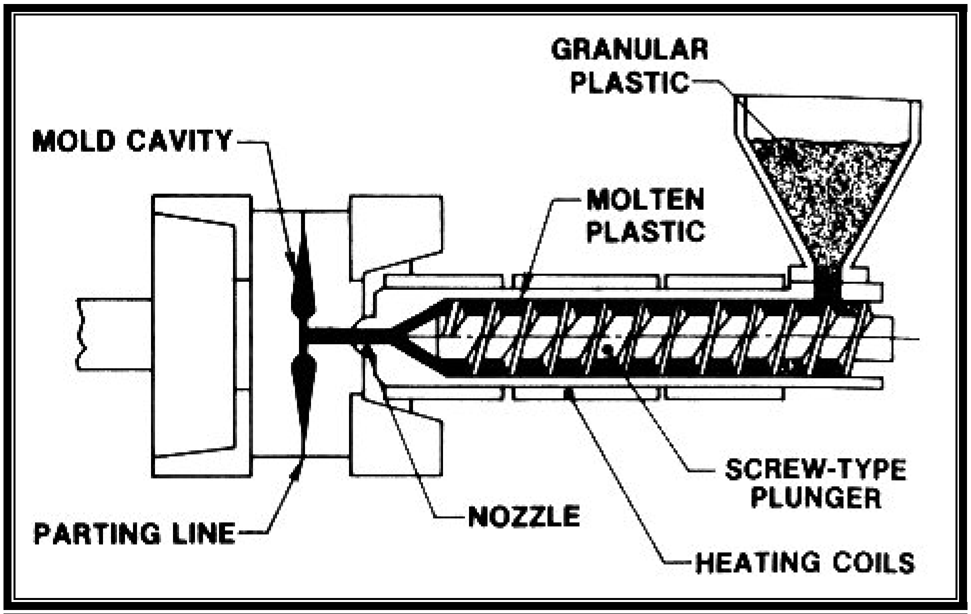

Figure 5 Injection Molding Process

2.1.2 Injection Mold Types

2.1.2.1 Conventional Mold

In a conventional mold see Figure 6 one mold is dedicated to one part. There may be many cavities of the part however each cavity is identical for all practical purposes. The advantage of conventional molds is that the molder can optimize the process for that specific part. The process does have some disadvantages however. The first disadvantage is that the molder may need to tool many cavities to make the run economical. Another disadvantage is that this process requires its own mold base and associated mold base preparation, which could add several thousand dollars to the mold.

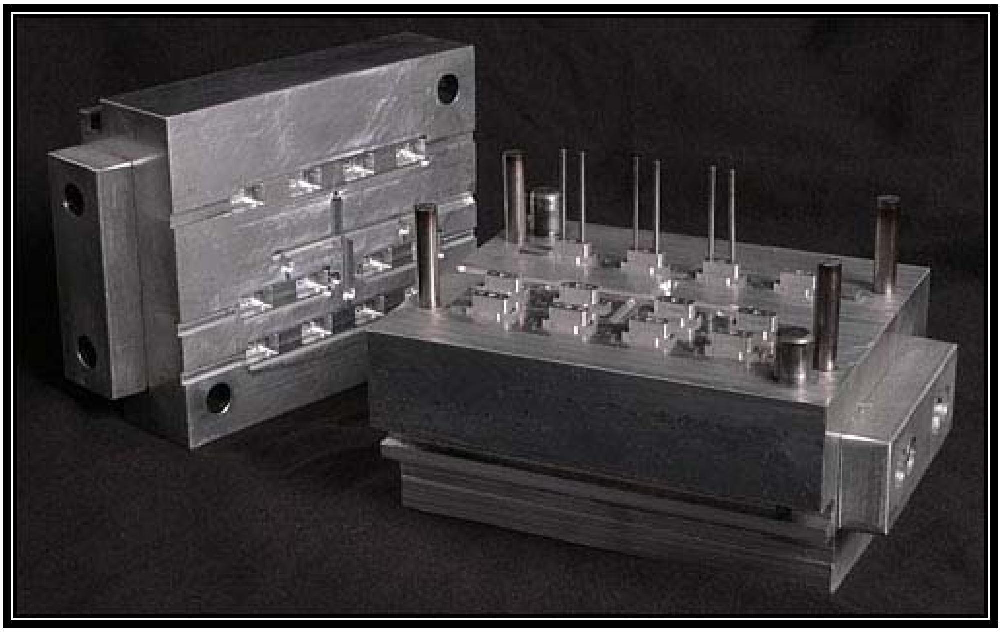

Figure 6 Conventional mold used in industry

2.1.2.2 Master Unit Die

In a MUD or Master Unit Die® the savings are associated with the fact that the mold maker is only required to buy the mold base once. The molder reuses this base and only buys the insert set that will be needed to produce the parts. Additionally the mold setup cost may be reduced due to the mold base already being setup within the press. The operator then slides the insert into a pocket in the mold base of the MUD see Figure 7.

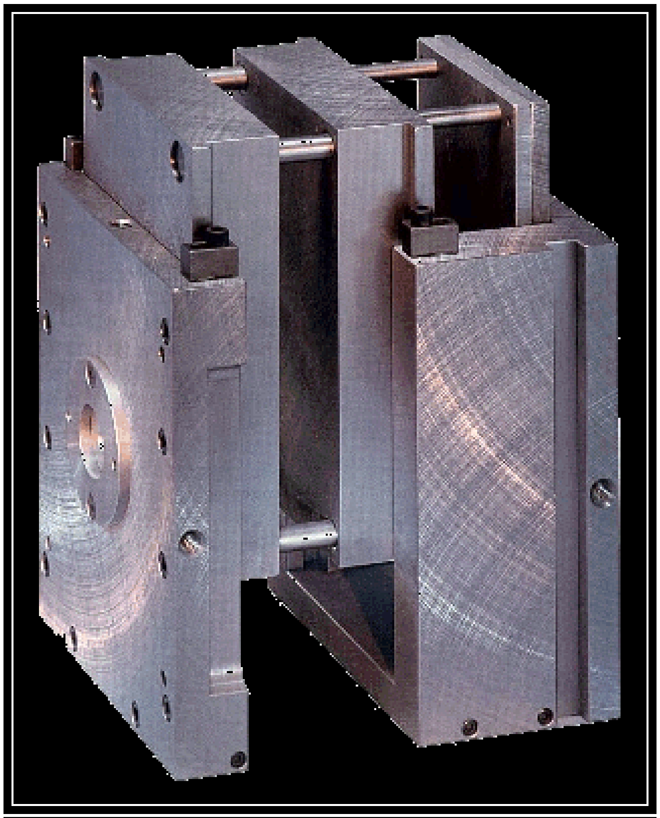

Figure 7 Master Unit Die (MUD) mold base and insert

2.1.2.3 Modular Mold

Modular molding has several of the advantages of other types of injection molding but does have some limitations. In modular molding each separate part has its own insert in the master mold base. Those inserts are interchangeable and on any given day the makeup of those inserts in the mold base change based upon the orders received by the molder. Each part shown in Figure 8 has its own insert and fits into the master mold base.

The advantages of this system are manifold. The first advantage is that the cost of the tool is potentially lower because the number of cavities to make an efficient operation of the press is shared between the inserts. Each insert shares the cost of the press. For example, if the total cost of the press is $50.00 per hour and we have 10 inserts, each insert would only be required to pay $5.00 per hour for its share of the total cost of operation. The benefit is that the molder tooled one-tenth the number of cavities that normally would have been required for the same part cost. This greatly reduces the cost of the mold. Another advantage is that because the press is typically dedicated to a specific material and the mold base is already installed in the machine, therefore the setup cost is reduced.

However, modular molding does have some disadvantages. The first disadvantage is that the customer must be able to accept the part in those materials that the molder uses for modular molding. The second disadvantage is that the parts are small and would fit in the palm of your hand. The third disadvantage is the press operation cannot be optimized to suit a particular part. This is due to the fact that many different inserts share the press.

Figure 8 Modular molding trees with many parts

2.1.3 Mold Design Types

A complete mold design description is beyond the scope of this document. The interested reader is directed to the many good mold design references available such as the one by Kluz [10].

There are many mold designs not considered for categorization and therefore not a part of this dissertation. We only consider the mold designs Straight, Spring Ejector, and Cam Action because these are the most common designs. The salient point is that these three mold designs have a different cost structure from each other. Therefore it makes sense to group the mold designs in this way before cost estimation.

2.1.3.1 Straight Molding

A straight molding is the most common, simplest, and least expensive injection mold to build. Shown in Figure 6 is a typical straight ejector mold used. The ejector system in an injection mold removes the part after molding. In straight ejector mold designs the injection-molding machine has a mechanism that pushes the part ejector system in the mold directly.

2.1.3.2 Spring Loaded Ejector

In a spring-loaded ejector mold, the mold itself must eject the part. The molding machine does not assist this process. The ejector system in the mold is spring-loaded and the force of the springs against the ejector system ejects the part. This additional work to the mold has a cost and must be included in the bid.

2.1.3.3 CAM Action

Cam action molds are designed so that internal features of the part may come from more than one direction. A typical cam action mold is shown in Figure 9.

Notice that in Figure 10 some internal features on the part are at right angles to one another. The metal in the mold that forms these internal features must be removed before ejection of the part out of the mold. The molding machine provides only one direction of movement. Therefore additional directions of movement must be provided by the mold itself. Cam action molds require extensive labor and this labor cost must be included in the cost of the mold.

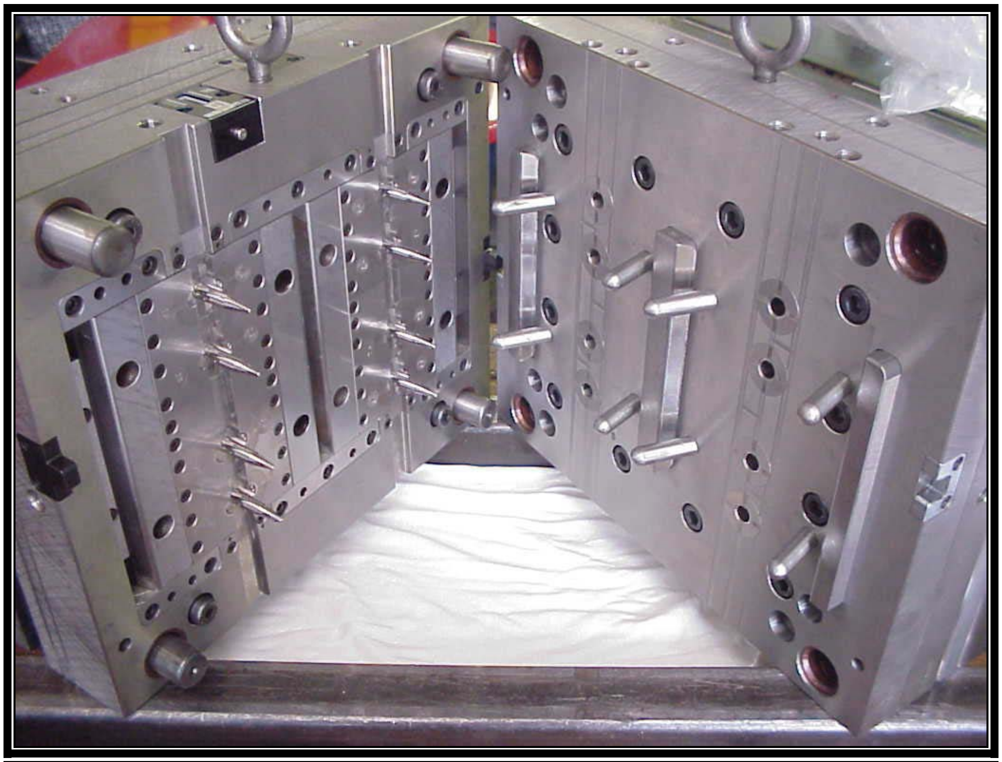

Figure 9 Cam action mold used in industry.

Figure 10 Injection molded part requiring cam action

Section 2.2 Methods to Estimate Cost of Injection Molds

As mentioned in the introduction the most common method used to estimate injection molds is ad hoc. The assumption of the ad hoc costing procedure is that an experienced bidder prepares the bid. The problem with this method is that it is dependant on the experience of the person doing the bid. Therefore methods have been developed to systematically use the information of previous jobs and several methods have been developed.

There have been five groups of methods to estimate the cost of the injection molds as outlined by Nagahanumaiah [7]. These methods can be classified as intuitive, analogical, analytical, geometric feature based, and parametric.

We group these five methods into two broad categories. In the first category the past experience of molds is used to estimate costs, these would include the intuitive and analogical. In the second category, the mold cost and the drivers of the costs are calculated mathematically. The analytical, geometric feature based, and parametric methods fall into this category.

2.2.1 Analogy Based Methods

The analogical method is to group similar jobs together in categories based upon the features of the part and mold. Several variables can be used including the size of the mold, material of the mold, complexity, ejector and gating method.

One method based on Group Technology was proposed by Poli [11]. The group technology approach works the best when applied to families or groups of parts. An injection molder that is a pure job shop may have over one hundred customers in various industries so it is difficult to know if it would be possible to create families of parts in that setting.

A combined blackboard and case based reasoning approach was taken by Kwong and Smith [12]. The contribution of the paper is that cost estimation is done at early design and concept generation stage. The blackboard design allows interaction with other systems and facilitates component design. Injection molded components are generally used in assemblies. Those assemblies may include springs, wire forms, stampings, castings, etc. This effort only selects the mold base and molding machine parameters based upon 90 box shaped parts. It does not calculate the mold cost itself and does not support other shapes such as round or cylindrical parts.

El-Mehalawi [13] presented another effort with a combination of tool information such as the type of mold and part information using geometric similarity. The type of mold is characterized by the part complexity, for example, whether or not the part has a straight or offset parting line, or whether the part requires cam action. This method does attempt to use both mold and part information to group the parts. However the method has some disadvantages. First, constructive solid geometry (CSG) is used in the comparison. Therefore the method of construction could affect the geometric similarity. Furthermore it uses 3D CAD models, which may be available in less than 30% of cases [3]. The model was not validated and was left for future work. Therefore its applicability is not known.

One commercial application was originally designed by Institut fuer Kunststoffverarbeitung (IKV). This system was expanded by Simcon Gmbh [14]. The current version is called Moldcalc. It groups similar jobs together with the idea that similar jobs will produce similar costs.

Wang et al. [15] used a case based reasoning system to evaluate the cost of the mold with partially known feature vectors consisting of geometry, features, tolerances and surface finishes. Various methods are used depending on the similarity of the retrieved cases to the current part under consideration. If they are similar enough (Over 60% of similarity as judged by a human expert), they use a current case. If not judged similar enough they use the Dixon Poli Method or Boothroyd Dewhurst method.

Another Case Based Reasoning attempt was a partnership by Legrand and Kaidara software. Legrand's Center for Studies and Research in Plastics Science based in Limoges, France and Kaidara developed the Rapid Cost Estimation for Plastic Parts Production (ERCP). This system has over 600 cases with 40 pieces of data including photos and CAD drawings. This enabled Legrand to estimate the cost of molds in 3 days as compared to 3 weeks in the previous method and reduced the cost to estimate by 30% [16].

2.2.2 Mathematical Based Approaches

Mathematical Based Approaches include buildup and parametric approaches. The buildup approach estimates the cost of sub tasks and adds them together or the final bid. The parametric approach estimates the cost by use of factors thought to relate to mold cost. These factors could be size, number of dimensions, or other factors that may have a bearing on cost.

One approach was developed by Raviwongse and Allada [17] which uses a back propagation neural network with 14 factors to calculate the complexity of parts . The authors acknowledged that qualitative factors may be needed and the complexity may not be linear.

Dennis Pearce at the IBM Plastic Development Center developed a program called MOLDCOST or (More Or Less Determining Costs Of Selected Tooling), this method used a stepwise regression technique based on the following six parameters. Number of dimensions, number of surface finishes, length of part, depth of part, tightest tolerance, and the number of cavities [18]. This same study mentions that the software is inaccurate on small parts where the linear relationship breaks down between these variables and the cost function.

An evaluation of Boothroyd Dewhurst Incorporated (BDI) Method are explored by Wong [14]. Wong used the BDI method and compared that to quoted prices for the parts. In some cases the method performed well, but in other cases it performed very poorly versus the quoted price.

Shehab and Abdalla [6] focused on estimating the cost of a machined component and an injection-molding component. Both the injection mold and the plastic part itself are estimated. Predominately considered are part size and the number of cavities. Part complexity is not considered. The method does have an interesting feature that tried to consider the fact that if a company were tooling many cavities the cost per cavity would be reduced. In another effort by the same authors, a method to get a handle on costs at an early design phase was implemented [19]. A fuzzy logic and expert systems approach was used. One problem with this method is that it was validated with only two molds.

A computerized price quoting system was developed by Chan et al [20]. In this work the design of the tool is considered. Standard costs referred to as primitives are assigned to activities such as milling, grinding, and EDM. Also considered are the components that make up the mold such as the mold base, ejector pins, and bushings.

The method employed in Nagahanumaiah et al [7] is to use the Quality Function Deployment Method to estimate the cost of the mold. This is a breakdown of the costs considering features such as cam action tools, and the machining method such as milling, turning, and EDM. The model does take into consideration the surface finish of the cavity of the mold. One potential problem with this method is that the model was validated with only 13 sample cases.

Sapene [21] used a detailed breakdown to estimate the cost of the mold. The author used the following general categories for the final bid. Total cost of standard components, total cost of parts manufactured by the mold maker, processing fees, tolerance of deviation in the cost estimation (10% of total cost), overhead and unexpected costs (10%), and profit (15%).

2.2.3 Overview of injection mold cost methods

An overview of the methods used to estimate the cost of injection molds is given in Table 1. Our research builds upon many of the ideas of previous efforts. However only one effort had automatic part matching like our effort. The previous effort most like our work is El-Mehalawi [13]. Therefore we would like to point out the differences between our approach and El-Mehalawi. The differences are shown in Table 2.

Table 1 Comparison of cost estimation research projects

Author	Type	Publication	Automated part matching	Validated	Implemented
11. Kwong, C.K. and G.F. Smith,	Analogy	Journal	No	No	Yes
12. El-Mehalawi, M.E.-S.,	Analogy	Dissertation	Yes	No	Yes
13. Wong, T.S.,	Analogy	Thesis	No	No	Yes
14. Wang, H., X.Y. Ruan, and X.H. Zhou,	Analogy	Journal	No	No	Yes
15. Bergmann, R.,	Analogy	Book Chapter	No	Yes	Yes
6. Shehab, E. and H. Abdalla,	Mathematical	Journal	No	No	Yes
7. Nagahanumaiah, N.P. Mukherjee, and B. Ravi,	Mathematical	Journal	No	Yes	Yes
16. Raviwongse, R. and V. Allada,	Mathematical	Journal	No	No	Yes
17. Merino, D.W.,	Mathematical	Dissertation	No	Yes	Yes
18. Shehab, E.M. and H.S. Abdalla,	Mathematical	Proceeding	No	No	Yes
19. Chan, S.F., C.K. Law, and K.K. Chan,	Mathematical	Journal	No	No	Yes
20. Sapene, C.,	Mathematical	Book	No	No	No

Table 2 Comparison El-Mehalawi vs. Hillsman

	El-Mehalawi	Hillsman
Year	1999	2009
Data Format	Constructive Solid Geometry	Images
Descriptors	Attributed Graph	Wavelet, Regional, Topological
Matching	Largest Common Subgraph	Hierarchical Clustering
Approach	Analogy	Analogy/Mathematical
Hierarchical	No	Yes
Validated	No	Yes
Pros	Detailed Mold Description	Detailed Part Description
Cons	Slow Matching, Not Validated	Small Sample, Preprocessing

Section 2.3 Comparison of Mathematical vs. Analogy Based Methods

Between the mathematical and the analogical approaches, we may wonder how to evaluate which one is better. This question has not previously been explored in injection molding domain but has been researched in other areas such as mechanical design and software engineering. Here we examine these comparisons to shed light on what may be most appropriate for our problem.

2.3.1 Mathematical vs. Analogy Based Costing Case Studies

Thibault et al. [22] provided a framework for using analogy based costing for mechanical components. They concluded similar to the work by H'Mida et al. [23] that a purely mathematical approach to the cost estimation of mechanical components would not be completely successful. In other words we must group similar components together in some way to get an accurate cost model.

Duverlie and Castelain [24] focused on comparing the mathematical to the analogy based costing approach for pistons. They show that that the analogy-based method was superior to the mathematical approach because of simplicity of use and the fact that it has a lower error. They do concede that it may be more difficult to implement.

A study by Sheppard showed that analogy based costing worked better in software cost estimation as well [25]. Specifically he compared analogy, linear regression, and stepwise regression. The measure for success was the mean magnitude of relative error (MMRE). Another related effort in software cost estimation is done by Sheppard et al [26]. Shepperd and Schofield used MMRE and the percentage of estimates that are within 25% of the actual cost to compare analogy vs. mathematical methods. They used 9 total datasets and concluded that the analogy based method performed better in 7 of the 9 datasets. They also list several reasons why the analogical approach is superior. First, coding the data can be simpler, second only things that actually happened are considered in the estimation, third the knowledge of failed cases is included, fourth is the ability to deal with poorly understood domains, and fifth acceptance by users.

Based upon the above studies, it can be seen that heterogeneous datasets need to be grouped into more homogeneous datasets before estimation, which is the approach taken in this dissertation.

Section 2.4 Part Similarity General Discussion

As explained in the 2.3.1 Mathematical vs. Analogy Based Costing Case Studies analogy based costing presents some advantages over the mathematical approaches. Therefore it makes sense to group the jobs together in some way. The most natural way is to group them by mold type Conventional, MUD, or Modular. Once that is done they can be grouped based upon the design of the mold whether Straight, Spring Ejector, or CAM Action.

However we still have not grouped them according to complexity and that is the purpose of this section. Suppose our total dataset has 5,000 molds and of those 3,000 are found to be mold type Modular. Suppose we sub query those 3,000 for only design type CAM action and are left with 1,000 molds. While this dataset may be unbiased it may also have a large variance. A preferred dataset should be sparse and reflects only those most similar molds for estimation. In our case we want those types and designs that also exhibit the same complexity as the job we are preparing a bid for. The best way to do that is to use the geometry as a measure of complexity and ultimately cost. Therefore in this section we review how similar geometries are grouped together and represented.

A core question when considering complexity is the representation of the part or geometry itself. The reason for this is that geometry similarity methods are inherently either 2D or 3D. Therefore the representation chosen is a key question to any implementation. In this chapter we examine 2D and 3D representation and similarity methods as well as the advantages and disadvantages of each method.

One observation is that 2D is fundamentally different from 3D. In 2D methods there is only one space. Therefore both the geometry and non-geometric information is contained in the same drawing. Non-geometric data could be dimensions, tolerances, geometric dimensioning and tolerancing, or notations. Typically in 3D methods the geometry is defined in the model space and non-geometric information is in the drafting space.

2.4.1 Surveys of geometry similarity methods

There are several active research groups working in the area of geometry similarity such as the Temple Shape Similarity Project which can be found at [27], the Princeton Shape Retrieval and Analysis Group at [28], and the Purdue PRECISE lab at [29]. Therefore a comprehensive coverage of the topic will not be given here. However, a general overview of the some main contributions and ideas are presented. The field can be broken down into two categories 2D shape similarity and 3D shape similarity. 2D shape similarity is well studied and represented by the literature. Some good reviews are found in Belongie et al. [30] and Loncaric [31]. 3D similarity has grown significantly in the literature in recent years. Some good 3D references are the survey papers for 3D shape similarity given in Iyer et al. [32], Tangelder et al. [33], Cardone et al. [34] and Bustos et al [35].

2.4.2 Common to all Methods

In this section we review some basic properties and issues related to all shape similarity. This will provide some background used in later sections related to a specific topic.

2.4.2.1 Topology

Topology alone cannot distinguish between dissimilar parts [36] Although related to geometry, topology is the study of how shapes are connected not the geometric shape of the item [37]. See Henle [38] for a good introduction to topology. Two similar topologies but different geometries such as a donut and coffee cup would be vastly different from a mold design and cost perspective. However elements of topology may be useful for our purposes. Some useful elements may be the Euler characteristic, Betti numbers, and Genus as they could help to distinguish between two images of the parts.

2.4.2.2 Metrics

Metrics play a role to measure similarities. Metric spaces are defined by the following four properties: Identity, Positivity, Symmetry, and the Triangle Inequality. These are commonly represented by the following equations.

(Identity) $$\forall x \in S, d(x,x) = 0 \qquad \textbf{(1)}$$

(Positivity) $\forall x \neq y \in S, d(x,y) > 0$ **(2)**

(Symmetry) $\forall xy \in S, d(x,y) = d(y,x)$ **(3)**

(Triangle Inequality) $\forall xyz \in S, d(x,z) \leq d(x,y) + d(y,z)$ **(4)**

Identity in Eq. (1) states that the same shape is identical to itself and the distance between two identical shapes is zero. Positivity expressed in Eq. (2) ensures that the distance metric should always be positive. To match two items we are comparing dissimilarity or distance between two parts. Therefore absolute values are used for comparison not direction of similarity. As expressed in Eq. (3), Symmetry states that the order of comparison should not affect the metric used to compare two shapes. The Triangle Inequality in Eq. (4) means that if *A* is close to *B*, and *B* is close to *C*, then A is close to C [39]. The Triangle inequality is important to indexing schemes making search more efficient. A good overview is given in Vleugels and Veltkamp [40].

Tangelder et al. [33] pointed out that when a measurement has all four properties it is referred to as a Metric. If it obeys all of the four except Positivity this is called a semi-metric. When all four properties except the Triangle Inequality are present then it is a pseudo-metric.

Some other properties that proposed metrics should have are Invariance, Robustness, and Computational Efficiency. Invariance ensures that rotation or translation should not effect the metric, Robustness means that the signature should change in proportion to the change in shape, and Computational Efficiency makes sure that the algorithm should be fast enough for its use according to Cardone et al. [34].

2.4.2.3 Transformations

Transformations are used to modify the image in several ways. Transformations control rotation, location, reflection, shear and scale relative to the coordinate space. These methods could affect a geometry similarity method depending on whether it uses a metric, semi-metric, or pseudo-metric so they are mentioned here for completeness. A good discussion about the classes of transformations is given in Dunn [41].

2.4.3 Two-Dimensional shape similarity

In this section we review 2D shape representation and shape similarity as background information for the following sections. In section 2.4.3.1 we review two-dimensional shape representation, and in section 2.4.3.2 we review how 2D images are converted to other 2D forms to make similarity assessment easier in some way.

2.4.3.1 Two-Dimensional shape representation

Two-dimensional shapes can be represented in many ways. Historically blueprints were used and produced on drafting tables. Today the most common method would be a CAD program of some type however blueprints are still used.

2.4.3.2 Conversion of images and similarity assessment

There are several formats that can be used for images. Here we will review some of the more common methods to facilitate the topics to follow.

Freeman chain codes are used to represent directions while tracing the boundary of an object in the image [42]. There are two types in common usage. In a 4-directional chain code see (Figure 11) the numbers zero though three are assigned to each 90-degree direction in a counter clockwise direction with zero representing east, one representing north, two representing west, and three representing south. In 8-directional chain codes a similar scheme is used at each 45-degree increment counterclockwise starting with zero being east. Using a 4-directional chain code if we trace the boundary of a square in a clockwise direction and starting at the lower left corner of the boundary the 4-directional chain code would be (0,1,2,3). After an image is converted to a Freeman Chain Code regular expressions are used to compare two geometries.

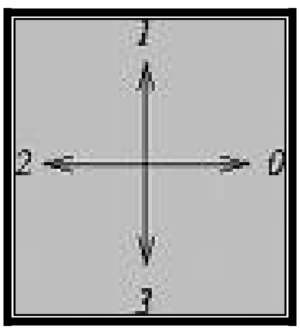

Figure 11 Four-Direction Freeman Chain Code

Another method uses minimum perimeter polygons (MPP). The theory behind this approach is that any shape can be approximated to any desired accuracy using polygons. The goal of this method is to enclose the boundary using minimum polygons given a desired approximation. An algorithm to do this is given in Gonzalez [9]. One attractive feature of this method is that varying the number of polygons has the effect of increasing or decreasing the generality of the shape. A low number of polygons will give a rough description and a high number a more accurate one. The sides of the polygon are often traced in one direction and represented as a chain code. This chain code is compared to other shapes by a regular expression.

Signatures convert a 2-D representation to a 1-D function. One method is to measure the distance from the centroid of object to the boundary for many different angles on the boundary. The distance measured becomes the Y coordinate and the X coordinate is the angle from zero to 360 degrees. Measuring the difference of the radii at each angle is how two geometries are compared.

Boundary segments decompose the boundary into convex and concave sections. The sections are divided where there is a transition from convex to concave or visa versa. There are several basis used to compare boundary segments including number convex sections, number of concave sections, length of the sections etc.

Skeletonization is a method where an object is thinned to its essential line. If for example we had an image of a human we would progressively thin this image until just the skeleton or stick man figure were visible and that skeleton would represent the shape of the person. After this is done the skeletons are compared to each other using several methods.

There are several techniques of simple boundary descriptions to represent the shape. Some are length, diameter, major axis, minor axis, basic rectangle, bounding boxes, and area.

Shape numbers are based upon the first difference of smallest magnitude according to Bribiesca and Guzman [43]. This method is independent of the size, orientation, and position. The order of the shape number is the number of freeman chain codes used to describe the object.

Another method is based on statistical moments; the boundary is sampled similar to signatures however in this case we normally use a cardinal axis to align one side with the x-axis. We reduce the 2-D representation to a signal and describe it with the moments treating the signal as a distribution after normalization. Hu [44] defined seven nonlinear functions on regular moments that were translation, scale, and rotation invariant.

2.4.4 Three-Dimensional Shape Similarity

In this section background will be reviewed to facilitate the topic to follow. In section 2.4.4.1 we review 3D representation, in section 2.4.4.2 we review 3D shape similarity.

2.4.4.1 Three Dimensional Representation

3D models are most commonly represented by two methods Constructive Solid Geometry (CSG) and Boundary Representation (B-Rep) [45]. These models define the geometry exactly.

A CSG represents a solid by set theoretic Boolean expression of primitives. It is a solid representation not a surface. In a CSG both the outside and the inside of the object are defined implicitly. A B-Rep model is represented as vertices, faces, and edges. The boundary is enclosed but it is hollow and only the shell. An excellent book with clear explanations is given in Adams [46]. Adams breaks down the process of fundamentals, modeling, rendering, and animation.

Mesh and voxel models only approximate the surface or true geometry as contrasted with B-Rep or CSG. Mesh is a tessellation of the surface made up of triangles and was developed for viewing over Internet. Another method is Voxelization where a shape is represented as a matrix of primitive shapes. The 3D shape is sectioned into intervals of the overall shape with each interval being either present or not present dependant on whether the 3D solid is present in that section. The most common primitive would be a cube.

2.4.4.2 Three Dimensional Shape Similarity

Moments have been extended from 2D to a 3D setting and are based upon statistical moments. These methods are based on the moments of inertia. The moments are defined by the Riemann integral. The function is assumed to be piecewise continuous. The set of moments uniquely define the 3D model. Generally the lower order moments are used, the higher moments that may be more prone to noise are discarded. If a voxel model is used, this method can be converted to a summation. Celebi et al. [47] compared invariant moments, Zernike moments, and radial Chebyshev moments on the MPEG-7 shape database and showed that radial Chebyshev moments achieved the highest retrieval performance.

Spherical harmonics characterize a shape based upon a parameterization on a sphere. The availability of a polygonal soup is assumed. For a good explanation the reader is referred to Funkhouser [48]. This method does suffer from the fact that any radii could be rotated with respect to the other radii on the sphere and the spherical harmonic would remain unchanged. From a mold design perspective this could cause serious problems if two models were returned by this method that appeared to be similar but were not.

Volumetric Error methods are based on the idea that two similarly shaped objects have little difference in volume after pose and scale normalization. Generally the comparison method uses a voxel representation. Novotni and Klein [49] used a distance measure to compare 2D distances of the voxel based upon slices first and then converts that to full 3D. Histograms are used for comparison. This method does not use a metric because it fails the symmetry property in Eq.(3). In another method by Sanchez-Cruz and Bribiesca [50] the number of voxels or volume is normalized between two shapes. The difference between shapes is the total number of voxels that had to be moved to convert between one shape and the other divided by the total number of voxels.

Manufacturing feature based methods are one of the oldest methods of similarity. The approach taken by Cicirello and Regli [51] is to extract features for the purposes of building a model dependency graph. Then the graph itself is compared to determine the similarity of two products. A good overview of feature recognition techniques is given in JungHyun [52].

Group Technology is a mature method. Burbidge [53] used a taxonomy to group similar parts together based upon codes. There are several coding schemes. The problem with this method is that the taxonomy is developed by the user and does not necessarily reflect natural groupings based upon geometry or topology.

One skeleton based method is Dilation Based Multiresolutional Skeletons (DBMS). DBMS were developed by Gao et al [54] in order to deal with weaknesses of the Medial Axis methods. A core strategy is the multiresolutional or hierarchical aspect of this technique. The method goes from rough matching to more detailed matching based upon three levels of detail. The model is complex and the data storage could be prohibitive when using this method. However good results were reported on matching based upon the F-Measure. Due to complexity, storage requirements, and the fact the method is based upon B-Rep we do not consider this method viable for our research.

Reeb graphs store slices of a shape as graphs. One effort by Biasotti et al [55] used this method to search and retrieve sub parts of a geometry that were similar. This approach allows the user to search for subgeometry within the overall geometry of the part. Another effort of this type by Bespalov et al. [56] used scale space feature extraction.

Topological Graph methods use a graph theory representation of the parts. However comparing two graphs is not a trivial matter [57]. A balance must be struck between computational complexity and the detail needed to distinguish between similar and dissimilar parts.

Model Dependency Graphs are constructed from 3D solid models. Generally B-Rep is used. The process outlined in El-Mehalawi and Miller [58] involves 3 steps. The first step is creation of a STEP file from a 3D solid CAD program such as SolidWorks or ProEngineer. From the STEP file an attributed graph is produced and the attributed graphs are compared for similarity. The process does offer some significant advantages over other methods. The advantages are the fact that topological data, geometric data, and part size may be compared using this method. The disadvantages are the graph comparison is slow and known to be NP complete, although a less precise algorithm is given in El-Mehalawi and Miller [59] that runs in polynomial time. El-Mehalawi and

Miller point out that STEP files are not unique and susceptible to the specific solid CAD software used and the order of construction. A solution to the order of construction problem was presented by Regli and Cicirello [60].

Statistics of shape is based upon landmark features, for example, the distance between two holes. This method is usually applied to similar shapes like the skulls of humans [61]. As such this method is not particularly well suited to this monograph but is mentioned for completeness.

Shape Histograms were explored by Dong et al [62]. Shape Histogram methods use sample points to represent the shape and then use histograms to record the values. For example, if we were recording points from the centroid to the edge of the part, we would record how many times these vectors were between 0.25 and 0.50 and that frequency would be used to compare the shapes.

Shape distributions are related to shape histograms and in fact the techniques are frequently used together. The principal is explained in a paper by Osada [63]. Osada et al. found that the D2 metric, which computes the distance between two random points, is preferred. In this method two random points on a shape are selected and the distance measured between them. This is done several times and the distances measured are placed into a shape histogram. See also Cardone [64]. This method however is not unique and it would be possible for two dissimilar shapes to have the same shape histogram. Therefore it is not recommended for our work.

Extended Gaussian Images is an orientation histogram of points on the surface of a three dimensional object. It is defined as the inverse of the curvature of given points on the surface of a Gaussian sphere. It can be used to recognize convex objects only as it is not unique for non-convex objects. EGI can also be used for pose and symmetry determination. Some references on this method are those by Horn [65] and another by Zouaki [66].

Local curvature can be regarded as an extension to Extended Gaussian Images and is used to compare objects that have natural shapes found primarily in nature under high curvature. This method would be used to compare a pear to an apple for example. Given the purpose it is not particularly well suited to the needs of this monograph in which shapes are represented by geometry and generally Euclidian geometry. One particular application of this method is Spherical Attribute Images. A reference to this method is given by Hebert [67]. These were developed to overcome the limitation of EGI inability to deal with non-convex objects and work by deforming a polyhedral sphere to approximate the shape of the object and storing at each point the angle or curvature at that point.

Slope Diagrams of convex shapes can be compared in a method outlined by Tuzikov [68]. The methods are invariant under translation and some are invariant to scale, rotation, and transformation. A slope diagram is parameterized where each facet is represented as a point, each edge connects the points on the unit sphere and each vertex is an intersection of edges. This method is only applicable to convex shapes and is not unique for non-convex shapes.

Weighted point based methods break a shape down into cubes. In each cube the shape is based upon a point within the cube called the salient point and a weight that represents the curvature of the salient point. The idea is that points with higher curvature are more descriptive and thus should carry more weight than straight sections. In the technique proposed by Tangelder [69] a shape is broken down into a 25 by 25 by 25 grid. For each of the cubes within the grid a salient point is assigned based upon three methods. The three methods are the Gaussian Curvature, Normal Variation, and Midpoint method. The shapes are then matched based upon a modified earthmovers distance. This method does have an advantage over conventional techniques in that this method can be used for polygonal soups on 3D models that are not watertight.

Aspect graphs define a 3D object by characteristic 2D aspect views of the object. Principally each aspect view is the same as another if it is topologically equivalent and connected by a continuous path. The question that this method tries to answer is how many views of a 3D object are needed and which ones are characteristic views of the object and which ones are redundant. This method does suffer from a couple of problems namely the storage requirements and slow computation. To offset these drawbacks some have taken the practical approach of sampling the viewing sphere at specific locations. One method of this type is introduced by Cyr and Kimia [70].

Geometric Hashing is a method where sample points from an image are taken, scaled, and rotated such that each pair of points called basis points is normalized at the origin on the x-axis. This achieves rotation and scale invariance. For each possible rotation that aligns the basis points with the x-axis all the other points' location are recorded in bins. A voting strategy is employed similar to the Hough Transform. The method is well outlined in Wolfson and Rigoutsos [71]. This approach was developed to recognize objects in a model database in the presence of occlusion. The approach has been extended to 3D models in van Dijck and van der Heijden [72].

Combined methods were recently proposed by Chu and Hsu [73]. In this paper they combine topological, form feature, and geometric methods into one method and search for the best match based upon several criteria. Chu and Hsu show the weaknesses of any single shape criteria and how the combined method is superior. They reduce the topological graph comparison by mapping into graph cliques thereby reducing the computational time greatly and reducing the problem to a linear programming problem by the Hungarian method. The implementation used CATIA ® and SmarTeam®. This method does have some limitations. The first is that the model does not consider chamfers or fillets and does not include things like the tolerances or texture of the part.

A Hybrid 2D/3D approach was proposed by Pu and Ramani [74]. They start with a 2D drawing that they convert to a 2.5D spherical harmonic and 2D shape histogram. They transform a 2D shape into a 3D shape then use a comparison between randomly chosen points to determine similarity.

2.4.5 Comparison of 2D vs. 3D Methods

In this section we want to compare 2D to 3D methods to determine which is the more powerful representation given the shape similarity methods available. In Section 2.4.5.1 conversion from one format to another is considered. In Section 2.4.5.2 we explore which methods are more commonly used and why multiple formats are still in use. In section 2.4.5.3 we examine the evidence of which methods are better 2D or 3D?

2.4.5.1 Conversion of 2D to 3D models and vice versa

The notion of 2D vs. 3D geometry may be somewhat artificial. Any 3D model can be reduced by one dimension to one or several 2D images though a projection. Also it is possible to reconstruct a 3D image from two or more 2D images using projective geometry. One example of this type of modeling is presented in Pollefeys et al. [75]. Another work based upon reconstruction of solids of revolution of orthographic projections is given in Lee and Han [76].

There have been several surveys on 2D to 3D reconstruction from line drawings. Notably one by Wang et al. [77] and another more recently by Fahiem et al. [78]. The methods reviewed in these papers generally follow a common sequence: find all 2D vertices, from 2D vertices find 3D vertices, from 3D vertices define 3D edges, and from 3D edges construct faces that define the geometry.

Hidden lines are sometimes used to facilitate the conversion. Three major methods are discussed here. The first way that hidden lines are used is to give a clue about occlusion. The hidden lines help determine the internal shape of the part that we could not see if looking at only solid lines without sectional views. This approach is taken by Shi-Xia et al. [79]. In another approach by Cicek et al. [80] the centerline was used to locate the center of a solid of revolution. The half profile of the hidden line was revolved about that centerline to form a solid of revolution. Extrusion was used to construct the outside profile of the part. In the paper by Dimri et al. [81] the hidden lines were used to determine whether a feature should be a protrusion or a depression.

The above showed that although 2D to 3D conversion is possible it is not simple nor without problems. Therefore in this dissertation we suppose that one representation is chosen and all data is converted to that format before we compare complexity between methods.

2.4.5.2 Two dimensional vs. Three dimensional representation

Despite claims that all manufacturing companies are abandoning 2D CAD in favor of 3D CAD the real scenario is more complicated [82]. The report states that although 71% of current CAD users plan on using 3D CAD, 77% that do use 3D modeling also still use 2D CAD. Overall manufacturing companies are adding 3D capabilities but not replacing 2D drafting. Some cited reasons for using 2D drawings include better early engineering development, compatibility from vendors or customers, lack of upper management support for 3D, training time for to 3D conversion, and lack of sufficient graphics hardware to support 3D.

2.4.5.3 Two dimensional vs. Three dimensional shape similarity

Historically 2D shape similarity comparison was developed first, and 3D shape similarity methods were developed later. Generally a shape similarity method is classified as either 2D or 3D as the methods do not transform well from one representation to the other. In this section we would like to answer whether 2D or 3D shape similarity performed better.

Zaharia et al. [83] compared 2D methods to 3D methods. The conclusion was that 3D shape similarity methods did not outperform 2D methods consistently. There were eight methods evaluated. The six 3D methods compared were Cord Histograms, Random Triangles Histogram, DF 3D Par Extended Gaussian Images, Optimized 3D

Hough3D descriptor, Vector quantized O3DHTD, MPEG-7 3D shape spectrum and two 2D methods Angular Radial Transform (ART) and Contour Scale Space descriptor (CSS). While the optimized 3D Hough3D descriptor was found to perform best, it has a higher computational cost. In addition, the 2D method ART performed almost as well as the optimized 3D Hough3D. It was also shown that the 2D method CSS performed about as well and sometimes better than the 3D methods.

A well cited paper specifically in the engineering shape similarity domain is the one by Jayanti et al. [84]. This paper tested feature vector based methods, statistics based methods, and 2D based methods. A total of 12 methods were tested including three 2D methods. In general the 2D methods out performed the 3D methods.

Matthews et al [85] also showed that 3D representations are not more powerful descriptors than 2D based methods once the 2D methods are restricted to orthogonal spaces.

Based upon all the evidence we have, it is the opinion of the author that the performance of similarity searching depends upon the dataset, specific problem, and measurement method used. We do not believe that 3D methods outperform 2D ones in all cases, which was one criterion in choosing 2D images for our methodology. We use wavelet based shape descriptors for similarity measures, which will be introduced in Section 2.5.

Section 2.5 Wavelets and wavelet descriptors

In this section, we introduce wavelets and wavelet based descriptors. In section 2.5.1 we state the background needed to understand wavelets. We formally define wavelets, multiresolution, and filters in section 2.5.2. Section 2.5.3 is an informal comparison between wavelet and Fourier methods and includes some discussion on the practical significance of those differences. In section 2.5.4 wavelet noise removal, compression, and approximation of functions is discussed. Finally in section 2.5.5 we describe the wavelet descriptor.

2.5.1 Background for wavelets

It is assumed that the reader has some familiarity with linear algebra, Fourier methods, and some digital signal processing. See standard texts by Anton [86] Strang [87] Oppenheim and Schafer [88] Hamming [89] and especially Lyons [90]. Weeks [91] has a good basic introduction to Fourier methods and wavelets.

Only the very basics of wavelets will be introduced in this section. The reader is encouraged to consult the good introductory texts available such as Walker [92] or Jensen and Cour-Harbo [93].

2.5.2 Wavelet definition

Wavelets are basis functions for the wavelet transform. We define V_1 as a complete vector space. We also define V_0 as a subspace of the complete vector space, called the scaling function. The subspace V_0 is a lower resolution subspace of the complete vector space V_1. The difference between the complete vector space V_1 and the scaling function subspace V_0 is defined as the wavelet W_0. Therefore wavelets and associated scaling functions constitute a basis for a complete vector space and are defined by $V_1 = V_0 \oplus W_0$. The wavelet and scaling functions are orthogonal complements defined by $\forall w \in W_0, \forall v \in V_0, \langle w \mid v \rangle = 0$ and form a complete basis for all dimensions of the vector space.

The scaling function subspaces form a multiresolutional analysis of the complete vector space such as defined by $V_{-5} \subset V_{-4} \subset V_{-3} \ldots V_0 \subset V_1$. At each level the scaling function and wavelet can be used to construct the next higher level. For example, the V_0 subspace can be reconstructed from the following relation $V_0 = V_{-1} \oplus W_{-1}$ where V_{-1} and W_{-1} are lower resolution subspaces to V_0.

This multiresolution analysis allows us to build a parsimonious yet descriptive model. We can select the resolution best suited to our needs. For the discrete wavelet transform where there are exactly j resolutions for N samples by the following equation $N = 2^j$. For example if we have eight samples we have three possible levels of resolution since $8 = 2^3$ we have the resolution with the full eight samples, a lower resolution with four samples, an even lower resolution with two samples. This may not be an issue with only eight samples however imagine we have over one thousand samples of a signal; it may be preferable to deal with a lower resolution version to speed up information processing.

To enable multiresolution the wavelets and scaling functions must be related to each other in specific ways. The scaling functions are all translations and dilations of a father wavelet. The wavelets are all translations and dilations of a mother wavelet. Both are related to their sons and daughters by the two scale equations. The scaling function two-scale equation is defined by $\varphi_{jk}(t) = 2^{j/2}\varphi(2^j t - k)$. The wavelet two-scale equation exists for the mother wavelet and is defined by $\psi_{jk}(t) = 2^{j/2}\psi(2^j t - k)$. In both equations t represents time or distance from the starting point j indicates the level of resolution and k represents the translation.

The series expansion of the scaling vector is $\varphi(t) = \sum h_\varphi(n)\sqrt{2}\varphi(2t - n)$ where h_φ is the expansion coefficient of the scaling vector. A similar series expansion exists for the wavelet vector and is defined by $\psi(t) = \sum h_\psi(n)\sqrt{2}\varphi(2t - n)$ where h_ψ is the expansion coefficient of the wavelet vector.

Scaling functions and wavelets are defined by the two scale equations but in practice are implemented as filter banks, as seen in Figure 12 and described in Wickerhauser [94] or Rao and Bopardikar [95]. The signal is filtered and downsampled at each level of resolution. Also at each level the signal is run through a low pass (LP) filter or averaging filter associated with the scaling function. This creates a lower resolution of the signal. No information was lost due to this filtering as the signal was also run through a high pass (HP) or differencing filter associated with the wavelet. The signal can be perfectly reconstructed from the LP and HP filters.

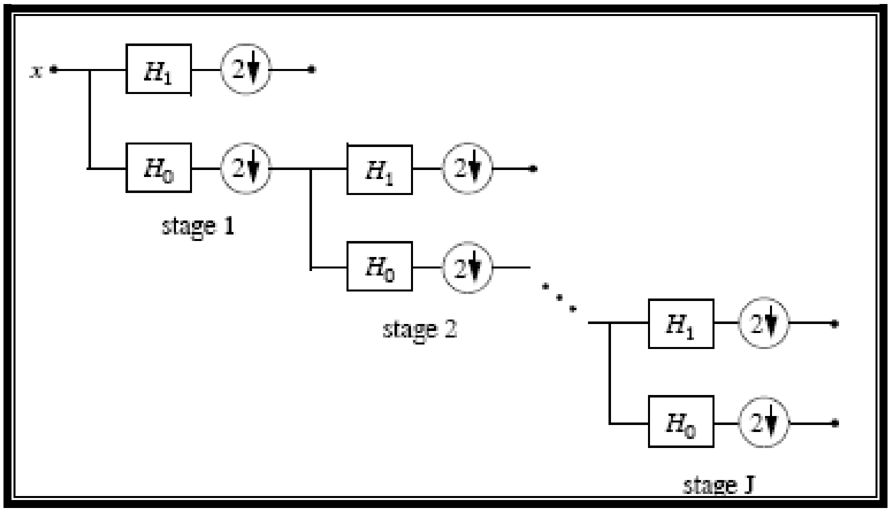

Figure 12 Wavelet filter bank

The Parseval identity for wavelets is useful for relating the energy of the signal both before and after the transformation. The Parseval identity for wavelets is defined by $\int f^2(x)dx = \sum_j \sum_k d_{j,k}^2$ where k indicates wavelet coefficients within a level and j represents the level of resolution as identified in Ogden [96]. The practical significance is that we can eliminate small wavelet coefficients as they contain little energy. After we eliminate the numerous small coefficients, we can reconstruct the signal with a much smaller number of significant coefficients achieving significant compression and enabling our earlier goal of building a parsimonious model. Eliminating all wavelet coefficients above a given level would be equivalent to using the low pass scaling signal only which could be considered a good enough approximation for signal comparison.

2.5.3 Wavelet and Fourier basics

We start by pointing out how wavelets and Fourier based methods are similar. Later we point to key differences that help the reader to understand why wavelets were developed and give the reader some intuitive explanation as to when to use wavelets over Fourier based methods.

Wavelets and Fourier based methods are similar in several important ways. Wavelets and Fourier methods both form an orthogonal basis for representing a signal. Both methods can be used to uncover the frequency components of a signal. Both methods have been used for function approximation, compression, and estimation. Both methods can be used to represent a multiresolutional subspace of the signal. Finally both methods have fast algorithms available.

Wavelets and Fourier methods differ in some fundamental ways that have an effect on their application. It is the opinion of the author that wavelets and Fourier based methods differs in four fundamental ways, as shown in Table 3

Table 3 Wavelet versus Fourier

	Fourier	Wavelets
Basis	Sine and Cosine	Adaptable to problem
Support	Infinity	Rapidly tend to zero
Regularity/Smoothness	Smooth	Smooth to non-differentiable
Transform	Frequency	Time and Frequency

The first difference between wavelets and Fourier based methods is the form of the basis function. Fourier based methods use one fixed basis function namely sine and cosine at various frequencies. Wavelet basis on the other hand are adaptable to a given problem and can assume any shape that is admissible under the definition of the wavelet. This ability to adapt a basis function to a given problem can enable better compression, approximation, and estimation. See Figure 13 for some examples of basis functions.

A second related way that Fourier methods differ from wavelets is regularity or smoothness of the basis function. Fourier methods use a smooth function that is differentiable everywhere. This may work in some applications, but being able to adapt the wavelet basis function to the problem is more desirable for many real world applications.

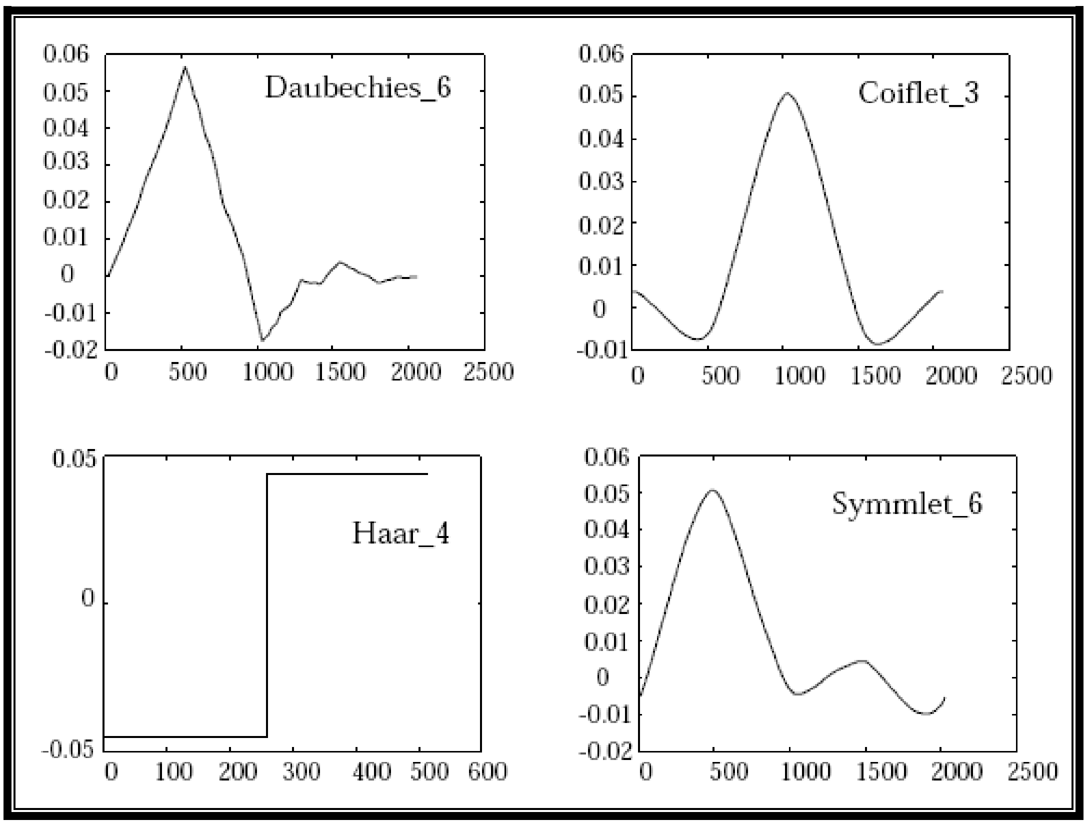

Figure 13 Wavelet basis functions

The third way in which they differ is by support of the basis functions. Fourier based methods all use sine and cosine basis vectors that have theoretically infinite support. Therefore Fourier series coefficients refer to the entire dimension of the vector space for a given frequency. Some wavelet basis functions have infinite support but they rapidly attenuate to zero therefore they are localized in time.

The fourth way they differ could be the most important as it points to one of the main reasons that wavelets were developed. After the Fourier transform, the spectrum reveals the frequency components of the signal through the Fourier coefficients of the Fourier series. In other words after the transform the coefficients are not well localized in time. To combat this, the short time Fourier transform (STFT) was developed see Figure 14. The problem is how we choose the window so that we do not miss important details while still being able to detect frequencies outside the window. There is no good answer to this question. The wavelet transform preserves the time and frequency components of the signal see Figure 15.

Because of these differences and properties we can conclude that wavelet based methods perform better on aperiodic signals and those which exhibit jumps (see Hubbard [97]). These are exactly the types of signals we expect to encounter in our work.

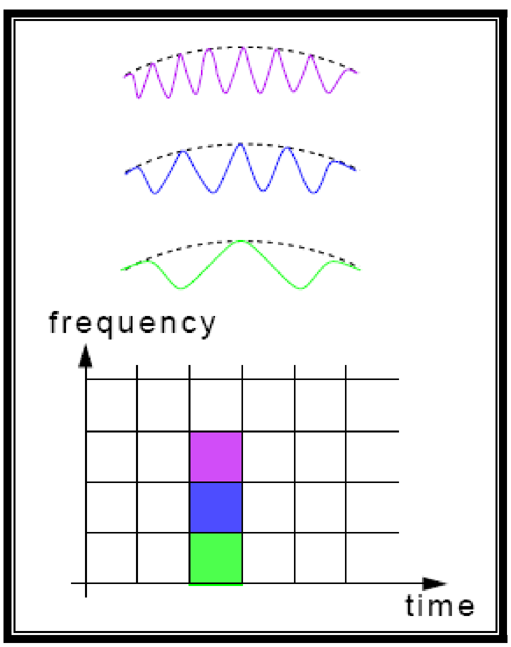

Figure 14 Short time Fourier transform

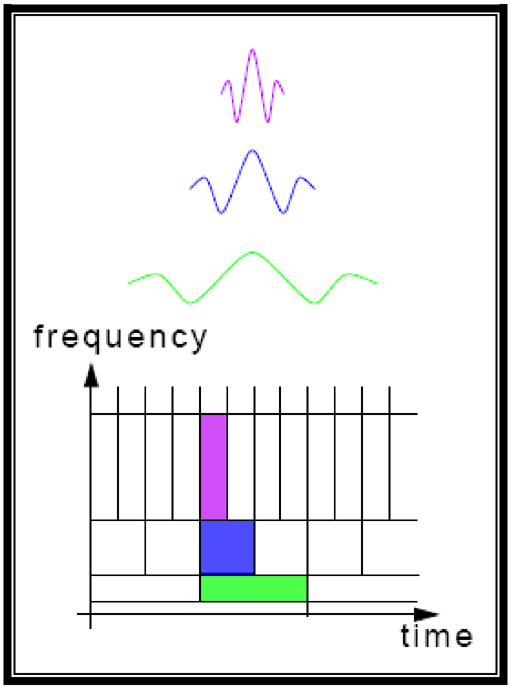

Figure 15 Wavelet tiling

2.5.4 Wavelet noise removal, compression, and approximation

Almost all real world signals contain both the intended signal and noise. Since we are sampling images in our work, we expect some noise. For example an image of a circle is not exactly round. One way to remove the noise is though filtering as explained by Mix [98]. Generally a low pass (LP) filter is used to remove noise. The scaling function is a low pass filter at a given resolution. The selection of the scaling function basis function is directly related to the selection of filters. Recall that scaling functions and wavelets are directly related. See Mallat [99] for a good discussion on the types of signals that are well represented by wavelet and fourier basis functions. In general, aperiodic signals that exhibit jump behavior are better approximated with fewer coefficients with wavelets. This results in higher compression and helps our goal of building a parsimonious model.

2.5.5 Wavelet descriptor

Wavelet descriptors are a multiscale or multiresolutional method as described in Costa and Cesar [100]. Wavelet descriptors are based upon the wavelet series expansion coefficients h_ψ and scaling function series expansion coefficients h_φ explained in section 2.5.2 of this monograph.

In this work we use the wavelet descriptor for several reasons, which will be greatly expanded in the methodology section. Suffice to say, for now, that the wavelet descriptor has some advantages over other shape descriptor methods.

One of the first reasons to use the wavelet descriptor is ease of computation as this method is implemented as a filter bank and available in many software packages such as Matlab see Jensen [93] or Mix [101] and S-Plus see Gao [102] as well as stand alone packages.

The second reason is multiresolution since our goal is to build a parsimonious model using the least number of coefficients possible. This enables another goal of being able to implement this system eventually into a commercial application. Since we are clustering similar shapes together we want to compress the data before clustering to speed up the process of clustering.

A third goal is to use a robust descriptor. Local changes to a signal remain local after the transform. A local disruption only affects those coefficients that are local to that area of the signal. When using Fourier based methods a local disturbance can affect the global coefficients.

A fourth reason is that Fourier based methods are not necessarily unique. After the transform we only know which frequencies occurred, we do not know when they occurred. Therefore two signals could have the same spectrum but represent different shapes.

A fifth reason is the ability to distinguish jumps or singularities in the signal. Since we are matching complexity those jumps or points of high curvature could contain information related to the complexity of the part and consequently the cost.

2.5.5.1 Wavelet descriptor research

Early work on wavelet descriptors was done by Chang and Kuo [103] where they showed that the wavelet coefficients can be interpreted as random variables and used a hierarchical approach on noisy images with good results compared to Fourier methods.

The work by Antoine et al. [104] is a 1D method of shape description of 2D images and shapes. In this work they use the W-Representation to detect and use dominate features for shape matching related to high curvature points.

Osowski and Nghia [105] investigated which descriptors worked well with which classifiers. They looked at both Fourier descriptors and wavelet descriptors using three neural networks. They concluded that the wavelet descriptor outperformed the Fourier descriptor at any noise level and the Kohonen classifier worked the best for all single classifiers. They did develop integrated classifier that outperformed any single method.

Hierarchical active shape models based upon wavelets were explored by Davatzikos et al. [106] They show that the wavelet based method works better when a limited number of samples are available.

A combined elliptical fourier and wavelet based approach was advocated by Lestrel et al. [107]. They argue that the elliptical Fourier methods can be used to represent the global aspects of the shape and the wavelet descriptor can be used on local features. Combining the two methods creates a more powerful descriptor than one method alone.

2.5.5.2 wavelet transformation invariance

A generalized uniqueness wavelet descriptor was developed by King-Chu et al. [108]. They are able to overcome the problem with wavelet descriptors, which are not invariant to starting point.

Wavelet descriptors that are invariant to affine transformations are developed by Rube et al. [109]. They show that the scaling coefficients help reduce noise sensitivity when using the hierarchical clustering method based upon wards linkage models.

The starting point problem of wavelet descriptors was taken up by Kith and Zahzah [110]. They propose four simple methods to normalize the starting point and show the usefulness of each method.

Shift invariance is researched by Chen and Xie [111] using the dual tree complex wavelet and support vector machines. They show their method works better than scaling function wavelets by about five percent on the dataset tested.

Fourier, generic Fourier, and wavelet descriptors were compared by Yadav et al. [112]. They concluded that taking the Fourier transform of the wavelet descriptor coefficients was a method to normalize the wavelet descriptor to affine transformations.

They also found that the wavelet Fourier descriptor was less sensitive to noise than other methods. Another paper by the Yadav et al. [113] concluded that wavelet Zernike moment descriptors outperformed both Fourier and Legendre moment based descriptors.

Kong et al. [114] used a centroid radii model and wavelet descriptor to achieve translation, rotation, and scaling invariance. They tested their method on the SQUID database and showed good results.

Based upon the research that has been done it is the opinion of the author that wavelet descriptors best fulfill our need for the methodology implemented in this monograph.

CHAPTER THREE: METHODOLOGY

Our research objective is to develop a methodology for automatic or semiautomatic cost estimation of injection molds considering part complexity and variation. See Figure 16 for an overview of this process. Shown in Figure 16 are the in-process parts of the methodology. The one-time sections of the methodology are not shown.

We would like to use only those mold designs that are the most similar for cost estimation. Therefore our task is to reduce the dataset of all molds to the dataset of relevant molds for the purpose of cost estimation. An overview was given in Section 1.4. Now we give more detail of the methodology and explore some of the reasons for choosing this methodology.

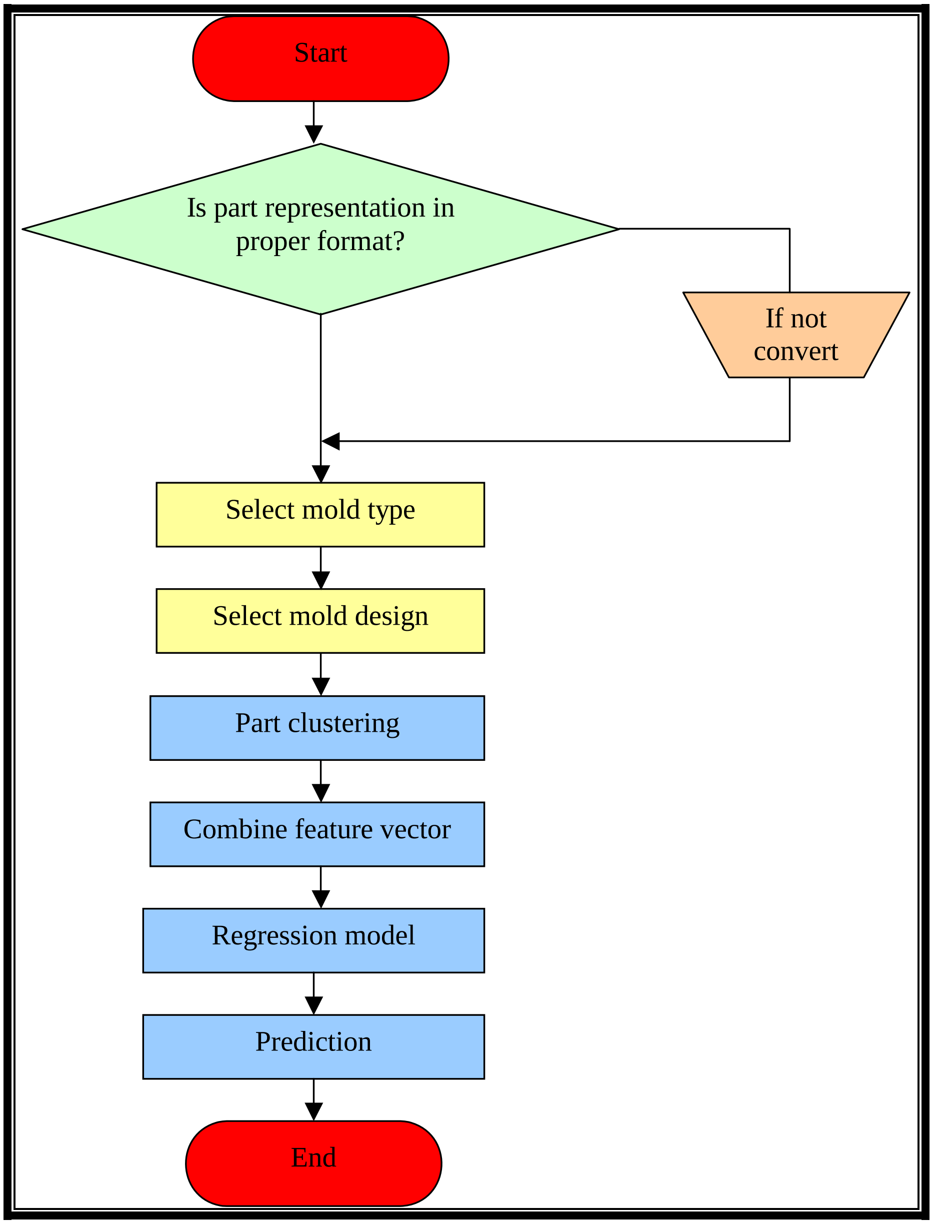

Figure 16 Overview of Methodology

The first part of this process in Section 3.1 is to convert all data to a common format. In Section 3.2 the mold type is selected. In Section 3.3 the mold design is selected. In Section 3.4 the part complexity is clustered. In Section 3.5 three-descriptor types, (regional, topological and wavelet) are combined into one feature vector for each part along with mold type and design data. In Section 3.6 the combined feature vector is used as a basis for a regression model. In Section 3.7 the experimental design and validation plan is covered. In Section 3.8 the data problems and remedies are discussed.

Throughout this work we are using limited data. We assume that most researchers working on this problem will encounter this same problem. Therefore our methodology was chosen to deal with limited data. This limited data manifests itself when putting together the feature vector.

One of the challenges when selecting a feature vector is the curse of dimensionality. This may be encountered when we partition a dataset. The number of observations that meet all the criteria of the partition reduces the number of observations in a particular partition, and makes prediction more difficult. See Hastie et al. [115] or Bishop [116] for a more detailed description of this phenomenon. Due to the curse of dimensionality we strive for parsimonious models with as few variables as possible, which still well describe the phenomenon. This could be considered a practical application of Occam's razor. This was one of the main reasons we compressed the observations in the wavelet descriptor to a minimum and further reduced the dimensionality to the cluster of the shape. Now that we know why we selected the method, we can now move on to the practical application of the methodology.

Section 3.1 Conversion of data to a common neutral format

The purpose of this section is to explain why the format was chosen in Section 3.1.1 and the mechanics of how the formats are converted to the neutral image format in Section 3.1.2.

3.1.1 Why 2D images were chosen as the neutral format

One unique feature of our work is that we convert all part data to a neutral format as images prior to further processing. There are several reasons for this.

If the methodology will eventually be developed into an industrial application, 2D images can be displayed in a more natural way. No special programs are required to display the part and mold in 2D images, as opposed to 2D or 3D CAD models. Integration with the cost estimation core of our process is much easier.

The second advantage is that the size of images and the data processing power needed to display them is reduced versus 2D or 3D CAD. Because we are just displaying images no special math coprocessors, graphics engine, nor special graphics cards are needed.

The third reason is that there are many algorithms and programs available to process images. Many programs that can do standard image processing or computer vision could be used.

The fourth reason is that 3D is not a more powerful representation than 2D. This was explored in Section 2.4.5.3. It has not been proven that 3D has any advantage over 2D methods for shape similarity.

The fifth reason is ease of conversion. Although conversion is possible between 2D and 3D as outlined in Section 2.4.5.1 it is difficult. The conversion is generally done semi-manually and is labor intensive. Indirect conversion from 3D to 2D is less labor intensive than 2D to 3D. Almost all 3D CAD programs have a 2D drafting mode. In this mode 3D CAD is very similar to 2D CAD. Therefore conversion from 3D drafting mode to 2D CAD is easier. The third common data format is blueprints. Blueprints can be scanned and converted to 2D images using image-processing methods.

3.1.2 Methodology of conversion to a neutral format

The first assumption of our data is that the non-geometric information has been removed either before or after the conversion process. The key is that the data was preprocessed prior to the part similarity comparison.

The conversion process itself is dependant on the format of the data received. For this dissertation, it is assumed that one of three formats is available. The three formats are blueprints, 2D CAD, or 3D CAD. However any format that can be converted to an image can be used.

If data were received as a blueprint, it would be scanned and converted to a 2D image directly through image-processing software. Some preprocessing such as thresholding may need to be done to obtain a cleaner image before its use. Thresholding is an image processing technique used to separate the background of an image from the lines.

If a 2D CAD file is received, it will be converted either by direct export to bitmap or screen capture. The specifics would depend on which 2D CAD software used to create the image.

If a 3D CAD file is received, this may require further preprocessing. For this work we assume the designer has used the 2D drafting mode in the 3D CAD program to create a drawing. If the designer has not made a 2D drawing then this process would have to be done prior to use in our methodology. Once a 2D drawing is available the conversion is similar to 2D CAD. The specifics depend on which 3D CAD program was used to create the geometry.

Section 3.2 Mold Type Selections

Selection of mold type is based upon both geometric and non-geometric information. The non-geometric information is equally important to the selection process. Therefore this process must remain at present a selection process done by an experienced bidder familiar with mold types. It is important to note that in our methodology the user selects the mold type. The mold type is not selected by the methodology. Comprehensive examinations of all of the factors that may affect mold type selection are beyond the scope of this document. However some factors that would be considered by the bidder are part material, estimated annual usage, lifecycle, geometry, surface finishes, geometric dimensioning and tolerancing.

We restricted ourselves to mold types of conventional, MUD, and modular because they were the most common types and would cover a large proportion of all molds.

Section 3.3 Mold Design Selections

Mold design is a process best left to a subject matter expert in mold design. Therefore we believe that using a manual section process as in our methodology is preferred.

There are several good texts available on mold design and construction such as the one by Kluz [10]. Although part demolding and part ejections are primary considerations, other factors are equally important. These other factors are beyond the scope of this dissertation. Some factors could be gating location, venting considerations, parting line locations, ease of repair, and many other factors that have a bearing on the mold design.

We restrict ourselves to mold designs of straight, spring ejector, and cam action because they are the most common types and would cover a wide variety of molds.

Section 3.4 Part Complexity Matching

In this section we describe the proposed process of part matching. For this work part complexity will be obtained from images of the part using several descriptors. The process consists of many steps. In Section 3.4.1 we read the images and detect the boundaries. For this work we propose to use one view of the part. This view will be the top view looking directly into the parting line. Section 3.4.2 describes the image normalization. Section 3.4.3 outlines the process by which we describe each boundary.

3.4.1 Read image and detect boundaries

Reading the images can be done with standard functions in the Matlab image-processing library. A typical image is shown in Figure 17.

After the images of the part are read the boundaries will be traced with third party libraries. Boundary detection or contour following is a common technique used in image processing. One good reference can be found in Parker [117].

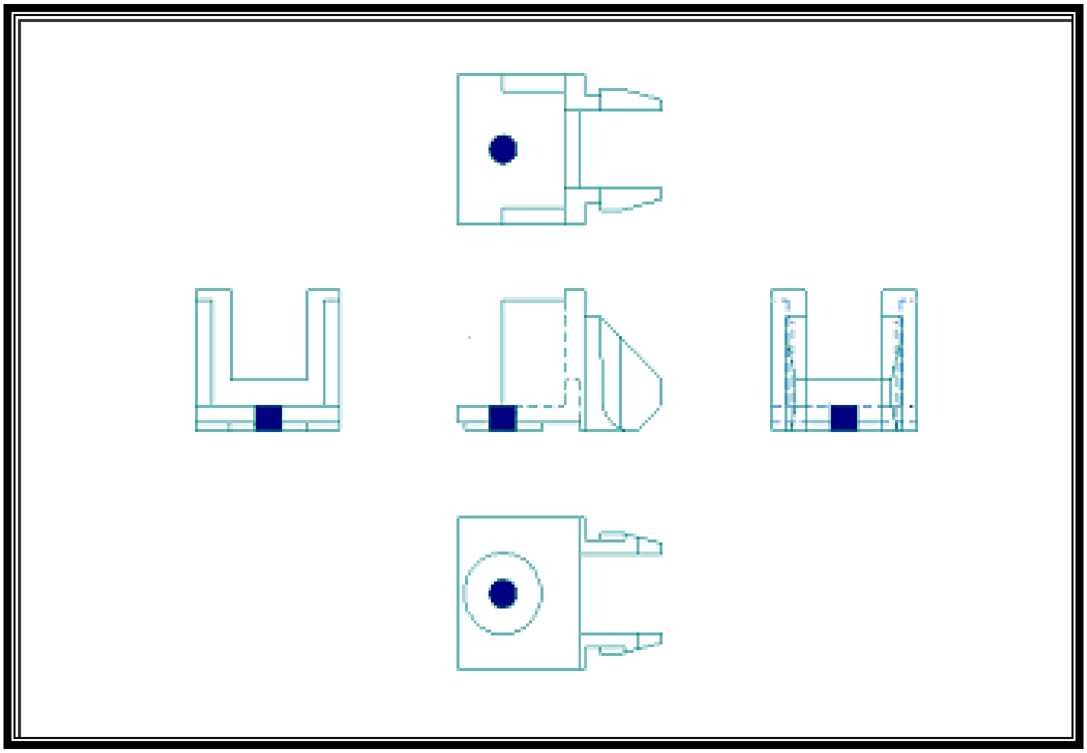

Figure 17 Typical image of part

The next step in this process will be segmentation. Segmentation is the process of dividing the 2D image of the part into regions as described by Gonzalez et al [9]. The purpose of this is to separate each view of the part in the image and isolate it. We want to deal with one view of the part at a time at this stage. Namely we want the top view looking into the parting line.

The next step of the process will be to normalize the images; this is done in order to create common translations and transformations for comparison.

3.4.2 Normalization

The first part of the normalization process is to translate the image to zero point s of x and y coordinates and crop the image. The translation assures a common location and cropping creates a bounding box around the part. In Figure 18 the upper left corner is the origin.

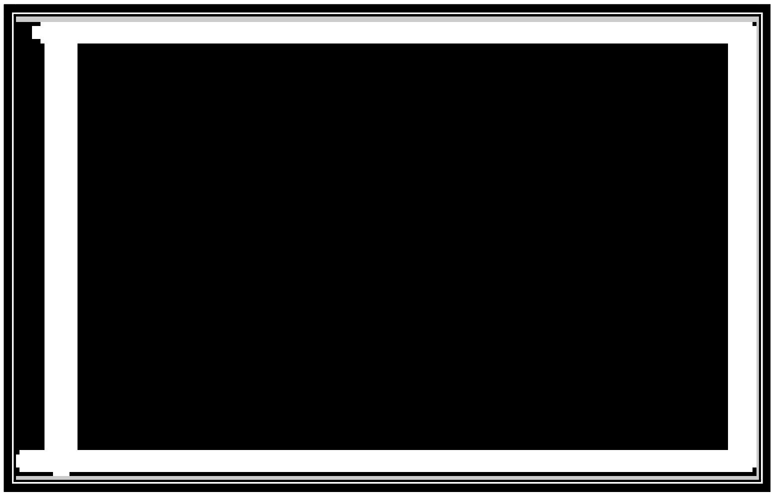

Figure 18 Cropped image

Rotation normalization is done so that the x-axis is aligned with the major axis of the part. For this work we assume that the coordinate axis was used to generate the drawings and therefore straight edges lie along either the x-axis or y-axis so that either the image is rotated ninety degrees or not rotated at all.

Mirror or reflection normalization is done by comparing the center of the bounding box to the center of mass of the boundary. For this work we assume that all images have been normalized so that the upper left quadrant contains the largest mass according to the largest boundary or profile of the image in a given view.

Scale is normalized so that a common pixel to length ratio is maintained for example 1000 pixels for one inch of length. The reason for this is that some variables such as area are likely to be significant for cost estimation, and the images therefore must be normalized so that areas are consistent from image to image.

3.4.3 Describe each boundary

It is important to note that the goal of using geometry is an indirect way to measure complexity, which is related to cost. This is slightly different from purely measuring shape similarity as reviewed in Section 2.4. Our goal is to create a feature vector of variables that are related to cost. Each boundary may have wavelet descriptors as well as other descriptors such as size, perimeter length, etc for each view of the part.

There are several ways to measure shape complexity. The first way to estimate complexity and cost is using regional descriptors such as size, eccentricity, bounding box, etc. The second way to estimate complexity is through topological information, for example, the number of enclosed boundaries, Euler numbers, Betti Numbers, and Genus. The third way is through wavelet descriptors.

3.4.3.1 Regional descriptors

Regional descriptors are properties that describe a boundary. A boundary is defined as a connected component. Connected components are continuous and so are boundaries.

There are several regional descriptors that are useful for this work. One regional descriptor that has potential to be significantly related to cost is area. Area is defined as the number of pixels in an enclosed boundary. It is well known in the molding industry that two parts with similar shapes but of different sizes will have a different cost structure. Therefore it is reasonable to assume that area will be a factor in cost estimation. A second possible regional descriptor is eccentricity. Eccentricity is the ratio of the distance between the foci of the ellipse and its major axis. Eccentricity is a scalar defined on an interval of zero to one. A value of zero would indicate a circle whereas a value of one would indicate a straight line. Therefore its application will be most useful in determining whether or not a given boundary is a circle or how far it deviates from it. A third is regional property image. The regional property image is a boundary with the same area as the bounding box of the region. It may be used to determine whether or not the boundary in question is a rectangular shape.

3.4.3.2 Topological descriptors

Besides geometry, the second way to capture complexity is through topographical information such as Euler numbers and the number of boundaries of the image. Recall that topological descriptors describe a view, whereas regional descriptors and wavelet descriptors describe a boundary.

The Euler number is the number of boundaries in the image minus the number of holes in those regions. This may be useful in determining complexity of the image.

The number of boundaries is thought to be significant, as intuitively an image of a plastic part with few boundaries would imply a simple geometry of the plastic part. This would translate into a simpler mold with a lower cost.

3.4.3.3 Wavelet descriptors

The third descriptor type will be the wavelet descriptor. Please see Section 2.5 for a detailed overview of the wavelet descriptor. It is done in several steps see Figure 19. Shown are only those steps that would be required in process. Not shown are some preliminary steps such as the selection of the wavelet.

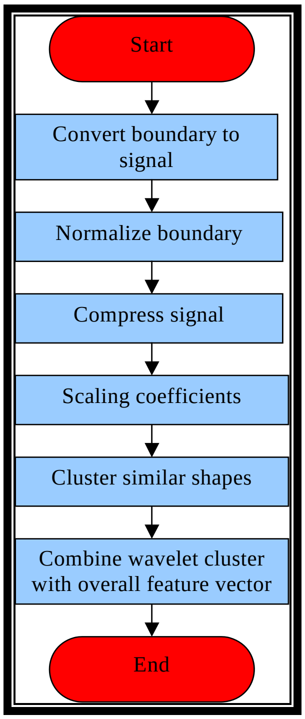

Figure 19 Wavelet descriptor process

In this section it is assumed that the images have been read into memory and the boundaries have been detected. This boundary is normalized for starting point. The starting point is the pixel on the boundary with the shortest Euclidean distance to the origin after translation, rotation, and mirroring.

The general algorithm is as follows. (1) Choose which wavelet to use; (2) convert the boundary to a signal; (3) normalize the length of the boundary; (4) compress the image; (5) calculate the scaling coefficients; (6) cluster similar shapes together; and (7) make the cluster part of the overall feature vector for cost estimation.

The first step will be the selection of which wavelet to use. For this work we use the Daubechies wavelet with four coefficients. This was done for several reasons. The first is that the Daubechies wavelets are some of the most studied wavelets. It is our desire to remain transparent so that others may extend what has been done. A second reason is that Daubechies wavelets have a property that is helpful to our work. See Property III in Walker [92]. Define J as the length of the wavelet filter. If the signal is approximately equal to a polynomial of degree less than J/2 over the support, then the wavelet coefficients will be approximately zero. For a Daubechies wavelet filter with four coefficients, this means that the wavelet coefficients will be approximately zero for all areas of the signal that are straight lines. Therefore most of the energy of the signal is contained in the scaling function coefficients, and the scaling function alone is a good approximation of a signal composed mainly of straight lines. These are the types of signals we expect to encounter for mechanical components. For curved sections and circles preliminary experiments indicate that these signals are also well represented by the scaling function coefficients. This is because circles are represented as sine waves in our methodology. Those sine waves contain approximately straight sections that can be approximated by the scaling function alone.

The second step after the wavelet has been selected is to convert the boundaries to signals. This is not difficult, when a boundary is traced, the x and y coordinates of the boundary are kept in an array. The first column of the array is the x-axis values or distance from the origin. The second column is the y-axis values or distance from the origin. To convert to a signal in Matlab we just pull off the column that we need and transpose that column to be the first row of an array. In Matlab a row vector is a signal.

However we also need to keep track of the distance from the starting point of the boundary.

For this work it was determined that the distance from the starting point would be defined as the parameter t. This would be the horizontal axis in Figure 20. The vertical axis is the changes in the x-axis as the boundary is traced see Figure 20 labeled X (t).

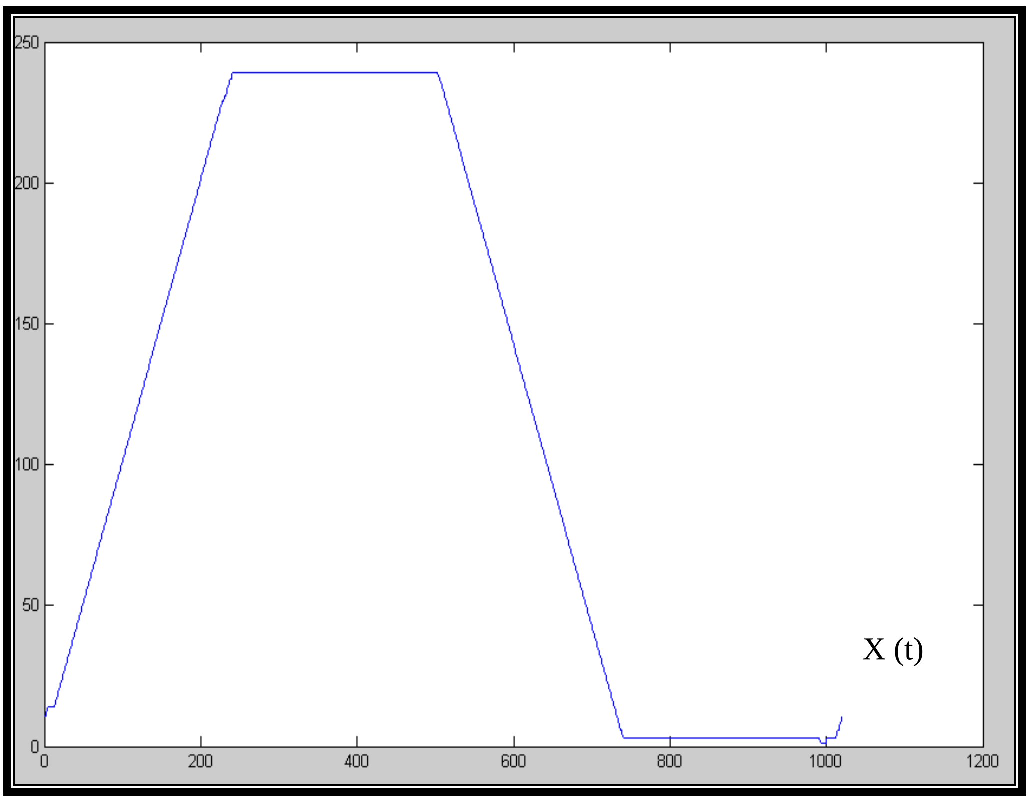

Figure 20 Boundary converted to a signal

The third step is to normalize the boundary length. The image is scaled so that the boundary is a standard length, which is a power of two. In preliminary tests we used a boundary length of 1024 pixels. After the image is scaled so that the boundary length is close to the target of 1024 pixels, a mirroring process is used to minimize edge effects when doing the convolution.

The fourth step is to compress the signal. It is critical to compress the signal for this work as it reduces the time required for further processing. Even after compression the signals retain their essential shape. Similar shapes of the signal indicate similar boundary shapes and dissimilar signals indicate dissimilar shapes of the boundary. A typical rectangular boundary and its associated compressed signal are found in Figure 21 and Figure 22, respectively. A typical circular boundary and its associated compressed signal are found in Figure 23 and Figure 24, respectively.

Deciding on how much to compress a signal with wavelets is a decision on how many scaling function coefficients are needed. This question is closely linked with how much we can compress the signal and still retain its essential shape. Recall that at each level of the filter bank the signal is downsampled and the energy is compressed into a shorter signal with half the coefficients. The number of coefficients will be determined by informal experiment, as it is data dependant. Generally four to sixteen coefficients are considered sufficient as described in Chuang and Kuo [103]. In our preliminary experiments we converted the boundary to a signal and compressed a signal with one thousand and twenty-four coefficients to just seventy. Sixty-four of these are the scaling function coefficients and six were due to the padding added to minimize the boundary effects from the convolution.

Because of these preliminary findings, it is not thought that we need to use a thresholding technique to determine which wavelet coefficients to keep. Thresholding is a technique that allows us to keep significant coefficients regardless of frequency or time.

For a good discussion on thresholding techniques please see Ogden [96]

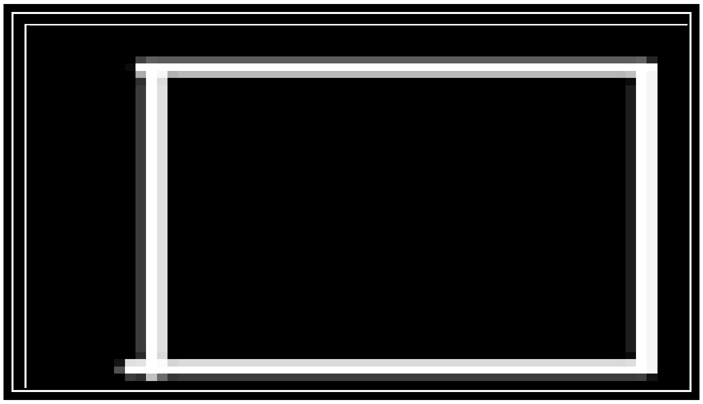

Figure 21 A typical boundary on a part print

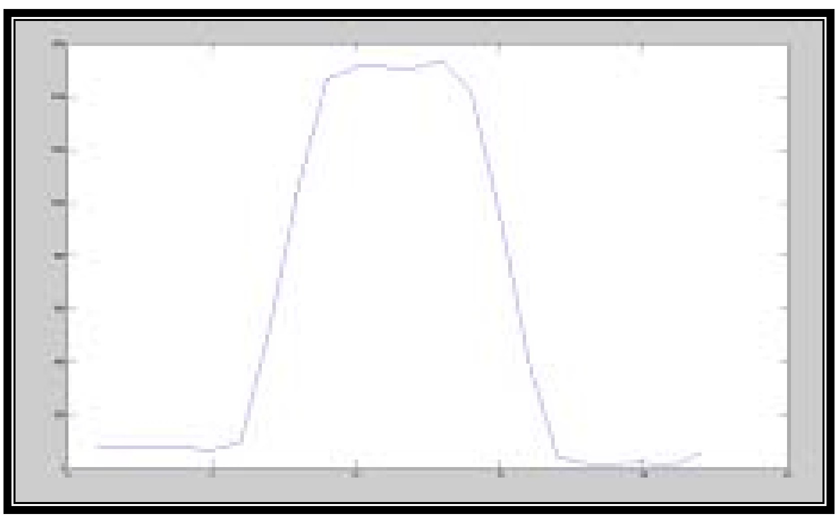

Figure 22 Compressed boundary

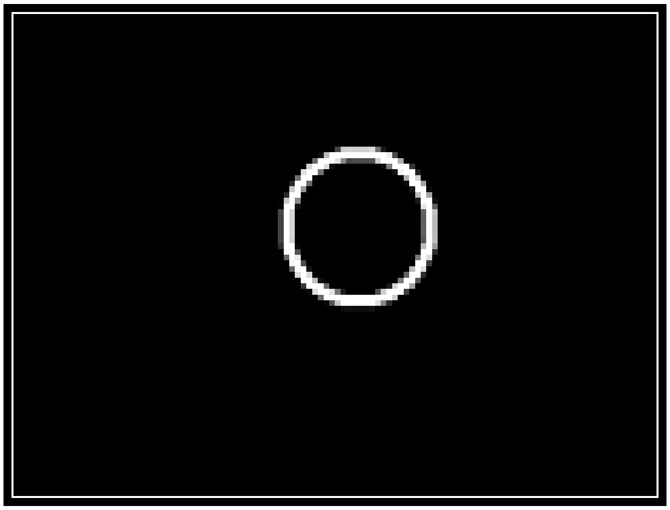

Figure 23 Circular boundary

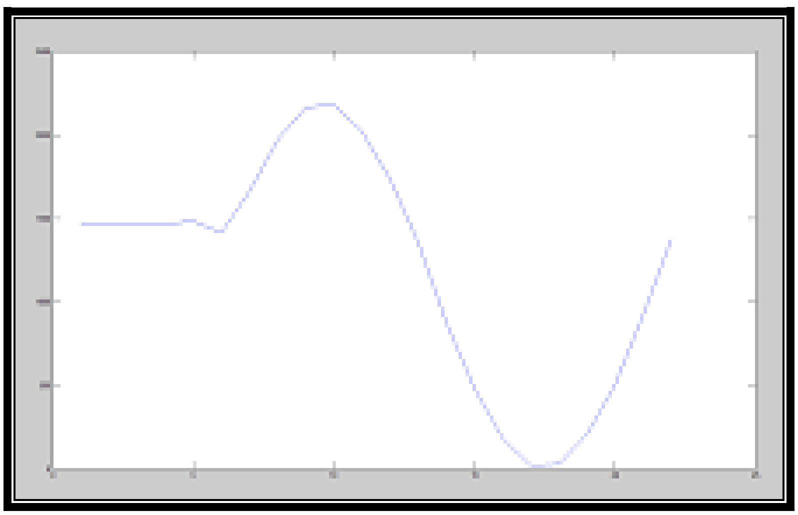

Figure 24 Circular signal

After the signal has been compressed, the fifth step is to represent the signal from the scaling function coefficients. These coefficients should contain the bulk of the energy of the signal and therefore are a good representation of the original signal. Shown in Figure 25 are the scaling function coefficients of the compressed signals for some test images. Note that there were seventy coefficients although only seven are shown.

	ca5{1,1} 1	ca4	1	2	3	4	5	6	7	
1	'Copy (2) of test3.bmp'	1	1469.7314	1472.4171	1472.0111	1467.9503	1487.3249	1416.4509	1665.1739	1
2	'test7.bmp'	2	1676.4602	1676.8527	1676.8722	1677.305	1671.9267	1696.7838	1589.4663	1
3	'test6.bmp'	3	1676.4602	1676.8527	1676.8722	1677.305	1671.9267	1696.7838	1589.4663	1
4	'test5.bmp'	4	1676.4602	1676.8527	1676.8722	1677.305	1671.9267	1696.7838	1589.4663	1
5	'test4.bmp'	5	1785.5805	1785.0021	1785.3759	1785.277	1785.6549	1783.1919	1801.9596	1
6	'test3.bmp'	6	1469.7314	1472.4171	1472.0111	1467.9503	1487.3249	1416.4509	1665.1739	1
7	'test2.bmp'	7	73.6585	73.9219	73.6778	74.9209	67.5412	95.4865	517.7308	1
8	'test1.bmp'	8	82.591	82.6176	82.4685	83.9967	75.0135	108.1156	540.0476	1
9	'Copy of test7.bmp'	9	1676.4602	1676.8527	1676.8722	1677.305	1671.9267	1696.7838	1589.4663	1

Figure 25 Wavelet coefficients

After we have the scaling function coefficients the sixth step is to cluster together similar signals. Recall that clustering similar signals is the same as clustering similar boundaries or shapes of boundaries. Shown in Figure 26 are preliminary results when using a k-means clustering algorithm on the wavelet descriptors. The left column is the cluster that the image belongs to and the right column is the name of the test image. Other clustering methods, which could be used, are considered in several references. One reference is Duda et. al [118]. Another text is the one by Russell and Norvig [119]. A third reference to do exclusively with self organizing neural networks can be found in Carpenter and Grossberg [120].

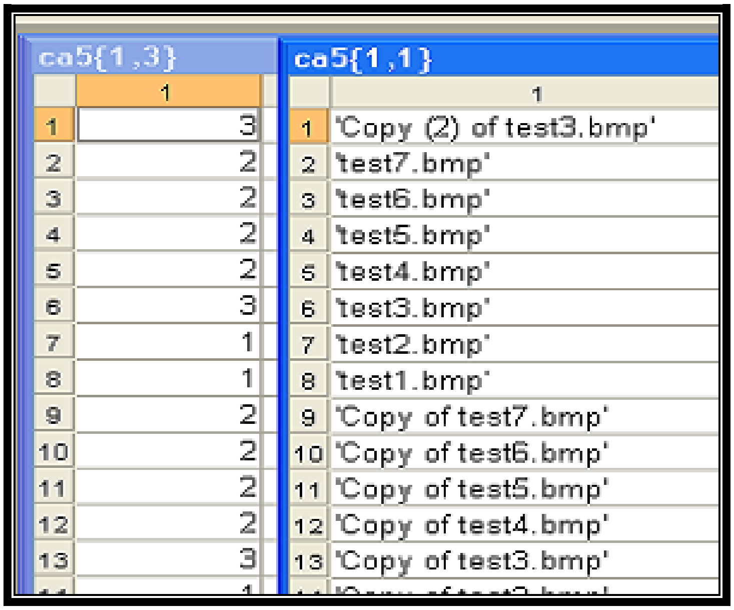

Figure 26 Clusters

The seventh step is to combine the wavelet descriptor, regional descriptors, and topological descriptors into one feature vector. This overall descriptor will be combined with the mold type and design data to form exactly one observation for each part. That is the process outlined in section 3.5.

Section 3.5 Create Overall Feature Vector

At this stage we have described the mold type, mold design and part complexity. Our task now is to combine what we know into one uniform descriptor. It is our desire to build a parsimonious model that accurately reflects the relationship between the independent variables and the target variable. Due to this fact the feature vector has been created in stages.

Recall that the bidder has already selected the mold type and design from a drawing of the part manually. Our methodology does not select these factors for the bidder. We also know the cost of similar molds. The cost will serve as our target in the supervised learning method in section 3.6. See a partial feature vector in Figure 27. The mold feature vector will be combined with feature vectors for part descriptions. It is important to realize that all descriptors of the mold and part form exactly one observation. The part and mold descriptors are the independent variables and cost is the dependent variable in the regression model.

Mold descriptors			
Mold ID	Mold Type	Mold Design	Cost

Figure 27 Feature vector

The next step is to describe what we now know about the part from the descriptors calculated from the images of the parts. This was a core part of our research.

From the view of the part we have topological descriptors such as Euler numbers, and number of boundaries. In addition, regional and wavelet descriptors are used to describe the boundaries of the part itself from the 2D image. Each boundary within the view is described with regional and wavelet descriptors. The clusters from the wavelet descriptors for x(t) and the regional descriptors form a complete descriptor for each boundary.

Section 3.6 Regression

In this section our goal is to use the information in the feature vector from Section 3.5 to determine the estimated cost of the mold from the information about the mold and the part descriptors. The multiple regression technique was chosen as the supervised learning method for several reasons.

The first reason was that regression provides a way to combine multiple data types. We expect our feature vector to have a mixture of data types of continuous, ordinal, and categorical data.

The second reason is that our desire is to weigh the variables. Each variable may have a correlation to the dependant variable of cost and we would like to know how they are related.

The third reason was to provide mean prediction along with prediction intervals. We would like to know how precise our estimate might be given what we know about the mold and part.

Section 3.7 Experiment and Validation plan

The first important consideration for this work is that we are dealing with observed data and not experimental data. We cannot in most cases change the factors and observe their outcome. We also have a limitation in the number of observations available in our dataset. Because of these two factors we do have some limitations on what we can do. We assume for this section that outliers have been investigated and the data has been transformed to make the relationship between independent variables more linear with the target variable, which is cost. A good discussion on experimental design and mathematical statistics can be found in Wackerly et al [121].

When investigating the relationship between continuous variables and the target we will primarily use correlation to investigate variables. Both Pearson correlation and Spearman correlation may be used.

For categorical variables our main tool is completely randomized analysis of variance (ANOVA). We will also use an F-test to determine whether or not there is a difference due to the factor we are exploring in that particular test. In addition, nonparametric tests such as Kruskal-Wallis may be used if there is evidence that the population violates one or more of the assumptions of ANOVA such as normality and equal variances.

3.7.1 Experiments

There are several factors we would like to investigate from the observed data. We would like to test some assumptions we have made and determine which factors are the most important for cost estimation. We would first like to test whether or not mold type has an effect on the cost. Second we would like to test whether or not mold design has an effect on cost. A third assumption is that the shape of the part itself has an effect on cost.

Both the Pearson correlation coefficient and the Spearman's rank correlation may be used to investigate the relationship between continuous variables and the cost of the mold. We may test Euler number, number of boundaries, eccentricity, and area.

3.7.2 Validation

After the regression model has been built, validation will be performed. Our main method for this will use a hold out sample technique. These jobs have not been used in model building and the actual cost will be compared to what the model would have predicted for that observation. In our test we plan on using the leave-one-out strategy.

Section 3.8 Data problems and remedies

3.8.1 Observational vs. experimental data

Because we have an observed dataset, not an experimental dataset, we cannot explore variables in isolation. The variables are not in most cases independent for an observed dataset see (Mendenhall and Sincich **[122]**). This is more of a problem in explanation than prediction. Furthermore, for some variables such as number of cavities, it would be possible to design an experiment whereas for others such as WaveClust and RegClust this would be more difficult.

3.8.2 Outliers

Since we were dealing with three populations of different mold types outlier detection was placed later in the methodology. In other words, outliers are difficult to detect when we have a dataset that may be heterogeneous.

For example from the graph of all cost data in Figure 28 there appear to be many outliers in our dataset. This is not necessarily true. We could have a mixed distribution and each potential outlier may be perfectly reasonable if placed into its partitioned dataset. Therefore outlier detection was delayed until our datasets were homogeneous.

This was done so that we could detect true outliers separate from apparent outliers at a higher level of data aggregation. These higher levels of aggregation were heterogeneous datasets and not representative of the true relationship of the observations or variables of the true model.

There were several techniques used to detect outliers. We used scatter plots, box plots, and more formal tests such as high leverage points and high-standardized residuals. High leverage points were given special emphasis and investigated using graphical methods and conversations with the molder that provided the data.

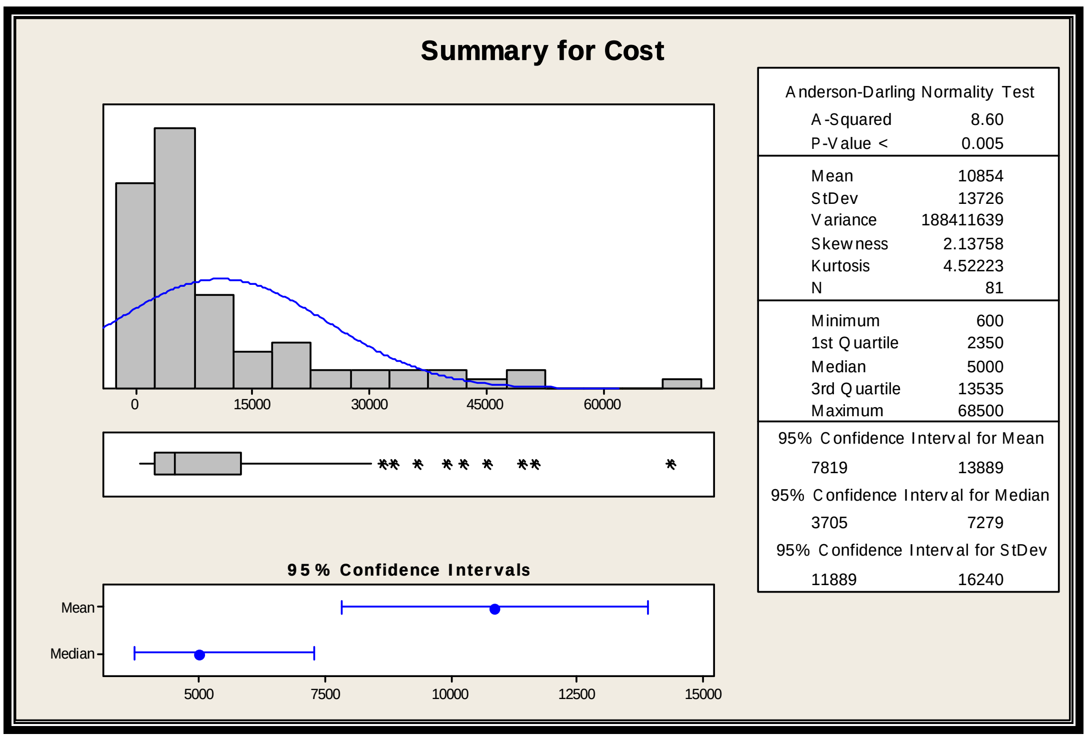

Figure 28 Apparent outliers

3.8.3 Non-constant variance

A summary of the target variable cost reveals that we have a very wide variety of costs. Our original dataset includes variables from \$600 to over \$60,000 or a low to high factor of over 100.

Problems may arise when we try to model data with such a wide variance. The first potential problem would likely be non-constant variance that manifests itself when modeling over a wide range. There are three common ways to deal with this (Jennrich [123]). The first way is to transform the target variable. While this may stabilize the variance it may also introduce bias into the model. Another potential solution is to use weighted regression, which is viable but more complex to implement and interpret. The third method is to partition the data before modeling. We chose to partition the dataset into smaller regions before we build the final models.

3.8.4 Nonlinearity with target and interaction effects

For each potential variable in the model we investigated both the quadratic and interactive effects. After many experiments we found that interactive effects were in general neither informative nor predictive. However second order effects were frequently useful therefore they were included in the model.

3.8.5 Noise

The first limitation is that we are dealing with limited data. We must also consider that the dataset was based upon quoted prices not actual hours. Therefore not all costs used may reflect the true cost. Some of the molds may have been intentionally or unintentionally overbid or underbid. We must also take into consideration that factors outside our model could have a significant effect on cost. They include but are not limited to flash, tolerance, surface finish, and decorative cavity graphics such as logos etc. We also know that during slow times molders offer molds at a discount. Another reason for a discount may be due to the relationship between the molder and the customer or to get the profitable parts orders.

3.8.6 Variable selection

Variable selection will be done in several ways. By using multiple methods it is hoped that we can determine true correlation from apparent correlation.

The first method will be visual. Scatter plots will be used to determine whether or not continuous variables are related to the target of cost. Box plots will be used to determine whether or not categorical variables seem to be related to cost.

The second method will be correlation. The correlation coefficient will be used along with gross measures of correlation to determine whether or not a variable is related to the target variable, which is cost. See Kachigan [124] for a discussion on correlation.

The third way will be through stepwise regression and best subsets regression. See Jennrich [123] for a good discussion of stepwise and best subsets regression. Stepwise procedures include variables to enter one by one at a given p-value whereas best subsets considers that different variable combinations may yield just as good results.

However the final decision as to which variables to include were arrived at by experiment by building multiple models and observing the results. We do this because we may know or have knowledge apriori to modeling or suspect that certain variables should be included or excluded based upon reason.

3.8.7 Multicollinearity

One variable Num_Bound number of boundaries is correlated with other variables such as WaveClust and RegClust. NumBound confused the effects of these variables without contributing to the predictive ability of the model.

3.8.8 Data preparation

Data preparation will be done in several steps. This is an interactive process and may take several iterations in order to transform the data into the proper format see (Pyle [125]).

The first step will be to fill in any missing values. An attempt will be made to justify why the values for any missing values were selected.

The second step in this process will be outlier detection. For the purposes of this dissertation outliers will be detected in two ways. The first way is visually through the use of scatter plots and box plots. The second way will be statistical selection. If an observation is more than three standard deviations outside the normal, it will be considered an outlier.

The third step will be residual analysis. We will consider linearizing transforms in order to make the independent variables linear with the target variable, which is cost. We select the transform by observing the shape of the residual plots. This is sometimes known as the bulge rule see Jennrich [123]. An attempt may be made to standardize the variance with variable transforms. However this can introduce bias and therefore may not be beneficial to our application.

3.8.9 Random sampling

One potential problem with the data selected is that it may not represent a true random sample. We selected 81 of the most recent jobs. A sample of this size at the job shop that provided the data encompasses many months work. As such we have many mold types, designs, and customers from different industries. We think of this dataset as a large test sample but not necessarily random. We did not use a stratified sample to create a sample that represented the population of all molds. Another problem is that all the data came from one molder. We assumed since this job shop is winning some jobs while loosing others they are competitive with prices from other molders. We do not believe the mold prices are biased.

CHAPTER FOUR: IMPLEMENTATION AND RESULTS

Section 4.1 Implementation

Implementation of the methodology was done in several steps using both commercially written software and custom software written by the author of this dissertation.

Microsoft Access was used to store information about the mold and cost. These variables were manually entered into our dataset as shown in Figure 29. These variables are not considered core to our research.

Matlab was used to implement the image processing section of our methodology. We use regional descriptors, wavelet descriptors, and topological descriptors to estimate cost from images. Matlab standard libraries were used for reading images, boundary detection, calculating the wavelet coefficients and the hierarchical clustering. A library from Gonzalez, Woods and Eddins [9] was used extensively for calculating the regional descriptors. The image normalization, creating the datasets and combining the various information sources was custom written as Matlab M-file scripts.

Ultimately all the information from the manual data entry and the automatic data collection was combined into Minitab for the statistical computations and regression analysis.

ID	Name	Num_Cav	Design	Type	Cost
1	K-1448	1	Straight	Mod	2809
2	K-1450	1	Straight	Mod	2500
3	K-1449	2	Straight	Mod	3700
4	K-1463	1	Spring	Mod	5000
5	K-1451	1	Straight	Mod	3000
6	K-1464	8	Straight	Mod	1500
7	K-1465	1	Straight	Mod	1509
8	K-1466	6	Straight	Mod	1509
9	K-1469	2	Straight	Mod	2500
10	K-1472	2	Straight	Mod	2200
11	K-1454	1	Straight	Mod	2300
12	K-1457	1	Spring	Mod	2400
13	K-1458	1	Straight	Mod	2400
14	K-1461	2	Straight	Mod	3900
15	K-1473	1	Straight	Mod	1900
16	K-1474	2	Spring	Mod	3000
17	K-1475	1	Straight	Mod	1500
18	K-1476	1	Straight	Mod	1800
19	K-1478	1	Straight	Mod	1800

Figure 29 Mold database

Typical data automatically collected from the part print is shown in Figure 30. Name is the unique name of the mold at the company that supplied the data. Columns 1 and 2 in the dataset named *record* are the image and boundary number for one image. The dataset labeled *T* is the wavelet cluster. The data *T2* is the cluster based upon the regional descriptors. The variable *Eccent* is eccentricity of the boundary. Through our investigation it was found that boundaries with an eccentricity below 0.40 were highly correlated with circular boundaries. Therefore the variable Eccent was used to determine the number of non-circular boundaries (NonCir) in the image. Round boundaries are easily machined and have a lower cost than non-circular boundaries.

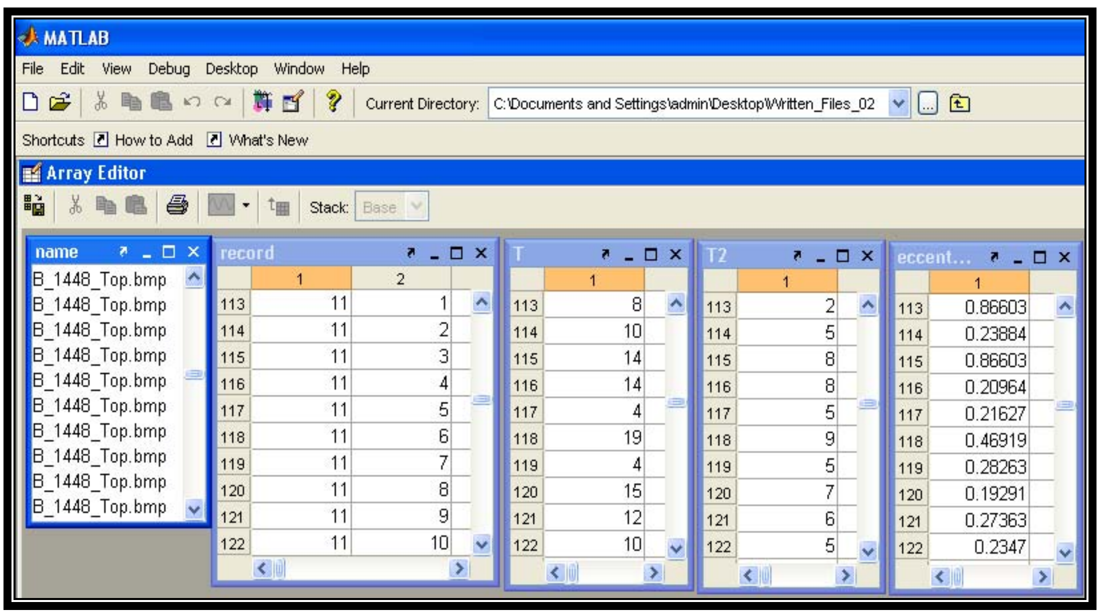

Figure 30 Matlab database

Section 4.2 Results

The dataset consisted of 83 molds of type conventional, master unit die and modular. There were 16 conventional molds, 18 master unit dies and 49 modular molds in our dataset.

Our goal is to create one master dataset used to build the regression model from the three disparate data sources. Shown in Figure 31 is the master dataset used in modeling.

	Name	Num_Cav	Design	Type	Cost	Num_Bound	WaveClust	RegClust	NonCir
1	B-450	1	Cam	Mod	5500	14	9	7	9
2	B-454	1	Cam	Mod	7500	6	5	4	1
3	F-109	1	Cam	Mod	5325	11	8	5	4
4	F-114	1	Cam	Mod	8900	24	19	11	14
5	K-1457	1	Spring	Mod	2400	4	4	1	2
6	K-1474	2	Spring	Mod	3000	4	4	1	4
7	K-1481	1	Spring	Mod	4000	8	8	2	2
8	K-1482	1	Spring	Mod	4000	6	6	2	0
9	K-1483	1	Spring	Mod	4000	6	6	2	0
10	K-1487	1	Spring	Mod	2750	2	2	1	0
11	K-1492	2	Spring	Mod	1800	19	20	9	9
12	K-1493	1	Spring	Mod	1600	12	12	6	6
13	B-402	1	Spring	Mod	2000	6	5	4	1
14	B-407	1	Spring	Mod	1000	11	8	7	4
15	B-444	8	Spring	Mod	6189	4	3	3	2
16	B-447	1	Spring	Mod	2500	2	2	2	0
17	B-453	1	Spring	Mod	7000	19	13	9	12
18	B-457	4	Spring	Mod	7800	2	2	2	2
19	K-1448	1	Straight	Mod	2809	8	8	2	0
20	K-1450	1	Straight	Mod	2500	12	12	6	7
21	K-1449	2	Straight	Mod	3700	3	3	1	0
22	K-1451	1	Straight	Mod	3000	6	6	2	1
23	K-1464	8	Straight	Mod	1500	4	4	1	0
24	K-1465	1	Straight	Mod	1509	11	11	5	2

Figure 31 Minitab dataset

To create the master dataset we used three sources of data. The first source of data is the invoices of the molds that include cost. The second source was the print of the mold as shown in Figure 32. Using the mold print we were able to determine factors such as the number of cavities and the design of the mold. These factors from the mold print were used to manually enter these factors into our dataset. The third source of data is prints of the plastic parts themselves as shown in Figure 33. The part prints were used in the core research as outlined in our methodology. Both wavelet descriptors and regional descriptors were calculated automatically from the part print. A computer program written in Matlab extracted this data automatically.

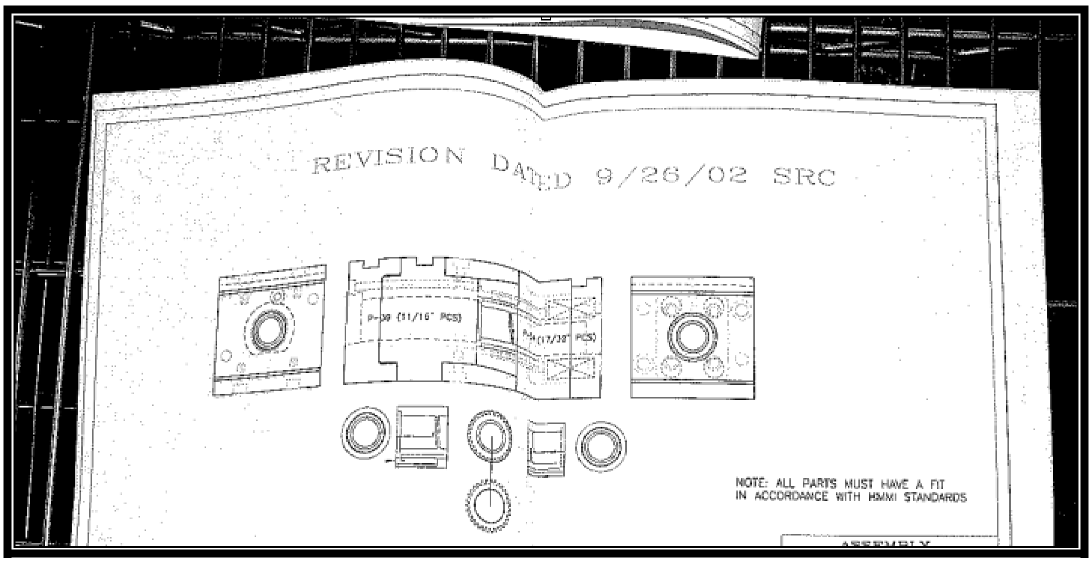

Figure 32 Mold drawing

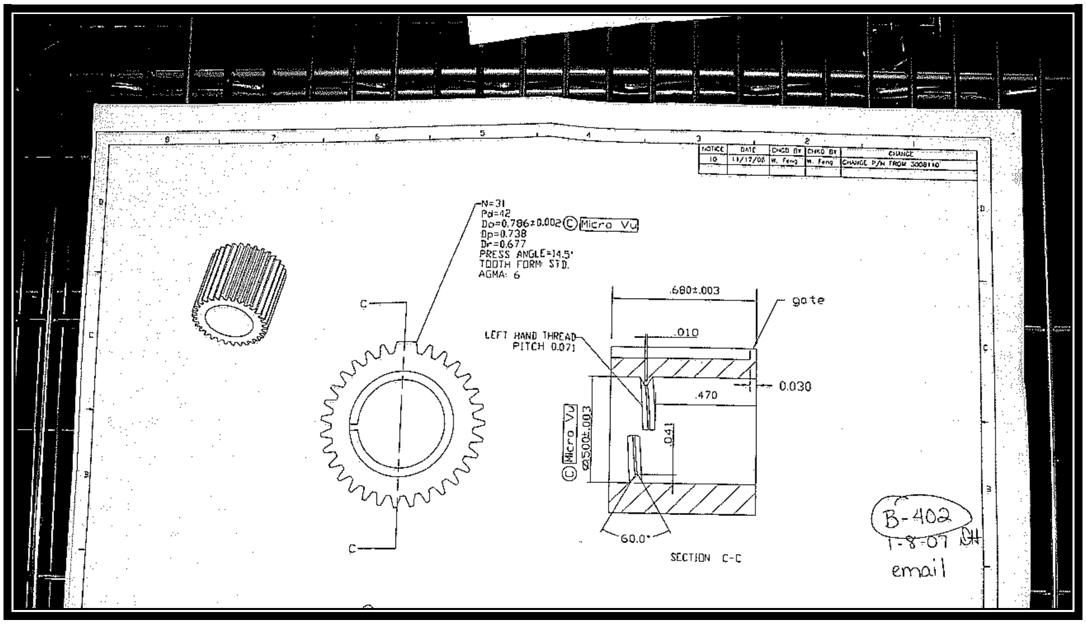

Figure 33 Part drawing

4.2.1 Non core research variables

The first four variables namely, *Name*, *Num_Cav*, *Design*, and *Type* were taken from the mold print. The next variable named *Cost* was taken from the invoice. The data from these first two datasets were calculated and entered manually and were not a part of our core research.

Name is the name of the mold and is the unique identifier for that mold within the company that supplied the data.

Num_Cav is the number of cavities in the mold. Intuitively, the more cavities a mold contains, the more labor and materials used. More labor and material should result in higher cost. We must be cautious, however. This variable is not in isolation due to the fact we have observed data.

Design is the mold design named straight, spring or cam. As mentioned in the methodology section these are three different engineering designs and are fundamentally different. Therefore they should serve as a partition for the dataset.

Type is the last data point taken from the mold print. It represents the system which was used to manufacture the mold either conventional (*cov*), master unit die (*mud*), or modular (*mod*). As mentioned in the methodology these are fundamentally different systems with a fundamental different cost structure.

Cost was the quoted and sale price. It was taken directly from the invoice to the customer. These prices were calculated based upon experience of the bidder. However, they may not reflect reality in all cases. Factors such as missed bids, intentional over or underbid, and factors not associated with the variables we record such as surface finish or tolerance may have been part of the bid.

4.2.2 Core research variables

The last four variables *Num_Bound*, *WaveClust*, *RegClust* and *NonCir* in Figure 31 were taken from the part print semi-automatically. They were a part of the core research so we would like to discuss this part of the methodology in greater detail.

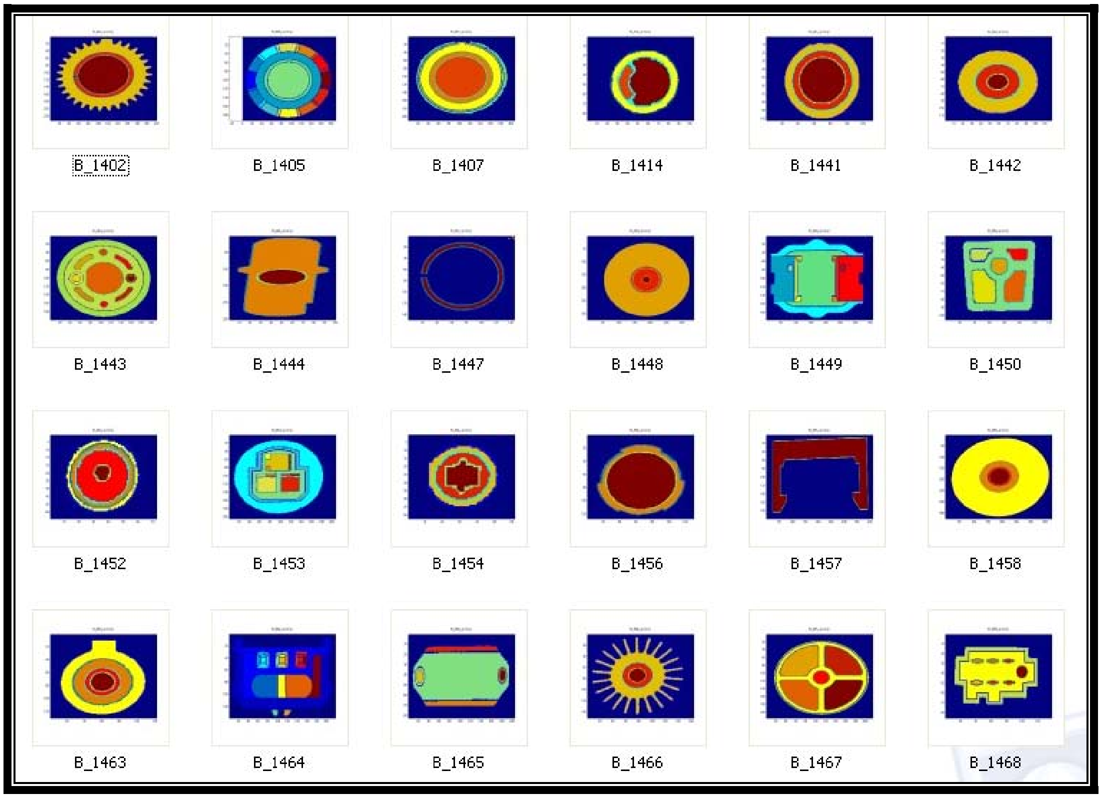

Figure 34 Panel one of parts

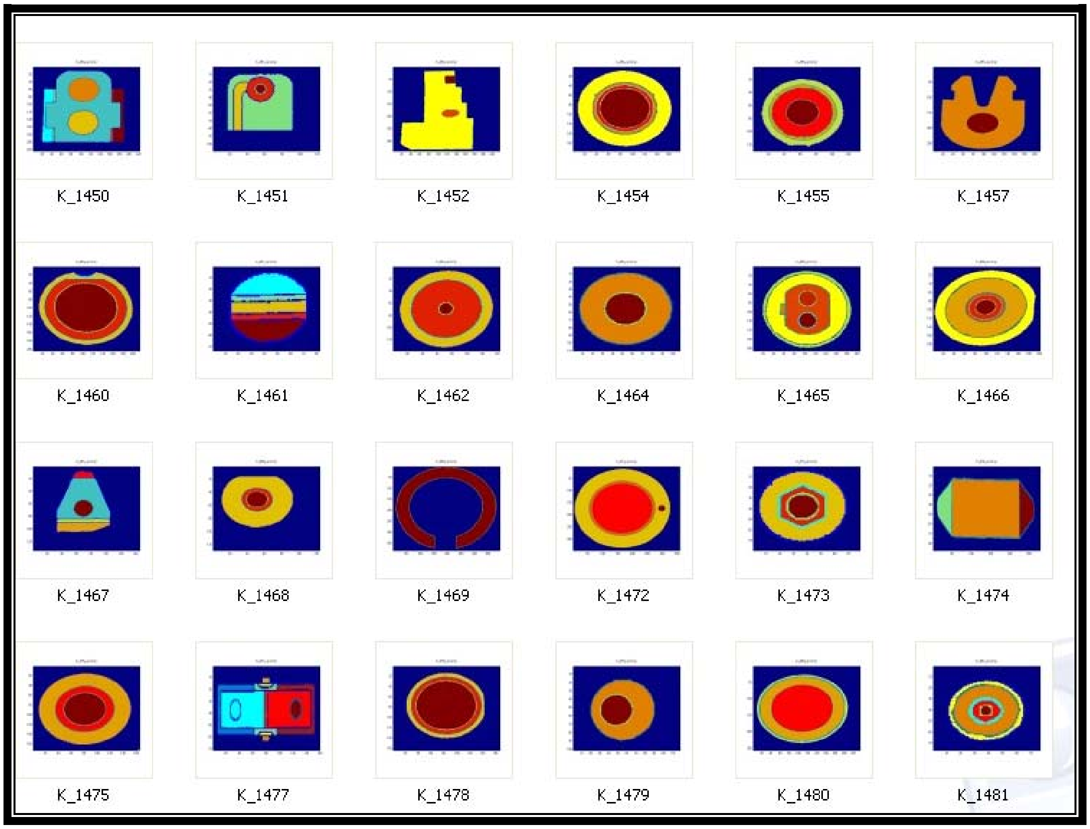

Figure 35 Panel two of parts

However before these variables can be calculated we must preprocess the data. Preprocessing such as removing lines not associated with the geometry of the part is not a part of our methodology. Preprocessing was done in order to test the methodology. In other words raw images of parts such as those displayed in Figure 33 are converted to those similar to Figure 36. Two panels of twenty- four converted parts are shown in Figure 34 and Figure 35. After preprocessing we can move to the semi-automatic data collection from the images.

4.2.2.1 Number of boundaries

The number of boundaries (*Num_Bound*) is the number of boundaries of the part image. The number of boundaries is related to the complexity of the part. For this work each boundary is considered to be associated with a feature on the part. Intuitively the more features the higher the resulting cost. In the example shown in Figure 36 the part has three distinct boundaries. These boundaries represent features on the part itself. The first boundary is the gear shaped outer boundary. The second boundary is internal and has a reversed letter c shape. The third boundary is also internal and circular with a tab located on the left side.

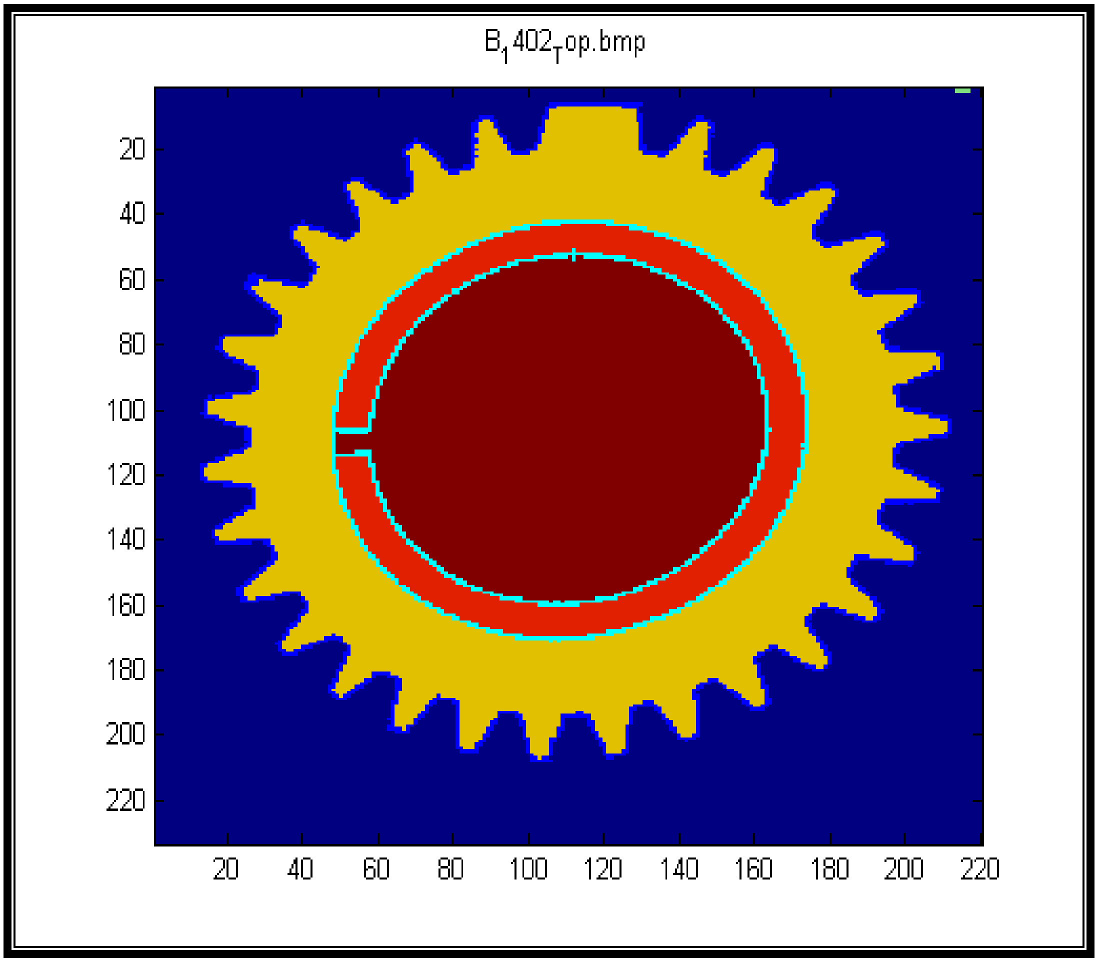

Figure 36 Part with three boundaries

4.2.2.2 Wavelet descriptor clustering

The wavelet cluster (*WaveClust*) is the total number of unique clusters for one part out of the twenty general shapes. In other words we defined twenty approximate shapes and clustered all boundaries for all parts into those twenty clusters. Then we counted the unique clusters for a given part.

This was an attempt to measure symmetry as more symmetric parts are assumed to have less complexity for a given number of boundaries. As an example, in Figure 31 the part in the first row named B-450 had exactly 14 boundaries. However some of these boundaries are in common clusters and have a similar shape. Therefore B-450 has 9 unique boundary shapes. A typical symmetric part is show in Figure 37. A typical non-symmetric part is shown in Figure 38.

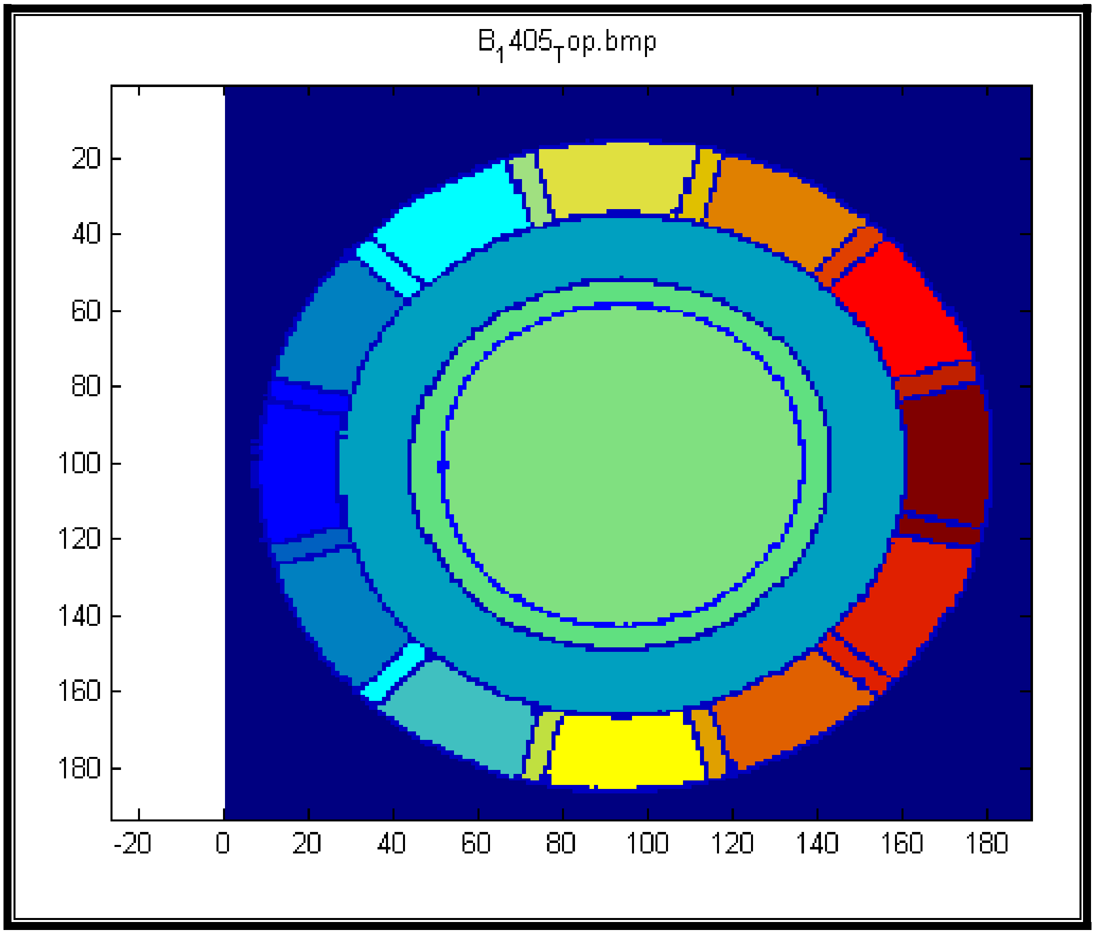

Figure 37 Symmetrical part

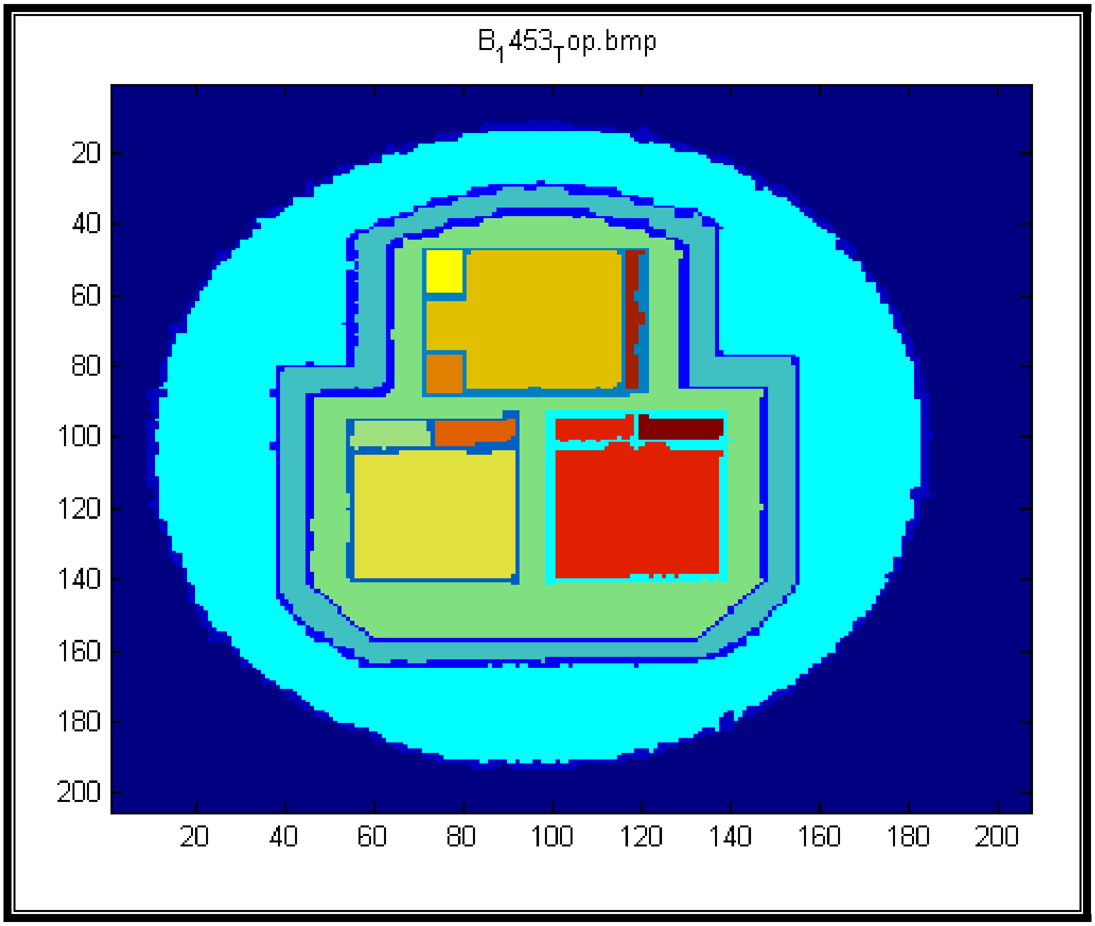

Figure 38 Non symmetrical part

4.2.2.3 Regional descriptor clustering

Regional cluster (*RegClust*) is different than the other variables in that it is a combination of several regional descriptors. Seven regional descriptors are used to compose this variable. The regional descriptors are *area*, *eccentricity*, *convex area*, *filled area*, *Euler number*, *extent*, and *solidity*. The primary reason we selected these regional descriptors was these are represented as a scalar and could be combined easily.

Area is the number of pixels in a region. Eccentricity is the ratio of the distance between the foci of the ellipse and its major axis. Convex area is the size of a convex polygon approximating the boundary shape. Filled area is the size of the bounding box. Euler number is the number of objects in a region minus the number of holes in those objects. Extent is the area of the boundary divided by the area of the bounding box for that region. Solidity is the area divided by the convex area.

The values for each individual regional descriptor were calculated for each boundary within an image. Those seven descriptors are clustered into ten approximate shapes for all boundaries for all parts. We then record the cluster number for each boundary. RegClust is defined as the total number of unique clusters for each part out of the possible ten clusters.

RegClust was an attempt to measure symmetry. The rationale was if a boundary was similar based upon all metrics then it is by definition similar and belongs to the same cluster. This was used to measure symmetry and indirectly complexity and cost. One example is shown in Figure 31. The first part B-450 had 14 boundaries but only 7 unique clusters based upon the regional descriptor clustering or RegClust.

One variable was calculated from a regional property, called eccentricity. Those boundaries with an eccentricity above 0.40 were highly correlated to non-circular (NonCir) boundaries. Intuitively non-circular boundaries are more difficult to machine and have a higher cost. This was a simple way to separate simple boundaries defined as circular boundaries from more complex non-circular boundaries.

Another variable washer, was defined as binary. If the number of circular boundaries was less than two we defined this variable as a washer. If not it was labeled as a more complex shape. This variable played a role in the regression model for mold type modular and design straight. Most molds for washers are design type straight. This variable is represented as a constant in the regression model.

Section 4.3 Analysis

4.3.1 Data partitions

Since we may be dealing with a mixed dataset with three or more distinct homogeneous datasets, we want to investigate good ways to partition the data. Partitioning the data has two immediate consequences. First, the variance of residuals is likely to be more constant, because we are dealing with data with a smaller range. The second consequence is that we have an easier time investigating outliers when placed within their prospective partitions.

The first partition was by mold type. Clearly from the box plot these are three separate distributions with different cost structures as shown in Figure 39.

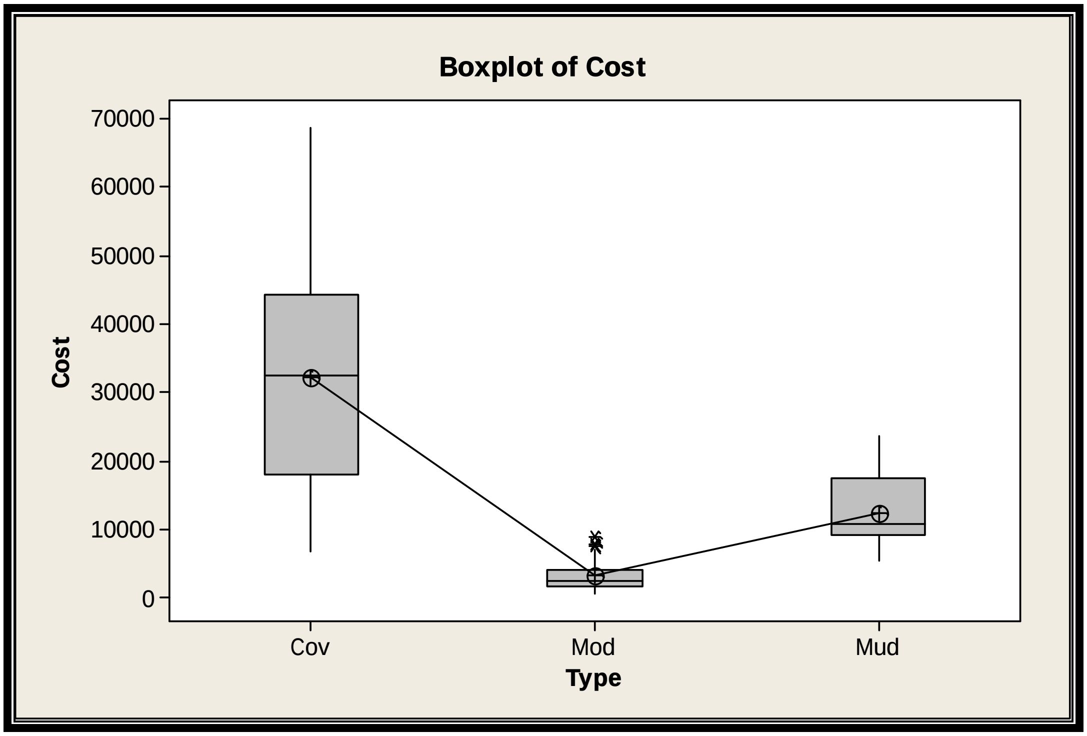

Figure 39 Box plot cost versus mold type

We confirm our intuition from the box plot with ANOVA and Kruskal-Wallis tests as shown in Figure 40. The low p-values for these tests confirm that indeed mold type can be used to partition the dataset.

The data is partitioned, and we use only the modular partition for further tests. Later we partition the data again by design and build two regression models on those sub partitions. We build one regression model on mold type modular and design straight. We build another regression model on mold type modular and design spring.

```
One-way ANOVA: Cost versus Type

Source  DF           SS          MS      F      P
Type     2  10076838183  5038419091  80.14  0.000
Error   80   5029807889    62872599
Total   82  15106646072

S = 7929   R-Sq = 66.70%   R-Sq(adj) = 65.87%


                           Individual 95% CIs For Mean Based on
                           Pooled StDev
Level   N   Mean  StDev  ---------+---------+---------+---------+
Cov    16  32029  17186                            (---*---)
Mod    49   3215   1933  (-*-)
Mud    18  12413   4970          (--*---)
                         ---------+---------+---------+---------+
                             10000     20000     30000     40000

Pooled StDev = 7929


Kruskal-Wallis Test: Cost versus Type

Kruskal-Wallis Test on Cost

Type       N  Median  Ave Rank      Z
Cov       16   32313      72.3   5.59
Mod       49    2500      25.5  -7.49
Mud       18   10623      60.0   3.59
Overall   83              42.0

H = 58.31  DF = 2  P = 0.000
H = 58.34  DF = 2  P = 0.000  (adjusted for ties)
```

Figure 40 Statistics cost versus mold type

When we fit a regression model on only modular molds without design being a variable, we have a classic case of non-constant variance. This is detected by the top right graph with fitted value versus residual shown in Figure 41. Therefore although the data is more homogeneous, we still require at least one more partition before building the regression models.

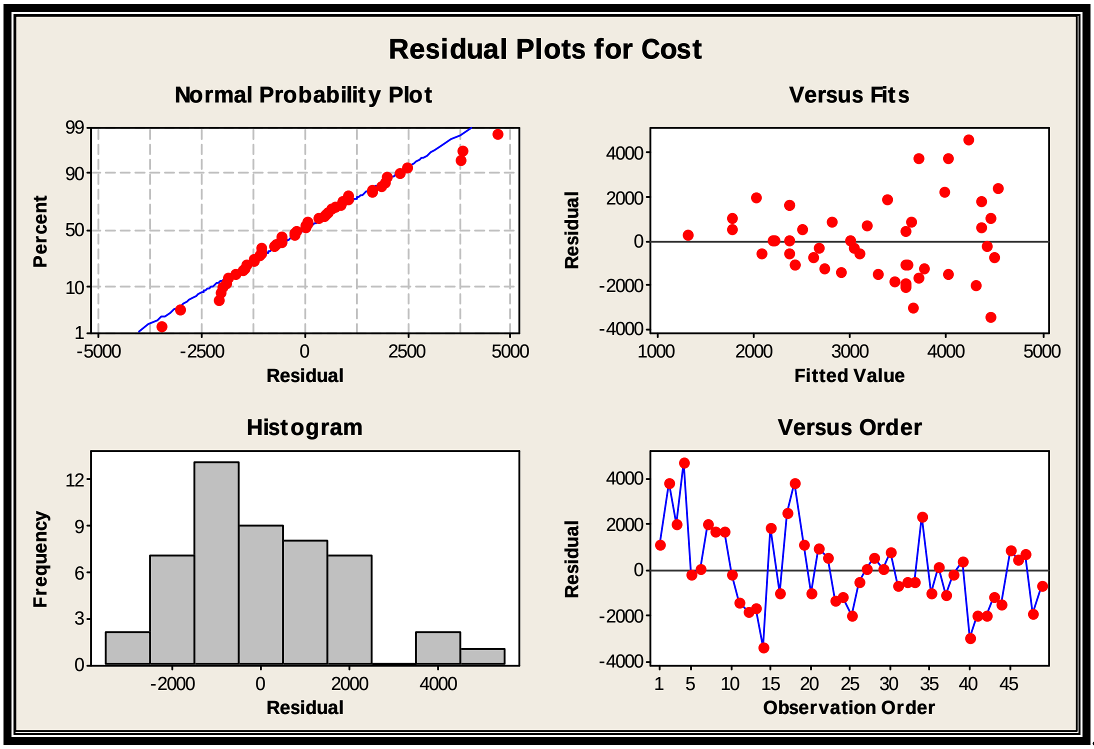

Figure 41 Non constant variance

To determine whether or not to partition on design we start with a box plot shown in Figure 42. It is very clear from the box plot that mold design cam has a very different cost structure than either straight or spring.

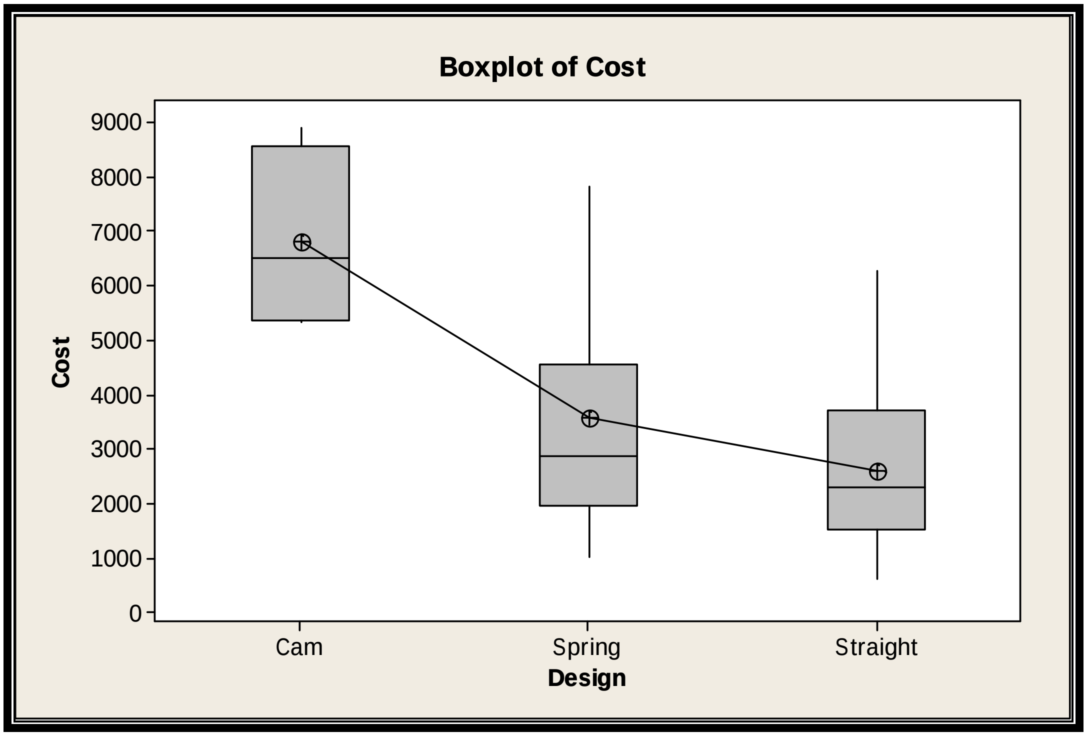

Figure 42 Box plot cost versus mold design

The Tukey multiple comparisons tests in Figure 43 clearly indicates that design cam is different from either spring or straight since the confidence level does not include zero.

```
Tukey 95% Simultaneous Confidence Intervals
All Pairwise Comparisons among Levels of Design

Individual confidence level = 98.07%


Design = Cam subtracted from:

Design    Lower  Center  Upper  -----+---------+---------+---------+----
Spring    -5395   -3232  -1069     (--------*--------)
Straight  -6245   -4217  -2190  (-------*-------)
                                -----+---------+---------+---------+----
                                 -5000     -2500         0      2500


Design = Spring subtracted from:

Design    Lower  Center  Upper  -----+---------+---------+---------+----
Straight  -2214    -985    243                 (----*----)
                                -----+---------+---------+---------+----
                                 -5000     -2500         0      2500
```

Figure 43 Statistics cost versus mold design

The difference between design spring or straight is less clear. Thus we perform a t-test to determine whether or not these are indeed good partitions. The p-value of 0.12 is an indication that the populations are indeed different as shown in Figure 44.

Two-Sample T-Test and CI: Straight, Spring

```
Two-sample T for Straight vs Spring

           N  Mean  StDev  SE Mean
Straight  31  2589   1270      228
Spring    14  3574   2088      558


Difference = mu (Straight) - mu (Spring)
Estimate for difference:  -985
95% CI for difference:  (-2257, 286)
T-Test of difference = 0 (vs not =):
T-Value = -1.63  P-Value = 0.120  DF = 17
```

Figure 44 Design straight versus spring

4.3.2 Models of the partitioned datasets.

Here we model two of the partitioned datasets and observe the results. The first partition is of mold type modular and mold design straight. The second partitioned dataset is mold type modular and mold design spring. Mold type modular and mold design cam were not modeled because our dataset only include four observations that met this criterion.

4.3.3 Mold type modular and design type straight.

This dataset had 31 observations and was considered to be a large enough dataset on which to build regression models and make inferences. The first step in this process was the removal of outliers.

4.3.3.1 Outliers

Our initial dataset included 31 observations, which were of type modular and design straight. Of those 31 observations we identified 6 as outliers. These 6 observations tended to have a high cost and tended to have an unusually high number of clusters for both WaveClust and RegClust. We believe this is a problem related to modeling a dataset with a high variance.

4.3.3.2 Variable selection

One variable, number of cavities (Num_Cav), was removed from the model because of a high p-value and did not appear to contribute to the predictive ability of the model. The quadratic version of this same variable was found to be significant and remained in the model.

4.3.3.3 The final model and explanations of the modeling process

Our final model includes 25 observations with approximately 68 percent of the variance in the data being explained by the model. This is shown in Figure 45. We believe this is a good representative model. However we would like to confirm our results with a larger dataset.

```
The regression equation is
Cost = 1981 + 304 WaveClust - 744 RegClust + 1152 NonCir - 13.2 cav_sq
       + 111 reg_sq - 103 cir_sq - 1707 washer - 25.7 wave_sq


Predictor      Coef  SE Coef      T      P
Constant     1981.4    907.2   2.18  0.044
WaveClust     303.6    248.3   1.22  0.239
RegClust     -744.4    336.0  -2.22  0.042
NonCir       1151.9    351.7   3.27  0.005
cav_sq      -13.204    9.514  -1.39  0.184
reg_sq       111.01    41.93   2.65  0.018
cir_sq      -103.00    36.33  -2.84  0.012
washer      -1707.4    724.0  -2.36  0.031
wave_sq      -25.70    17.70  -1.45  0.166


S = 620.958   R-Sq = 67.9%   R-Sq(adj) = 51.8%


Analysis of Variance

Source          DF        SS       MS     F      P
Regression       8  13029227  1628653  4.22  0.007
Residual Error  16   6169421   385589
Total           24  19198648


Source     DF   Seq SS
WaveClust   1   465320
RegClust    1  3825877
NonCir      1  1396592
cav_sq      1  1260823
reg_sq      1  1667941
cir_sq      1  1407341
washer      1  2192455
wave_sq     1   812877
```

Figure 45 Mold type modular design straight

4.3.3.4 Non-normality

From the normality plot and the histogram of residuals we do have some non-normality of the residuals. This is shown in Figure 46. We view this as not too serious given the nature of the problem and the limited dataset.

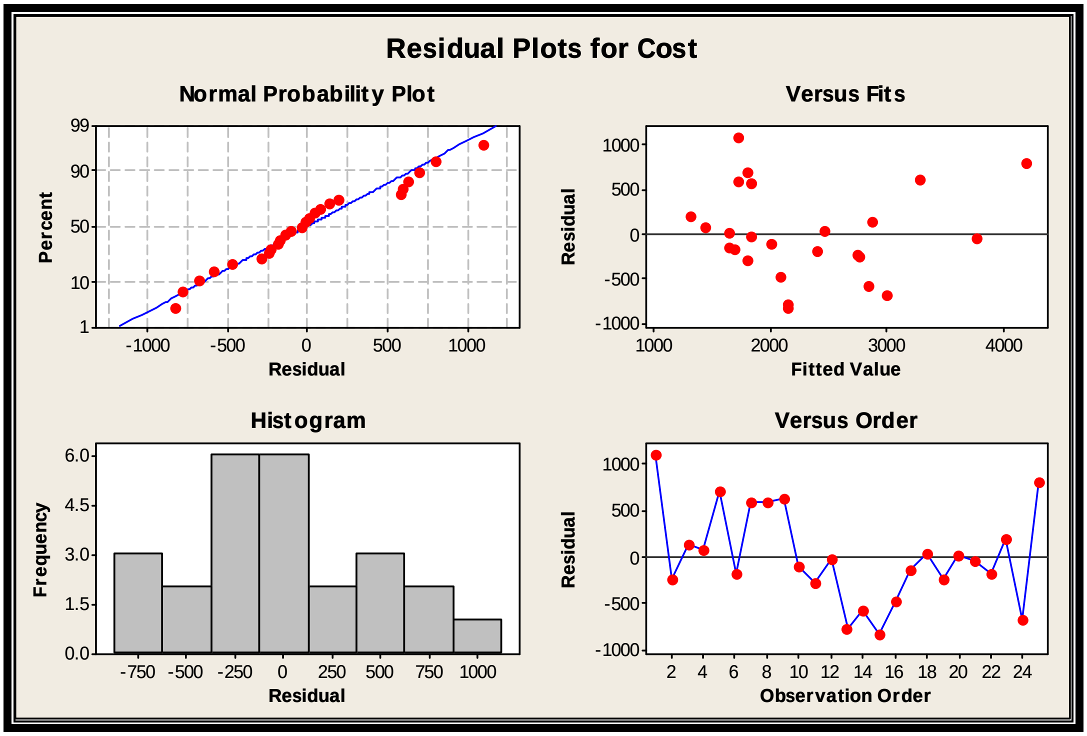

Figure 46 Residual plot mold type modular design straight

4.3.4 Mold type modular and mold design spring

Our dataset included 14 observations that were of type modular and design spring. While this dataset is a bit small we do believe it is possible to build models and infer some results. We start by detecting outliers.

4.3.4.1 Outliers

There was one outlier found in this dataset. This was a true outlier as it was found after investigation that this mold was incorrectly coded. It was an unusual design type called double spring load.

One observation, namely observation 12, had high leverage but was left in the model for a few reasons. The first reason was the concern that we are starting with limited data so we should be careful as to what we consider an outlier. The second reason was that although the data point had high leverage its removal only resulted in more data points that had high leverage. Thirdly the predictive ability for the dataset with or without this observation was very good. Therefore leverage may not be a good outlier detection method on a model with such a low sum of residuals.

4.3.4.2 Variable selection

Two variables were excluded from the final model. The first was reg_sq and was the second order version of the original variable RegClust. This variable was found to have a high p-value and did not contribute to the predictive ability of the model. The other variable the number of boundaries (*Num_Bound*) was found to be highly correlated with RegClust and WaveClust and therefore were removed from the model.

4.3.4.3 The final model and explanations of the modeling process

The model for spring was remarkably good as rated by low residual error as shown in Figure 47. We approach such results with cautious optimism. However, the results while significant should be backed up by a larger dataset in future results.

```
The regression equation is
Cost = 1737 + 731 Num_Cav + 562 WaveClust - 653 RegClust - 701 NonCir
       + 20.8 num_sq - 33.5 wave_sq + 67.2 cir_sq


Predictor      Coef  SE Coef      T      P
Constant     1737.2    315.5   5.51  0.003
Num_Cav      730.77    44.48  16.43  0.000
WaveClust    562.31    83.28   6.75  0.001
RegClust    -652.68    95.57  -6.83  0.001
NonCir      -701.27    90.29  -7.77  0.001
num_sq       20.750    8.450   2.46  0.058
wave_sq     -33.477    3.499  -9.57  0.000
cir_sq        67.15    15.98   4.20  0.008


S = 253.209   R-Sq = 99.1%   R-Sq(adj) = 97.9%


Analysis of Variance

Source          DF        SS       MS      F      P
Regression       7  37114637  5302091  82.70  0.000
Residual Error   5    320574    64115
Total           12  37435212


Source     DF    Seq SS
Num_Cav     1   7962559
WaveClust   1      2420
RegClust    1     43392
NonCir      1   6267487
num_sq      1   9733798
wave_sq     1  11973185
cir_sq      1   1131796


Unusual Observations

Obs  Num_Cav    Cost     Fit  SE Fit  Residual  St Resid
 12     1.00  7000.0  6991.7   252.8       8.3      0.57 X
```

Figure 47 Mold type modular design type spring

4.3.4.4 Normality

If we observe the normality plot in Figure 48, it appears that non-normality is present in this model. However we do not consider the effects too severe for two reasons. The first reason is that we have limited data. We do not know for certain whether or not the non-normality would be present if given a larger dataset. The second reason is that we do not expect data such as this to be perfectly homogeneous. Our goal was merely to make the data more homogeneous in order to build accurate and representative models. We believe it was achieved through our methodology.

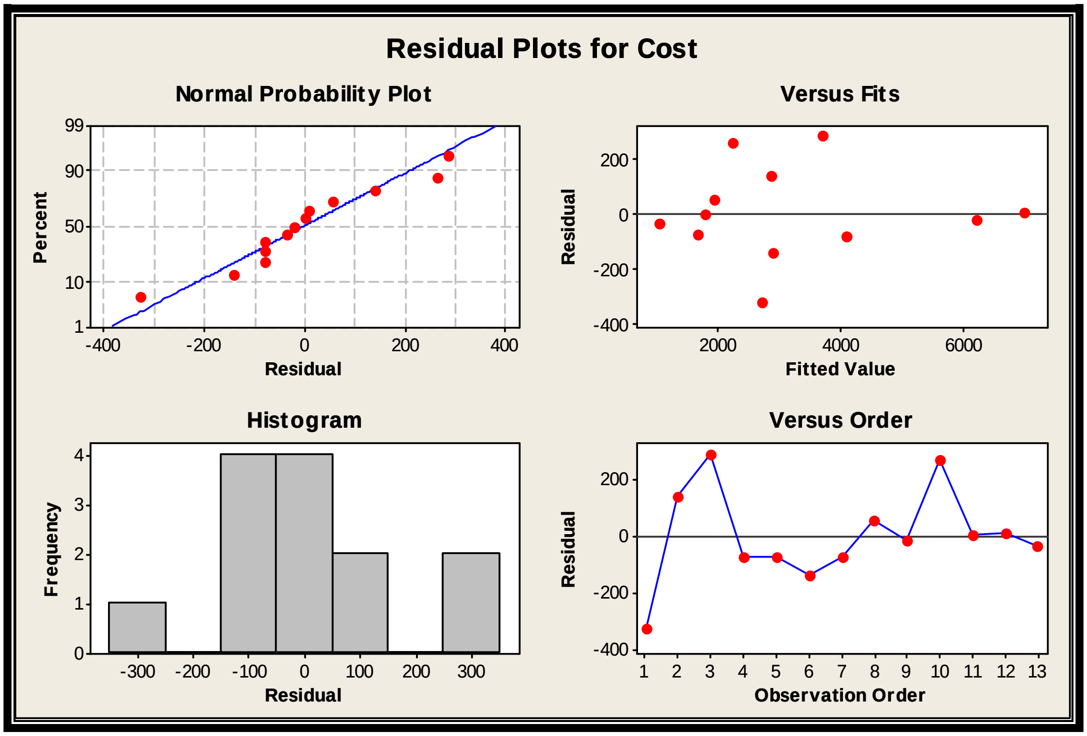

Figure 48 Residual mold type modular mold design spring

Section 4.4 Cross fold validation

Here we validate the models using cross validation. Due to the small sample sizes we chose to use the leave-one-out strategy. The leave-one-out strategy is to take one observation out of the model and use the rest for modeling. The one observation left out is used for validation. As a concrete example we use the straight dataset. The straight dataset included 25 observations once outliers were taken out. Therefore we randomly select one observation for validation to leave out. The other 24 observations are used to build the model. At the next iteration another observation is chosen. It is then used for validation. We still have 24 observations because the one left out of the previous iteration is replaced and used for modeling. In our case we performed the leave-one-out cross validation ten times and recorded the results for both the design straight and design spring models.

First we define which metrics we choose for validation and then discuss the results. Using this methodology we calculated several metrics from the regression model for validation as shown in Figure 49 and Figure 50. Reading from left to right the first metric is the prediction (*Predict*) based upon the regression model. The following columns are the ninety-five-percentage lower (*Lower*) and upper (*Upper*) prediction intervals, the absolute percentage error (*APE*), the prediction interval length (*PI*), the name of the hold out (*Name*) and the actual cost (*Cost*).

In the last row in Figure 49 and Figure 50 we see the mean absolute percentage error (*MAPE*). It is defined as the absolute value between the predicted value and the actual value divided by the actual value. It is a common metric to determine the accuracy of the model and can be found in Upton and Cook [126].

For the straight model we see that the MAPE was twenty one percent as shown in Figure 49. This means that on average the model is off by twenty one percent. The spring model was more accurate with an average MAPE of twelve percent as shown in Figure 50.

This is remarkable considering several factors. The first factor is the range of the data with a high of $60,000 to a low of $600. The second factor is that these models were built based upon bids not actual costs. Even if we had actual cost, it is well known in a job shop environment that some jobs go well and others do not, which results in noisy data.

Straight						
Predict	Lower	Upper	APE	PI	Name	Cost
1523	222	2824	45.78%	2602	K-1448	2809
2868	1323	4414	4.40%	3091	K-1451	3000
1388	-715	3490	8.02%	4205	K-1465	1509
2429	754	4103	10.41%	3349	K-1472	2200
1756	356	3157	26.83%	2801	K-1458	2400
2146	304	3988	12.95%	3684	K-1473	1900
3066	1560	4571	36.27%	3011	K-1480	2250
1702	62	3342	13.47%	3280	B-441	1500
2886	1291	4481	15.44%	3190	B-452	2500
3255	1618	4893	41.52%	3275	B-448	2300
MAPE			21.51%			

Figure 49 Validation of straight model

Spring						
Predict	Lower	Upper	APE	PI	Name	Cost
2873	2194	3554	19.71%	1360	K-1457	2400
2637	1471	3803	12.10%	2332	K-1474	3000
3349	2586	4113	16.28%	1527	K-1481	4000
4122	3156	4700	3.05%	1544	K-1482	4000
4123	3157	5088	3.08%	1931	K-1483	4000
2972	2028	3916	8.07%	1888	K-1487	2750
1927	370	3484	20.44%	3114	K-1493	1600
1916	957	2875	4.20%	1918	B-402	2000
8088	903	15273	30.68%	14370	B-444	6189
1735	-4062	7537	3.61%	11599	K-1492	1800
MAPE			12.12%			

Figure 50 Validation of spring model

CHAPTER FIVE: CONCLUSIONS

From this research we have shown that a semiautomatic cost estimation system is possible for estimating the cost of injection molds. However we believe it would be difficult to fully automate this process and believe that an experienced bidder should always remain in the loop.

Our main contributions are as follows. First, we have shown that we can extract features automatically from images and relate them to the cost of the mold. This had not been done previously. Using a unique feature vector we combined knowledge of the mold type, mold design, and geometry into a central dataset. The central dataset was later used in the regression models. Using simple descriptors, wavelet descriptors, and regional descriptors, we related geometry to complexity and cost. Second, we developed our own unique variables to relate images to cost. Specifically we developed three new variables. The first variable is the number of non-circular boundaries (NonCir). This variable was shown to relate to cost. The next two variables were a combination of the number of boundaries and symmetry. Specifically they are the number of wavelet clusters for a part (WaveClust) and the number of regional clusters (RegClust). Third, we showed that images could be used as the common data format for the part complexity portion of mold cost estimation. Previous researchers could not use all data formats. Fourth, we showed that partitioning mold cost data before modeling could be useful. This is at the heart of our combination analogy and mathematical approach. We partitioned the data into homogeneous datasets or the analogy portion. Then we built regression models or the mathematical portion. We showed that a combined analogy and mathematical approach is in this domain. Fifth, we provide estimates and prediction intervals on relevant molds only. This was to provide a foundation for later risk management efforts.

Our work does have several limitations. The first limitation is that a human subject matter expert is still needed due to those factors not captured in our model. From the beginning this effort was conceived of as an assistant to an experienced bidder. Our concept is to provide a ballpark cost estimate that could serve two purposes. One use is to provide a double check on the bidders estimate. The other use is a guide to assist in early design decisions. The second limitation is that all data must be converted to the neutral image format before use. Depending on the current data format of the mold and part data this could take significant time. The third limitation is that data preprocessing such as removal of lines not associated with the geometry of the part must be done prior to our methodology. For our preliminary experiments non-geometric line removal was done manually. This manual process took approximately fifteen minutes per part. We believe that non-geometric line removal could be done automatically but was not a focus for this project.

CHAPTER SIX: FUTURE EXTENSIONS

This work has been conceived of as an injection molding cost system. However this process could be extended to other net shape manufacturing processes. These would include casting, blow molding, transfer molding, compression molding, stamping, etc. The methods provided here form a basis for estimating the tooling cost of many processes. In this way it could be a possible backbone to a blackboard system for many parts within an assembly.

A second extension could be the use of hidden lines for better image processing and granularity. For this work only those features visible from the outside of the part were used. Internal features such as threads or undercuts may be an extension of this work in order to bid a wider variety of molds.

A third extension could be to design a wavelet for a specific application. In this work we used a standard wavelet. However, we could see applications where this may not be adequate. For example, a custom wavelet may allow for better compression or detection of certain features which are relevant to cost estimation in a specific domain.

A fourth extension would be to move beyond wavelets to wavelet packets or other representations. These methods include ridgelets, ridgelet packets, brushlets, countourlets, and various other methods. These methods are mentioned in the book by Welland [127].

A fifth extension could be to use separable 2D wavelets instead of the 1D wavelets we used in our methodology. It may be possible using 2D wavelets directly to detect features. For example with 2D wavelets it is be possible to detect a square with the horizontal and vertical coefficients being significant and the diagonal wavelet coefficients being not significant. We would expect a circle to have significant coefficients on all three directions vertical, horizontal, and diagonal. It may be possible to derive variables from this information to separate shapes.

LIST OF REFERENCES

1. Hunley, K., *IMM's Benchmarking Report: Fourth quarter 2002, data group 3 of 3.* Injection Molding Magazine, 2003.
2. Christman, A. and J. Naysmith, *Trends In CAD/CAM For Mold Makers* 2001.
3. Fallbohmer, P., et al., *Survey of the die and mold manufacturing industry - practices in Germany, Japan, and the United States.* Journal of Materials Processing Technology, 1996. **59**(1-2): p. 158-168.
4. *Annual Census of Manufacture.* 2005.
5. *Census Industry Series Report, All Other Plastic Product (NAICS code 326199).* 2002.
6. Shehab, E. and H. Abdalla, *An Intelligent Knowledge-Based System for Product Cost Modelling.* The International Journal of Advanced Manufacturing Technology, 2002. **V19**(1): p. 49-65.
7. Nagahanumaiah, N.P. Mukherjee, and B. Ravi, *An integrated framework for die and mold cost estimation using design features and tooling parameters.* International Journal of Advanced Manufacturing Technology, 2005. **26**(9/10): p. 1138-1149.
8. Naitove, M., *Working Harder, Working Smarter.* Plastics Technology, 2003. **Vol. 49**(Issue 1): p. p9.
9. Gonzalez, R.C., *Digital Image Processing using Matlab.* 2004.
10. Kluz, J., *MOLDMAKING and DIE CAST DIES for Metalworking Trainees*. 1981.
11. Poli, C., Escudero, Juan, Fernandez, Ricardo. . . , . Retrieved August 6, 2007, from ABI/INFORM Global database. (Document ID: 594364). *How Part Design Affects Injection-Molding Tool Costs.* Machine Design, 1988. **60(28)**(November): p. 101-105.
12. Kwong, C.K. and G.F. Smith, *A computational system for process design of injection moulding: Combining a blackboard-based expert system and a case-based reasoning approach.* The International Journal of Advanced Manufacturing Technology, 1998. **14**(5): p. 350-357.
13. El-Mehalawi, M.E.-S., *A geometric similarity case-based reasoning system for cost estimation in net-shape manufacturing*. 1999, The Ohio State University: United States -- Ohio.
14. Wong, T.S., *Evaluation of a software package to estimate cost of an injection mold*. 1999, University of Massachusetts Lowell: United States -- Massachusetts.
15. Wang, H., X.Y. Ruan, and X.H. Zhou, *Research on Injection Mould Intelligent Cost Estimation System and Key Technologies.* The International Journal of Advanced Manufacturing Technology, 2002. **V21**(3): p. 215-222.
16. Bergmann, R., *Developing Industrial Case-Based Reasoning Applications.* 2003.
17. Raviwongse, R. and V. Allada, *Artificial neural network based model for computation of injection mould complexity.* The International Journal of Advanced Manufacturing Technology, 1997. **13**(8): p. 577-586.

18. Merino, D.W., *A statistical based cost estimation system for plastic molded parts: A design for cost approach*. 1998, Stevens Institute of Technology: United States -- New Jersey.
19. Shehab, E.M. and H.S. Abdalla, *A design to cost system for innovative product development.* Proceedings of the Institution of Mechanical Engineers -- Part B -- Engineering Manufacture, 2002. **216**(7): p. 999-1019.
20. Chan, S.F., C.K. Law, and K.K. Chan, *Computerised price quoting system for injection mould manufacture.* Journal of Materials Processing Technology, 2003. **139**(1-3): p. 212-218.
21. Sapene, C., *Cost Analysis of Plastic Injection Molds.* 2006.
22. Thibault, A., A. Siadat, and P. Martin. *A Framework For Using A Case Based Reasoning System Applied To Cost Estimation*. 2006.
23. H'Mida, F., P. Martin, and F. Vernadat, *Cost estimation in mechanical production: The Cost Entity approach applied to integrated product engineering.* International Journal of Production Economics, 2006. **103**(1): p. 17-35.
24. Duverlie, P. and J.M. Castelain, *Cost Estimation During Design Step: Parametric Method versus Case Based Reasoning Method.* The International Journal of Advanced Manufacturing Technology, 1999. **15**(12): p. 895-906.
25. Martin Shepperd, C.S., Barbara Kitchenham *Effort estimation using analogy* IEEE Computer Society, 1996. **Proceedings of the 18th international conference on Software engineering** (Proceedings of the 18th international conference on Software engineering): p. 170-178
26. Shepperd, M. and C. Schofield, *Estimating software project effort using analogies.* Software Engineering, IEEE Transactions on, 1997. **23**(11): p. 736-743.
27. Latecki, J. and R. Lakaemper. *The official home page of the Shape Similarity Project at Temple University, Philadelphia, Pennsylvania.* 2008 [cited; Available from: http://knight.cis.temple.edu/~shape/.
28. Chazelle, B., et al. *Princeton Shape Retrieval and Analysis Group*. 2008 [cited; Available from: http://www.cs.princeton.edu/gfx/proj/shape/.
29. Ramani, K. and C. Hoffman. *Purdue Research and Education Center for Information Science in Engineering (PRECISE)*. 2007 [cited; Available from: https://engineering.purdue.edu/PRECISE.
30. Belongie, S., J. Malik, and J. Puzicha, *Shape matching and object recognition using shape contexts.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. **24**(4): p. 509-522.
31. Loncaric, S., *A survey of shape analysis techniques.* Pattern Recognition, 1998. **31**(8): p. 983-1001.
32. Iyer, N., et al., *Three-dimensional shape searching: state-of-the-art review and future trends.* Computer-Aided Design, 2005. **37**(5): p. 509-530.
33. Tangelder, J.W.H. and R.C. Veltkamp. *A survey of content based 3D shape retrieval methods*. 2004.
34. Cardone, A., S.K. Gupta, and M. Karnik, *A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications.* Journal of Computing and Information Science in Engineering, 2003. **3**(2): p. 109-118.

35. Bustos, B., et al., *Feature-Based Similarity Search in 3D Object Databases.* ACM Computing Surveys, 2005. **37**(4): p. 345-387.
36. Duda, R.O. and P.E. Hart, *Pattern Classification and Scene Analysis.* 1973.
37. Zomorodian, A.J., *Topology for Computing.* 2005.
38. Henle, M., *A Combinational Introduction to Topology.* 1994.
39. Fagin, R. and L. Stockmeyer, *Relaxing the Triangle Inequality in Pattern Matching.* International Journal of Computer Vision, 1998. **30**(3): p. 219-231.
40. Vleugels, J. and R.C. Veltkamp, *Efficient image retrieval through vantage objects.* Pattern Recognition, 2002. **35**(1): p. 69-80.
41. Dunn, F. and I. Parberry, *3D Math Primer for Graphics and Game Development.* 2002.
42. Kulkarni, A.D., *Computer Vision and Fuzzy-Neural Systems.* 2001.
43. Bribiesca, E. and A. Guzman, *How to describe pure form and how to measure differences in shapes using shape numbers.* Pattern Recognition, 1980. **12**(2): p. 101-112.
44. Hu, M.K., *Visual Pattern Recognition by Moment Invariants.* IRE Trans. Inform. Theory, 1962. **IT-8**(February): p. 179-187.
45. Hoffmann, C., *Geometric and Solid Modeling.* 1989.
46. Adams, L., *High Performance CAD Graphics in C.* 1989.
47. Celebi, M.E. and Y.A. Aslandogan. *A comparative study of three moment-based shape descriptors*. in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*. 2005.
48. Funkhouser, T., et al., *A search engine for 3D models* ACM Trans. Graph., 2003. **22**(1): p. 83-105.
49. Novotni, M. and R. Klein. *A geometric approach to 3D object comparison*. in *Shape Modeling and Applications, SMI 2001 International Conference on.* 2001.
50. Sanchez-Cruz, H. and E. Bribiesca, *A method of optimum transformation of 3D objects used as a measure of shape dissimilarity*. Image and Vision Computing, 2003. **21**(12): p. 1027-1036.
51. Cicirello, V.A. and W.C. Regli, *An approach to a feature-based comparison of solid models of machined parts.* AI EDAM, 2003. **16**(05): p. 385-399.
52. JungHyun, H., M. Pratt, and W.C. Regli, *Manufacturing feature recognition from solid models: a status report.* Robotics and Automation, IEEE Transactions on, 2000. **16**(6): p. 782-796.
53. Burbidge, J.L., *The first step in planning group technology*. International Journal of Production Economics, 1996. **43**(2-3): p. 261-266.
54. Gao, W., et al., *Multiresolutional similarity assessment and retrieval of solid models based on DBMS.* Computer-Aided Design, 2006. **38**(9): p. 985-1001.
55. Biasotti, S., et al., *Sub-part correspondence by structural descriptors of 3D shapes.* Computer-Aided Design, 2006. **38**(9): p. 1002-1019.
56. Bespalov, D., W.C. Regli, and A. Shokoufandeh, *Local feature extraction and matching partial objects.* Computer-Aided Design, 2006. **38**(9): p. 1020-1037.
57. Steiner, D. and A. Fischer, *Finding and defining the generators of genus-n objects for constructing topological and cut graphs.* The Visual Computer, 2004. **20**(4): p. 266-278.

58. El-Mehalawi, M. and R. Allen Miller, *A database system of mechanical components based on geometric and topological similarity. Part I: representation*. Computer-Aided Design, 2003. **35**(1): p. 83-94.
59. El-Mehalawi, M. and R. Allen Miller, *A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment.* Computer-Aided Design, 2003. **35**(1): p. 95-105.
60. Regli, W.C. and V.A. Cicirello, *Managing digital libraries for computer-aided design.* Computer-Aided Design, 2000. **32**(2): p. 119-132.
61. Rao, C.R. and S. Suryawanshi, *Statistical analysis of shape of objects based on landmark data.* Proceedings of the National Academy of Sciences, 1996. **93**(22): p. 12132-12136.
62. Dong, K. and P. Rae-Hong, *Classification of 3D objects with curved surfaces based on cross-entropy between shape histograms.* Electronics Letters, 2000. **36**(19): p. 1614-1615.
63. Osada, R., et al., *Shape distributions.* ACM Trans. Graph., 2002. **21**: p. 807-832.
64. Cardone, A.P.D.d., , College Park, United States -- Maryland. , *A feature-based shape similarity assessment framework*, in *Department of Mechanical Engineering*. 2005, University of Maryland: College Park, Maryland. p. 240.
65. Horn, B.K.P., *Extended Gaussian images.* Proceedings of the IEEE, 1984. **72**(12): p. 1671-1686.
66. Zouaki, H., *Representation and geometric computation using the extended Gaussian image.* Pattern Recognition Letters, 2003. **24**(9-10): p. 1489-1501.
67. Hebert, M., K. Ikeuchi, and H. Delingette, *A spherical representation for recognition of free-form surfaces.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1995. **17**(7): p. 681-690.
68. Tuzikov, A.V., J.B.T.M. Roerdink, and H.J.A.M. Heijmans, *Similarity measures for convex polyhedra based on Minkowski addition.* Pattern Recognition, 2000. **33**(6): p. 979-995.
69. Tangelder, J.W.H. and R.C. Veltkamp. *Polyhedral model retrieval using weighted point sets*. in *Shape Modeling International, 2003*. 2003.
70. Cyr, C.M. and B.B. Kimia, *A Similarity-Based Aspect-Graph Approach to 3D Object Recognition.* International Journal of Computer Vision, 2004. **57**(1): p. 5-22.
71. Wolfson, H.J. and I. Rigoutsos, *Geometric hashing: an overview.* Computational Science and Engineering, IEEE [see also Computing in Science & Engineering], 1997. **4**(4): p. 10-21.
72. van Dijck, H. and F. van der Heijden, *Object recognition with stereo vision and geometric hashing.* Pattern Recognition Letters, 2003. **24**(1-3): p. 137-146.
73. Chu, C.-H. and Y.-C. Hsu, *Similarity assessment of 3D mechanical components for design reuse.* Robotics and Computer-Integrated Manufacturing, 2006. **22**(4): p. 332-341.
74. Pu, J. and K. Ramani, *On visual similarity based 2D drawing retrieval.* Computer-Aided Design, 2006. **38**(3): p. 249-259.
75. Pollefeys, M., et al., *Visual Modeling with a Hand-Held Camera.* International Journal of Computer Vision, 2004. **59**(3): p. 207-232.

76. Lee, H. and S. Han, *Reconstruction of 3D interacting solids of revolution from 2D orthographic views.* Computer-Aided Design, 2005. **37**(13): p. 1388-1398.
77. Wang, W. and G.G. Grinstein, *A Survey of 3D Solid Reconstruction from 2D Projection Line Drawings.* Computer Graphics Forum, 1993. **12**(2): p. 137-158.
78. Fahiem, M.A., S.A. Haq, and F. Saleemi. *A Review of 3D Reconstruction Techniques from 2D Orthographic Line Drawings*. in *Geometric Modeling and Imaging, 2007. GMAI '07*. 2007.
79. Shi-Xia, L., et al. *A matrix-based approach to reconstruction of 3D objects from three orthographic views*. in *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*. 2000.
80. Cicek, A. and M. Gulesin, *Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution.* Journal of Materials Processing Technology, 2004. **152**(3): p. 291-298.
81. Dimri, J. and B. Gurumoorthy, *Handling sectional views in volume-based approach to automatically construct 3D solid from 2D views.* Computer-Aided Design, 2005. **37**(5): p. 485-495.
82. Aberdeen Group, I., *The Transition from 2D Drafting to 3D Modeling Benchmark Report.* 2006.
83. Zaharia, T.B. and F.J. Preteux. *3D versus 2D/3D shape descriptors: a comparative study*. in *Image Processing: Algorithms and Systems III*. 2004. San Jose, CA, USA: SPIE.
84. Jayanti, S., et al., *Developing an engineering shape benchmark for CAD models.* Computer-Aided Design, 2006. **38**(9): p. 939-953.
85. Matthews, I., J. Xiao, and S. Baker, *2D vs. 3D Deformable Face Models: Representational Power, Construction, and Real-Time Fitting.* International Journal of Computer Vision, 2007. **75**(1): p. 93-113.
86. Anton, H., *Elementary linear algebra*. Sixth ed. 1991: John Wiley & Sons Inc.
87. Strang, G., *Introduction to linear algebra*. 1998, Wellesley Cambridge Press: Wellesley MA.
88. Oppenheim, A.V. and R.W. Schafer, *Digital Signal Processing*. 1975, Englewood Cliffs, NJ: Prentice Hall.
89. Hamming, R.W., *Digital Filters*. Third ed. 1989, Mineola, New York: Dover Publications.
90. Lyons, R.G., *Understanding Digital Signal Processing*. Second ed. 2004, Upper Saddle River, NJ: Prentice Hall.
91. Weeks, M., *Digital Signal Processing using Matlab and Wavelets*. 2007: Infinity Science Press.
92. Walker, J.S., *A primer on wavelets and their scientific applications*. Studies in advanced mathematics, ed. S.G. Krantz. 1999: CRC Press.
93. Jensen, A. and A.l. Cour-Harbo, *Ripples in mathematics*. 2001: Springer.
94. Wickerhauser, M.V., *Adapted wavelet analysis from theory to software*. 1994, Wellesley MA: A K Peters Ltd.
95. Rao, R.M. and A.S. Bopardikar, *Wavelet transforms introduction to theory and applications*. 1998: Addison Wesley Longman Inc.
96. Ogden, R.T., *Essential wavelets for statistical applications and data analysis*. 1997, Boston MA: Birkhauser.

97. Hubbard, B., *The world according to wavelets*. 1998.
98. Mix, D.F., *Random signal processing*. 1995, Englewood Cliffs, NJ: Prentice Hall.
99. Mallat, S., *A wavelet tour of signal processing*. 1999: Academic press.
100. Costa, L.d.F. and R.M.C. Jr., *Shape analysis and classification theory and practice*. Image processing series, ed. P.A. Laplante. 2001: CRC Press.
101. Mix, D.F. and K.J. Olejniczak, *Elements of wavelets for scientists and engineers*. 2003, Hoboken NJ: John Wiley & Sons.
102. Bruce, A. and H.-Y. Gao, *Applied wavelet analysis with s-plus*. 1996: Springer.
103. Chuang, G.C.H. and C.C.J. Kuo, *Wavelet descriptor of planar curves: theory and applications.* Image Processing, IEEE Transactions on, 1996. **5**(1): p. 56-70.
104. Antoine, J.P., et al., *Shape characterization with the wavelet transform.* Signal Processing, 1997. **62**(3): p. 265-290.
105. Osowski, S. and D.D. Nghia, *Fourier and wavelet descriptors for shape recognition using neural networks--a comparative study.* Pattern Recognition, 2002. **35**(9): p. 1949-1957.
106. Davatzikos, C., et al., *Hierarchical active shape models, using the wavelet transform.* Medical Imaging, IEEE Transactions on, 2003. **22**(3): p. 414-423.
107. Lestrel, P.E., et al., *A Fourier-wavelet representation of 2-D shapes: sexual dimorphism in the Japanese cranial base.* Anthropological Science, 2004. **112**(1): p. 3-28.
108. King-Chu, H., et al. *The generalized uniqueness wavelet descriptor*. in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. 1999.
109. Rube, I.E., et al. *Wavelet approximation-based affine invariant 2-D shape matching and classification*. in *Image Processing, 2004. ICIP '04. 2004 International Conference on*. 2004.
110. Kith, K., K. Kith, and E. Zahzah. *Discrete wavelet descriptor for 2D shape*. in *Image and Graphics, 2004. Proceedings. Third International Conference on*. 2004.
111. Chen, G.Y., G.Y. Chen, and W.F. Xie. *Pattern recognition using dual-tree complex wavelet features and SVM*. in *Electrical and Computer Engineering, 2005. Canadian Conference on*. 2005.
112. Yadav, R.B., et al., *Retrieval and classification of shape-based objects using Fourier, generic Fourier, and wavelet-Fourier descriptors technique: A comparative study.* Optics and Lasers in Engineering, 2007. **45**(6): p. 695-708.
113. Yadav, R.B., et al., *Retrieval and classification of objects using generic Fourier, Legendre moment, and wavelet Zernike moment descriptors and recognition using joint transform correlator.* Optics & Laser Technology, 2008. **40**(3): p. 517-527.
114. Kong, X., et al., *A new shape descriptor based on centroid-radii model and wavelet transform.* Optics Communications, 2007. **273**(2): p. 362-366.
115. Hastie, T., R. Tibshirani, and J. Friedman, *The elements of statistical learning*. 2001.
116. Bishop, C., *Pattern recognition and machine learning*. Information science and statistics, ed. M. Jordan, J. Kleinberg, and B. Scholkopf. 2006 New York, NY: Springer.
117. Parker, J.R., *Algorithms for Image Processing and Computer Vision*. 1997.

118. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification*. 2006.
119. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2003.
120. Carpenter, G. and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Networks*. 1991.
121. Wackerly, D.D., W. Mendenhall, and R.L. Scheaffer, *Mathematical Statistics with Applications*, Third, Editor. 2002.
122. Mendenhall, W. and T. Sincich, *Statistics for Engineering and the Sciences*. 1995.
123. Jennrich, R., *An introduction to computational statistics*. 1995, Englewood Cliffs, NJ: Prentice Hall.
124. Kachigan, S.K., *Multivariate statistical analysis*. 1991, New York, NY: Radius Press.
125. Pyle, D., *Data preparation for data mining*. 1999, San Francisco, CA: Morgan Kaufmann.
126. Upton, G. and I. Cook, *Oxford dictionary of statistics*. 2004.
127. Welland, G.V., *Beyond wavelets*. Studies in computational mathematics, ed. C.K. Chui, P. Monk, and L. Wuytack. Vol. Ten. 2003: Academic Press.