Institut für Rechnergestützte Ingenieursysteme

Fakultät Informatik, Elektrotechnik und Informationstechnik

Universität Stuttgart
Universitätsstrasse 38
D – 70569 Stuttgart

Master Thesis Nr. 3374

Implementation of an Optimized Distance Function for Retrieval and Similarity Comparison of Non-rotational Parts

Javier Aguilera Díez

Studiengang:	INFOTECH
Prüfer:	Univ-Prof. Hon-Prof. Dr. Dieter Roller
Betreuer:	M.Sc. Leila Zehtaban
Begonnen am:	17.07.2012
Beendet am:	16.01.2013
CR-Klassifikation:	J.6, I.3.5, I.3.6, I.3.8, I.5.2

Acknowledgments

Me gustaría en primer lugar agradecer a mi familia en general el apoyo que siempre me han dado, en particular a mi hermana y mis padres que siempre han estado ahí para lo bueno y lo malo, y en especial a mis cuatro abuelos que siempre se han preocupado demasiado por mí.

Por otro lado, agradecer también a los amigos del colegio y del barrio que llevamos juntos prácticamente una vida entera, desde los cuatro años, y que hemos seguido manteniendo un contacto muy estrecho incluso después de separarnos cuando comenzamos nuestra nueva etapa universitaria.

En cuanto a mi etapa universitaria, también querría nombrar a todos los amigos nuevos que he hecho durante todo este tiempo, pero sobre todo agradecer a los poquitos con los que he compartido prácticamente toda mi etapa universitaria con ellos, en la que ha habido buenos y malos momentos, pero sobre todo buenos.

Finalmente, me gustaría hacer mención de mi última etapa universitaria como ERASMUS en Stuttgart ya que es donde he hecho mi proyecto fin de carrera. Posiblemente el mejor año de mi vida hasta el día de hoy por todas las nuevas y buenas experiencias vividas, y eso se debe entre muchas cosas a todas las buenas personas que he conocido durante este periodo.

Abstract

In the recent decades, advances in design of 3D solid models have been obtained due to the novel software and computing developments in addition to innovative design possibilities by using Computer Aid Design (CAD) in new product design. One of the superior methods to optimize the design process of 3D shapes is reusing of existing knowledge to assist in the design of new parts. In this concept, a classification system makes use of this huge amount of existing knowledge available and possible for the design as well as for the manufacturing. The advantage of using these knowledge results in reducing time and cost in the design process of a part. To achieve this goal, the philosophy of Group Technology that aims to group and classify parts has been applied. As a well-known approach of Group Technology, the Opitz Code System has been selected for this project which classifies manufacturing features of rotational and non-rotational solid models generating a shape signature. This Thesis aims to analyze, develop and realize the non-rotational section of the Opitz Code. To recognize and extract the features of non-rotational parts, IGES format has been implemented. This format identifies the boundary representation of a shape and generates a file with all their values. For the similarity comparison, a new distance function to compare Opitz codes has been implemented. To conclude this research, a Graphic User Interface (GUI) has been implemented to manage properly the full developed system including the database.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

Design and manufacturing of solid models have had several changes along the last decades. One of the immense leading factors is the possibility of new technologies as computers and software that are advancing the design process of solid models. In the topic of design, Computer Aid Design (CAD) has introduced the use of computer systems to assist in the creation, modification, analysis or optimization of a 3D model design beside Computer Aid Manufacture (CAM) which supports computer software to control machine tools and related machinery in the manufacturing of workpieces. These technologies generate several advantages in the creation process of solid models, reducing cost and time of design and manufacturing.

As a connection between CAD and CAM, techniques with the goal of part classification and grouping have been developed by researches. Group Technology (GT) is one of the effective techniques with the aim of grouping parts to optimize some manufacturing problems; the philosophy of GT is improve the manufacturing efficiency identifying and grouping parts.

One possible classification technique to obtain a good connection between CAD and CAM that follows the GT philosophy is the generation of a shape signature for each part. A shape signature is a vector of digits based in attributes or characteristics of a solid model. There are several methods to extract and generate shape signatures, for example; manufacturing feature recognition, graph based techniques, global feature techniques, methods based in histograms, etc. [IJLKR04].

Opitz Code System is a classification system using GT technique to generate a five digits vector based on features of the part. Opitz Code is divided in two main categories from the beginning; rotational parts and non-rotational parts. This thesis and research work focuses on studying and realization of non-rotational parts. This category includes classification of parts as flat, long or cubic class; the main shape that can be rectangular, angular, triangular, etc. The amount or directions of main bores, machining surface and auxiliary holes can be classified as well in the code.

Once the classification of parts is done by shape signature as the Opitz code, the similarity measure between shape signatures has to be developed. The similarity measurement aims to optimize the design due to retrieval of similar designs which results design time reduction and consequently cost reduction. With similarity techniques, the possibility of storing shape signatures in form of Opitz codes in a database and comparing them using distance functions is an important advantage for the design process of 3D shapes.

1.2 Motivation and problem statement

As a motivation, the achievement of develop a properly system able to classify and compare non-rotational parts to reuse the existing design knowledge of 3D shapes with the aim of reducing time and cost in the design process of parts is going to be developed. For the classification part, a feature extraction method is developed to generate a shape signature based on Opitz Code System. For the comparison between parts, the development of a new distance function to assess similarities between Opitz codes is performed.

The amount of possibilities to extract the properly characteristics of the part and to perform a properly distance function difficult the task to execute. On one hand, the big amount of techniques to generate shape signatures makes the choice of one difficult. Focusing in the literature review, the use of feature extraction methods could be a good application; but the amount of possible features of the part is a big issue too. On the other hand, the classification of non-rotational part is not as common as rotational parts. The amount of possibilities of features in non-rotational parts and the variation of the part according to the perspective and position of the part in the axis difficult considerably the task; also the amount of researches for non-rotational parts is much lesser than for rotational-parts. Finally, the development of a distance function for similarity assessment for Opitz Code System is totally new, there aren´t any research focusing in this topic.

1.3 Objectives of the thesis

The main objective of this Thesis is to develop a system to recognize and extract features for non-rotational parts; with this information generate a shape signature of each part to classify them and compare them with a developed distance function. In the next steps the objective is presented better specified:

- Study and review the different methodologies of shape signature generation and classification, especially different models of GT and Opitz coding system.
- Investigate the feature recognition and extraction of non-rotational parts of products based on Opitz code.
- Automatic Opitz code generation for non-rotational parts.
- Study and implementation of an optimum distance function for Opitz Code System.
- Validate the system generating a database of parts and testing them.

1.4 Organization of the thesis

Based on the objectives defined for the research, this thesis consists of seven chapters explained as following:

- **Chapter 1:** The introduction with the background, the motivation and the objectives of the thesis.
- **Chapter 2:** The literature review of several 3D shape searching techniques based on shape signatures is presented in this section.
- **Chapter 3:** Introduction and explanation of Group Technology philosophy and Opitz Code System.
- **Chapter 4:** Introduction and explanation of the technique based on feature recognition and extraction for solid models. In this chapter, the functionality of IGES format is presented too.
- **Chapter 5:** A literature review of similarity measures for the comparison of shape signatures is presented. The new implemented distance function for Opitz Code System is explained too in this chapter.
- **Chapter 6:** The developed system for classification and comparison of non-rotational parts is presented in this chapter.
- **Chapter 7:** This chapter attaches a final example of the developed program with the validation and a classified database.
- **Chapter 8:** The conclusion of the research is presented in this chapter.

Appendix A, an additional appendix explaining the IGES file entities used in the developed system is included to this Thesis.

Chapter 2

3D SHAPE SEARCHING TECHNIQUES BASED ON SHAPE SIGNATURES

2.1 Introduction

One of the efficient ways to asses shape similarity is to first abstract 3D object parts into a shape signature that could be a graph, a vector or an ordered collection of numeric values [CGK03] and use these shape signatures to perform similarity comparison. So, one of the most important points of comparing 3D shapes is the techniques that generate a shape signature of a 3D model. These signatures should describe the features of the 3D model needed for similarity assessment. The following criteria are important to generate a proper shape signature [IJLKR04]:

- Scope: the shape signature must be able to describe all classes of shapes.
- Uniqueness: there should be only one possibility between the part and the shape signature.
- Stability: the shape signature must be stable for small changes in the shape.
- Sensitivity: the shape signature must be capable of capturing even details of the part.
- Efficiency: it should be easy to compare shape signatures.

There are different ways to classify the techniques that generate shape signature of 3D models; different researches that categorize the methods with different approaches; all of these categorizations are similar because they are based on the same idea of classification methods focus in shape signatures. In the following research it is possible to see the difference [CGK03, IJLKR04]. This survey tries to give an idea of this categorization focus in the paper [IJLKR04].

2.2 Categories of the 3d shape searching techniques

In the following section a classification of different techniques to extract shape signatures of 3D models that is based on [IJLKR04] is presented.

2.2.1 Global-feature-based-techniques

The global properties of 3D model such as moments, invariants, Fourier descriptors or geometry ratios are used in this technique. The shape signature usually is a feature vector with parameters, moments, spherical harmonics, etc. One of the most important advantages of this technique is the high computational efficiency. In the other hand the disadvantage of this technique is that the technique fail to capture the specific details of a shape because the technique is not very robust and it fail to discriminate among locally dissimilar shapes too.

Cybenko et al. [CBC97] developed a 3D base system that a series of feature vectors are extracted, namely, second order moment invariants, spherical-kernel moment invariants, bounding box dimensions, the object centroid and surface areas.

2.2.1.1 Moments

3D Moments are defined in terms of the Riemann integral [IJLKR04].

The most important advantage of this technique is the fast computation and comparison, one of the important characteristics of this method is classification of a big amount of shapes. On the other hand the disadvantage of this technique is the low stability for some classes of shapes due to the big amount of possible moments that are able to be calculated in the shapes.

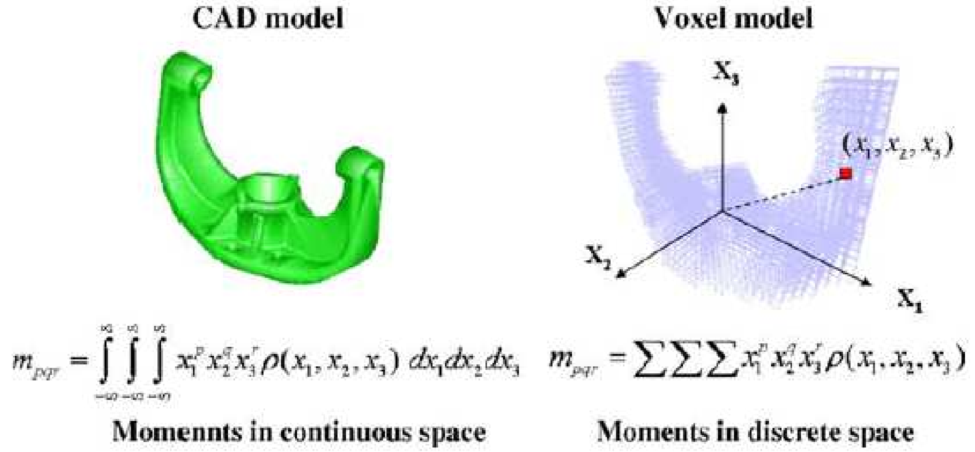


Figure 1: 3D moments in continuous and discrete space [IJLKR04]

2.2.1.2 Spherical harmonics

Spherical harmonics are a decomposition of spherical functions by finding the Fourier transform on a shape. The spherical harmonic coefficients can be used to reconstruct an approximation of the underlying object at different levels. Similar to moments, a partial yet accurate description of the part can be obtained by using a limited subset of Fourier coefficients.

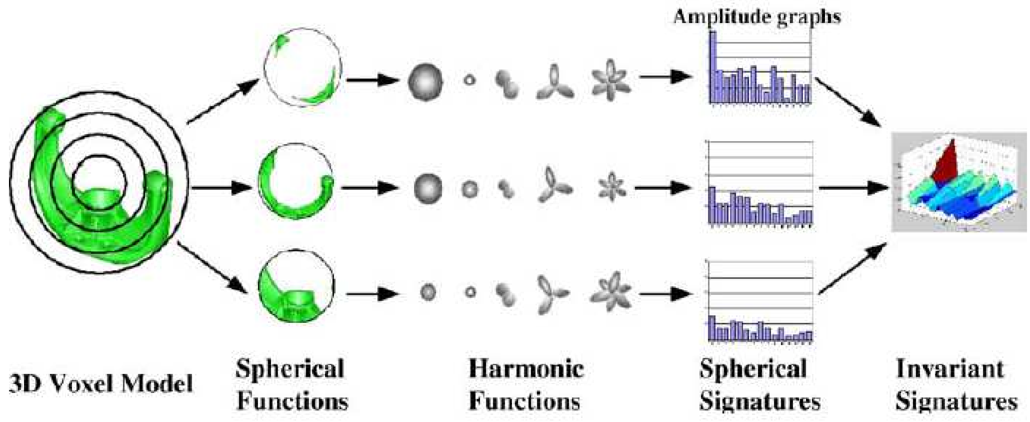


Figure 2: Spherical Harmonics-based representation of a 3D model [IJLKR04]

2.2.1.3 Geometric parameters

Geometric parameters such as surface area to volume ratio, compactness, crinkliness, convex hull features, bounding box aspect ratio, and Euler numbers are also used as shape representations.

2.2.2 Manufacturing feature recognition-based techniques

The shape representation for a 3D model is determined by various properties of extracted machining and manufacturing features, these features usually are holes, slots or pockets. The shape signature can be a vector or a graph [IJLKR04].

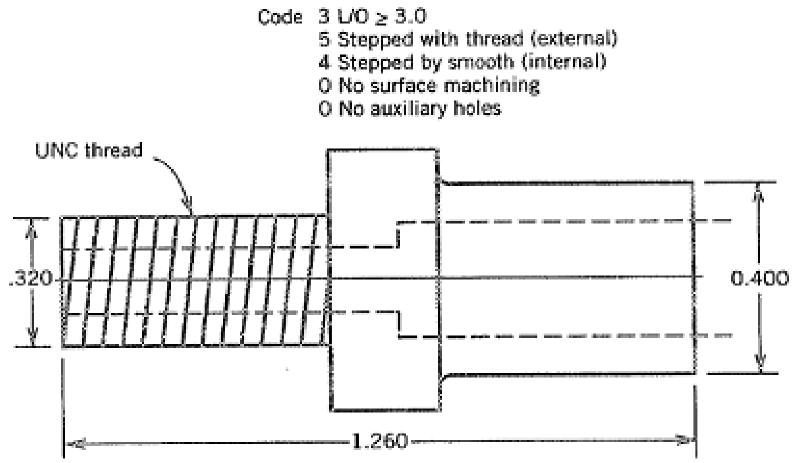


Figure 3: Example of Opitz code that extracts shape features [Hawo68]

The advantage of this technique is that uses manufacturing domain knowledge in the process of the creation of the shape signature. On the other hand there are some disadvantages; for example, if there are same shapes, they may have different recognized features and the technique may require human intervention.

A feature recognition-based technique to identify similarities between parts was developed by Ramesh et al. [RYD01].

2.2.3 Graph-based techniques

The topology of 3D models is represented in the form of a relational data structure such as graphs and trees. Tree comparison tends to be faster and easier compared to graph comparison. However, most engineering components cannot be represented as trees [IJLKR04].

One of the most important advantages of this method are first that the technique allow representation at multiple levels of detail and facilities matching of local geometry and second that graphs can be matched based on exact or inexact matching techniques. On the other hand the most important disadvantages are that the technique is intractable for large graphs because of the high complexity of graph matching problems and graph comparison costs increase proportionally with graph size.

2.2.3.1 B-rep graph matching

3D models are usually represented as Boundary Representation (B-Rep), the most common formats are IGES (Initial Graphics Exchange Specification) and STEP (Standard for Exchange for Product Data). The B-Rep format represents a shape as a graph in terms of bounded B-Spline surface.

The most important advantages of this representation are that the graphs contain detailed shape information and a partial matching is possible. On the other hand the disadvantages of this technique are that a large graph size may affect efficiency or sometimes could happen that similar 3D models do not have the same graph. Other important disadvantage is that sometimes a simple part tends to have a large and complicate shape signature.

El-Mehalawi and Miller [EM03] used the attributed graph matching approach to compare CAD models of engineering parts in the STEP format. The models are converted from the STEP format to graphs with geometric attributes.

2.2.3.2 Spectral graph theory

Spectral graph theory is a branch of mathematics that relates the Eigenvalue spectra of the adjacency matrix of graphs with other geometric invariants of the graphs. One example of this technique was developed by McWherter [MPRS01].

2.2.3.3 Reeb graphs

Reeb graphs define a skeleton structure (tree) which is determined using a continuous scalar function on an object; this function is integrated over the whole body to make it invariant to the starting point. Multi-Resolution Reeb Graph is a 3D model divided into a numbers of levels based on the value of the scalar function [IJLKR04].

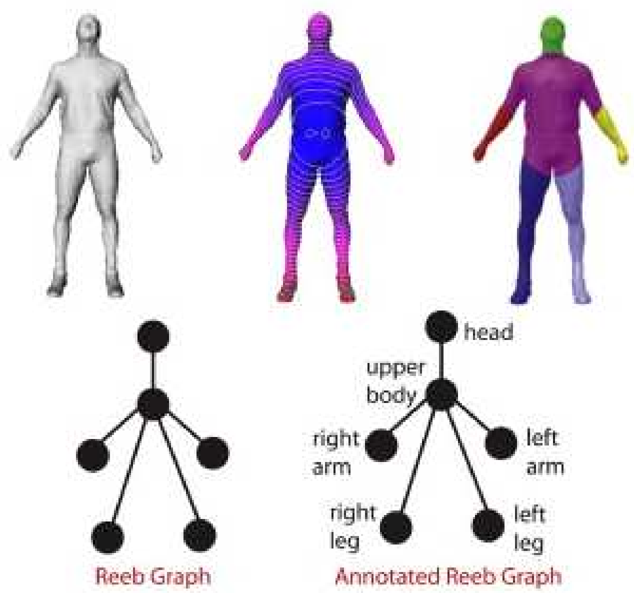


Figure 4: Reeb Graph of a 3D human model [JS12]

This graph representation has the advantage of being more simple than the B-Rep representation and the matching and the comparison is faster than B-Rep too.

Otherwise, the main disadvantages are that this representation is not applicable to all classes of 3D models and the choice of Reeb function could affect results significantly.

Hilaga et al. [HSKK01] used Multi-resolution Reeb Graphs (MRGs) for representing the topology of 3D shapes as an extension of the original Reeb Graphs.

2.2.3.4 Skeletal graphs

Skeletal graphs techniques compute the skeleton of a model and convert it into a skeletal graph (tree) as its shape representation. Skeletonization can be performed by various methods such as distance transform, Thinning or Voronoi-based [IJLKR04].

This graph representation has the advantages of being topology preserving and being smaller in size than B-Rep so they can be used for subgraph isomorphism at a very low computational cost. Other important advantage is that the local part attributes can be stored for a more accurate comparison. Otherwise, the main disadvantage is that many engineering 3D model parts are not amenable to skeletonization by Thinning so this representation is not applicable for all kind of shapes.

Sundar et al. [SSGD03] presented a skeletal graph-based method using distance transform.

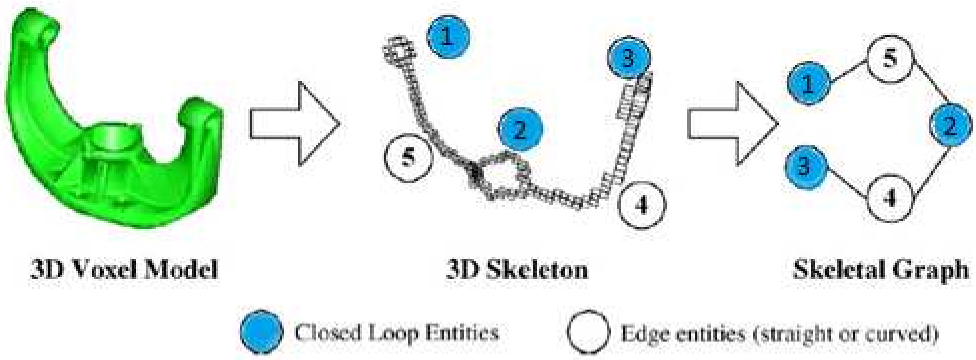


Figure 5: Skeletal Graph based representation [IJLKR04]

2.2.4 Histogram-based-techniques

Histogram-based methods sample points on the surface of the 3D model and extract characteristics from the sampled points. The shape representation is a histogram or distribution and the characteristics are organized in it.

Histogram-based methods have a tradeoff between computational cost and number of sampled points. Sampling a lower number of points leads to very low accuracy.

Ankerst et al. [AKKS99] used shape histograms in searching or 3D protein structures. Before the computing of shape histograms, models are normalized with respect to translation and rotation.

2.2.4.1 Shape histograms

Shape histograms are based on a partitioning of space in which 3D model reside. The space is decomposed into disjoint cells, which correspond to the bins of the histogram. Three techniques are suggested for space partitioning: shell, sector and spiderweb model.

The main advantages of shape histograms are that is a good method for general shape classification and the possible quick comparison in large databases. On the other hand a large number of bins are required to accurately describe a shape and this may be a good disadvantage [IJLKR04].

2.2.4.2 Shape distributions

The shape representation is a probability distribution sampled from a shape function measuring the geometric properties of a 3D model. The main problem of these distributions is that sometimes different 3D model parts can have similar distributions.

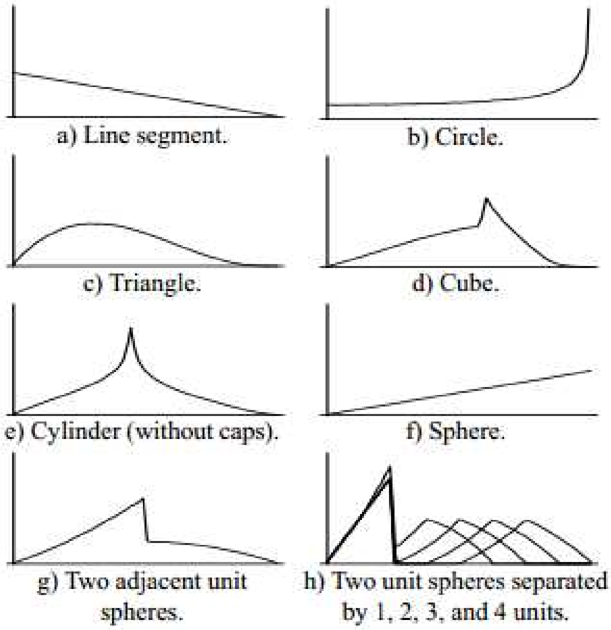


Figure 6: Shape Distributions of 3D models [OFCD01]

2.2.5 Product information-based techniques

These techniques are specifically designed for the domain of engineering parts. Part designs are described either based on their manufacturing attributes or on their geometry.

A system for exchanging of product information between designers and manufacturing service providers over the Internet in a distributed setting was developed by Kalyanapasupathy et al. [KLM97].

2.2.5.1 Group Technology

Group Technology is a manufacturing philosophy in which the parts having similarities (geometry or manufacturing process) are grouped together to achieve higher level of integration between the design and manufacturing functions of a firm. The shape descriptor will be a GT Code [GT12]. In the next chapter, a full description of this philosophy will be explained.

The main advantage of GT is using enterprise specific domain knowledge but otherwise this technique is difficult to automate because of the high dependence on human intervention.

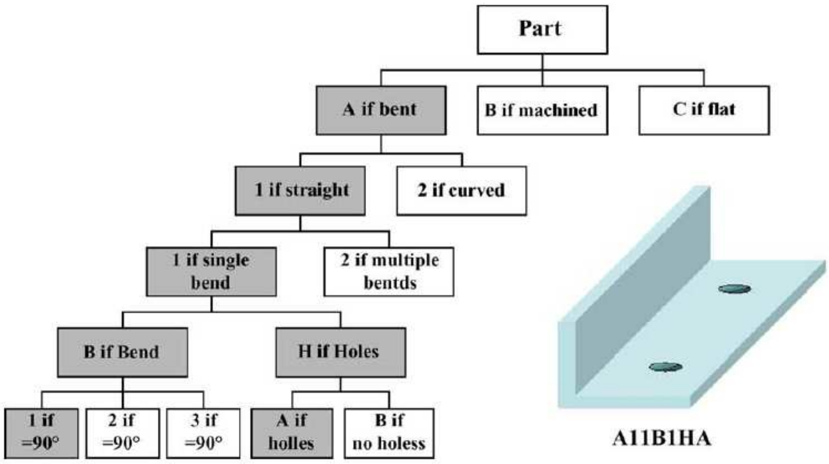


Figure 7: Example of a Group Technology Code [IJLKR04]

2.2.5.2 Section image-based

These methods rely on the 2D silhouette representation of parts of their sections to determine similarity. The shape descriptor is often a feature vector.

This technique works well for legacy drawings in 2D and it is useful for searching 3D models from 2D drawings. Otherwise, this technique need a large amount of data or time required for training neural network for complex shapes.

2.2.6 3D object recognition-based techniques

In the section is presented three methods that have been used for detecting shape similarity from a database of models [IJLKR04]. The most important disadvantage of these techniques is that have been tested for limited shapes and have high storage/computational costs.

2.2.6.1 Aspect graph

The aspect graph representation indentifies regions of the viewing sphere where equivalent views and neighborhood relations on the viewing sphere generate a graphical structure of views.

The main advantage of this representation is that this technique is useful for image based recognition of components. On the other hand the main problem is that this technique requires a very high storage space.

Cyr and Kimia [CK01] used aspect graphs to assess the similarity between 3D models.

2.2.6.2 Extended Gaussian Images (EGI)

Depth maps or needle maps computed for real-world scenes are processed to create an orientation histogram for the visible half of the Gaussian sphere in segmented objects.

This technique has a unique representation for convex objects, which is a good advantage, but on the other hand an infinite number of non-convex objects can possess the same EGI.

Shum proposed a variant of the EGI in [SHI96].

2.2.6.3 Geometric hashing

A 3D object is parsed into basic geometric features such as surface points.

This technique has the advantage that it is very useful for extract matching between 3D models. Otherwise the main problem is the very high storage space that this technique requires.

Wolfson and Rigoustos [WR97] described the 3D implementation aspects of the geometric hashing algorithm.

Chapter 3

GROUP TECHNOLOGY & OPITZ CODE

3.1 Group Technology

3.1.1 Definition

Nowadays due to the huge amount of data that humans has to manage, it is important to find ways to organize this data for reducing access and searching time. To accomplish this, lot of methods have been developed. Focus on this Thesis, in the field of manufacturing parts; a method that classifies parts and organizes them could be a good idea to reduce time and costs in design and manufacture of shapes.

Group Technology is a design and manufacturing philosophy in which parts having similarities (geometry, manufacturing process or function) used to be categorized into groups in order to achieve economies of scale normally associated with high-volume production. For the grouping of the parts a Group Technology Code is used [GT12].

3.1.2 Applications

The Group Technology technique has a large among of application but the major role of GT is in design systems and manufacturing systems. The most important categories of application of GT are:

- Systems based on part design attributes.
- Systems based on part manufacturing attributes.
- Systems based on both design and manufacturing attributes.

With Group Technology it is possible to develop design systems that are useful to design shapes models with retrieval information and promote design standardization and manufacturing systems that lead to similar manufacturing machines or process for similar parts, called Cellular Manufacturing [Tolo10].

3.1.3 Classification of shapes

Based on [Sund94], there are three ways for classification of parts that have been identified.

1. Visual inspection: may use photos or part prints and these methods utilize subjective judgment.
2. Grouping by parts coding (PCA): coding systems are used to classify the shapes.
3. Production flow analysis (PFA): information contained on the route sheet is used in these methods.

3.1.4 Advantages of group technology

In this section, some of the most important advantages that can be resulted using GT in design and manufacture of parts are listed as [TR84] follows:

- Reduce engineering costs.
- Enable cellular manufacturing.
- Accelerate product development.
- Improve costing accuracy.
- Simplify process planning with design standardizations and similar manufacturing systems.
- Reduce tooling costs.

3.1.5 Structure of Group Technology codes

A Group Technology (GT) code is an alphanumeric string which represents important information about the products (features, volume, size, characteristic, etc.). Comparing the GT codes of two products is a quick and efficient method for estimating product similarity. GT codes can be used to search a database of products and retrieve the designs and process plans of those products which are similar to a given design, to generate new process plans automatically using a knowledge-based system, and to assess manufacturability of a product design.

Group Technology codes can be made with different structures. Three main structures for creating GT codes based on [GT12] are discussed:

3.1.5.1 Hierarchical structures (monocodes)

Here, each digit (or position) in the code represents a feature/sub-group. For example, in the figure 8, one digit divide the parts in two groups, rotational and prismatic shapes and other digit divide the parts in different features, with or without holes [LMS08].

In this sense, each subsequent digit is qualified by the preceding digits (or, in an object-oriented sense, each subsequent digit inherits the properties of the previous digits).

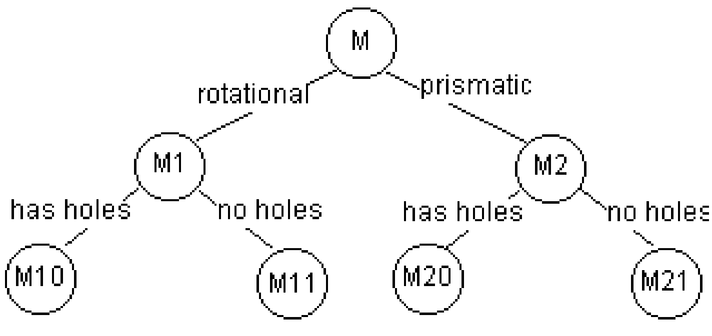


Figure 8: Monocode structure [LMS08]

Advantages of monocodes:

- With just a few digits, a very large amount of information can be stored.
- The hierarchical structure allows parts of the code to be used for information at different levels of abstraction.

Disadvantages:

- Impossible to get a good hierarchical structure for most features/groups.
- Different sub-groups may have different levels of sub-sub-groups, thereby leading to blank codes in some positions.

3.1.5.2 Chain codes (polycodes)

In this method, the code digit represents one feature. Thus, the value of any given digit (or position) within the code has no relation to the other digits [LMS08].

Advantages:

- Easy to formulate

Disadvantages:

- Less information is stored per digit; therefore to get a meaningful comparison of, say, shape, very long codes will be required.
- Comparison of coded parts (to check for similarity) requires more work.

3.1.5.3 Hybrid codes

In this case, the code for a part is a mixture of polycodes and monocodes. It retains the advantage of both structures. This is the most commonly used method of coding and classification. One important example is the Opitz Code that is a Hybrid Code [LMS08].

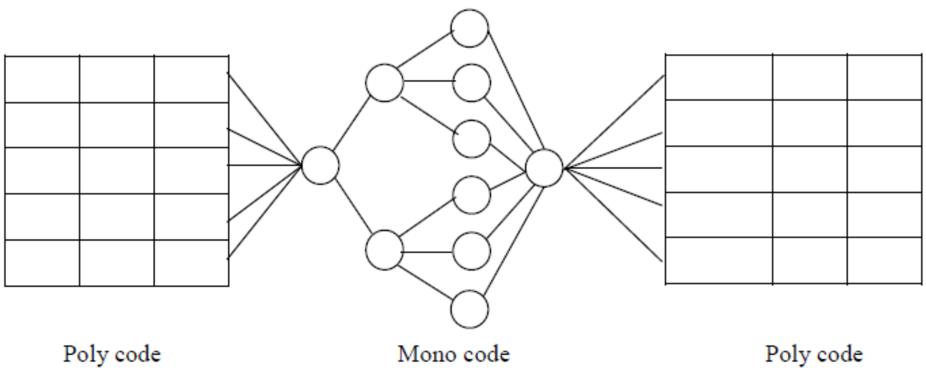


Figure 9: Hybrid structure [LMS08]

3.1.6 Different GT codes

Some of the most famous GT codes are explained in the next section [CM 7, MS, CG 07].

3.1.6.1 Opitz code

Opitz code is one of the most important codes for classifications shapes in Group Technology. It can have 3 sections; it starts with a five-digit "geometric form code" followed by a four-digit "supplementary code" and then at the end it may be followed by a company-specific four-digit "secondary code" for describing production operations and sequencing [Hawo68].

In the next table is represented the basic structure of the first nine digits of the code.

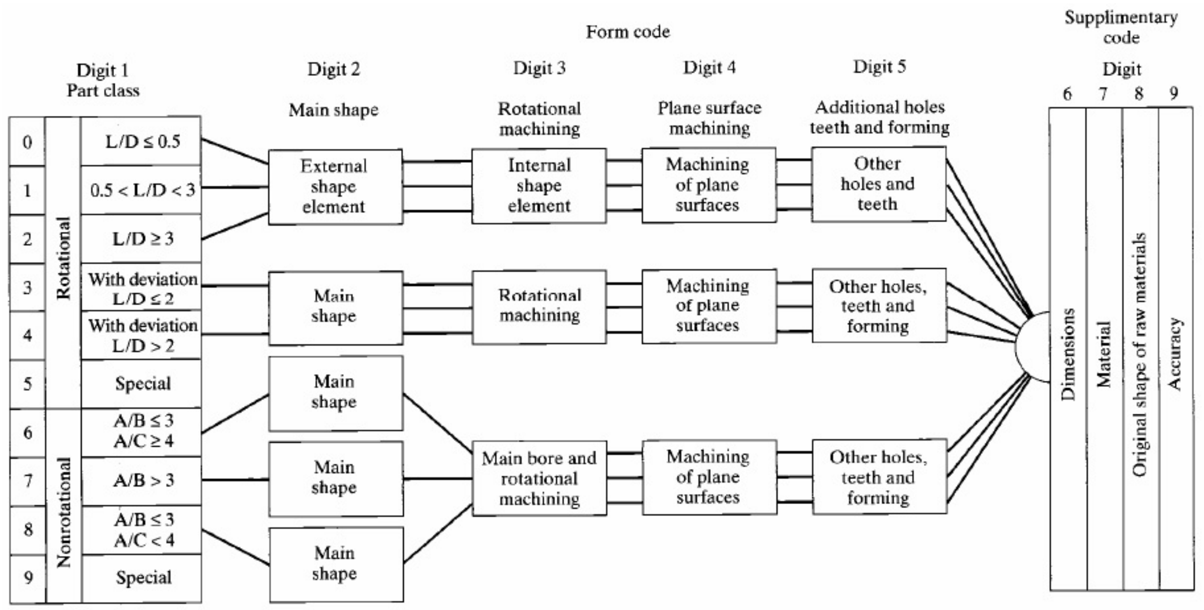


Figure 10: Basic structure of the Opitz Code [Opit68]

3.1.6.2 DCLASS code

Design and Classification Information System (DCLASS) was developed at Brigham Young University. Several premises were adopted and used as the basis for the development of the DCLASS code:

1. A part may be best characterized by its basic shape, usually its most apparent attribute.
2. Each basic shape may have several features, such as holes, slots, threads, and grooves.
3. A part can be completely characterized by basic shape; features; size; precision; and material type, form, and condition.
4. 4 Several short code segments can be linked to form a part classification code that is human recognizable and adequate for human monitoring.
5. Each of these code segments can point to more detailed information.

The DCLASS part family code is formed of eight digits partitioned into five code segments as it can be seen in the figure 11. The first segment, composed of three digits, classifies the basic shape. The form features code is the next segment and it has only one digit; this code is used to specify the complexity of the part, which includes features (such as holes and slots), heat treatments, and special surface finishes. The one-digit-size code is the third segment of the part family code. The fourth segment denotes precision; it is one digit in length. The final two digits, which comprise the fifth segment of the part family code, are used to classify the material type [ES07].

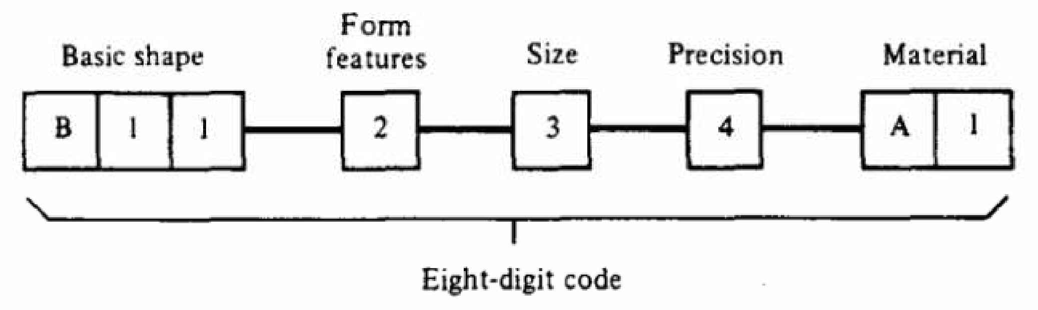


Figure 11: DCLASS structure code

3.1.6.3 MICLASS code

MICLASS stands for Metal Institute Classification System and was developed by the Netherlands Organization for Applied Scientific Research (TNO). It was started in Europe about five years before being introduced in the United States in 1974. The MICLASS was developed to help automate and standardize a number of design, production, and management functions, which includes:

- Standardization of engineering drawings
- Retrieval of drawings according to classification number
- Standardization of process routing
- Automated process planning
- Selection of parts for processing on particular groups of machine tools
- Machine tool investment analysis

The MICLASS classification number can range from 12 to 30 digits. The first 12 digits are a universal code that can be applied to any part. Up to 18 additional digits can be used to code data that are specific to the particular company or industry [ES07].

The workpiece attributes coded in the first 12 digits of the MICLASS are as follows:

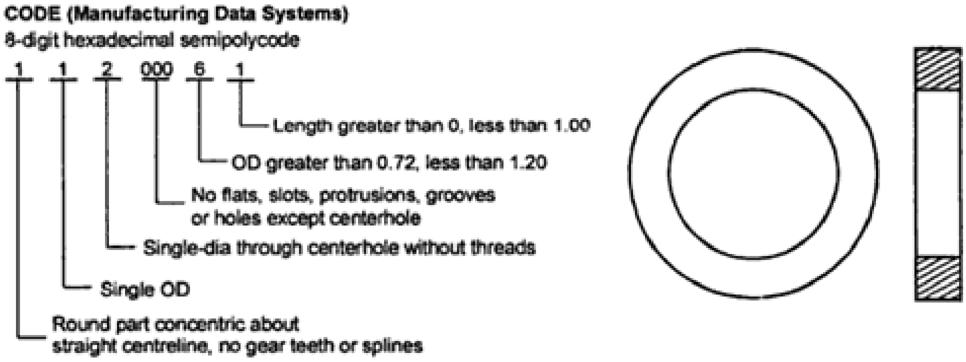
Code position	Item
1	Main shape
2	Shape elements
3	
4	Position of shape element
5	Main dimension
6	
7	Dimension ratio
8	Auxiliary dimension
9	Tolerance codes
10	
11	Material codes
12	

Figure 12: The first twelve digits of MICLASS Code

3.1.6.4 CODE system

The CODE system is a parts classification and coding system developed and marketed by Manufacturing Data Systems (MDSI). Its most universal application is in design engineering for retrieval of part design data, but it also has applications in manufacturing process planning, purchasing, tool design and inventory control.

The CODE number has eight digits. For each digit there are 16 possible values (zero through 9 and A through F) which are used to describe the part´s design and manufacturing characteristics. The initial digit position indicates the basic geometry of the part and is called the Major Division of the CODE system. This digit would be used to specify whether the shape was a cylinder, flat piece, block, or other. The interpretation of the remaining seven digits depends on the value of the first digit, but these remaining digits form a chain-type structure. Hence the CODE system possesses a hybrid structure [ES07].



2nd and 3rd digits	Basic geometry and principal manufacturing process.
4th, 5th and 6th digits	Secondary manufacturing process, e.g. threads, grooves, slots, etc.
7th and 8th digits	Overall size of the part.

Figure 13: Code System structure

3.1.6.5 KK-3 code

It is a general-purpose classification and coding system for machined parts, developed by JSPMI in 1976. The feature of KK-3 is its length, 21 digits, and each digit may have 10 attributes attached to it. It is able to carry more information than all other coding systems. It classifies the dimensions and aspect ratio of the parts [CM12].

Digit	Item	Description
1	Part name	General classification
2		Detail classification
3	Materials	General classification
4		Detail classification
5	Chief dimensions	Maximun length
6		Maximun diameter
7		Tatio of length/diameter
8	Extermal surface	General outer shape
9		Concentric screw threads
10		Functional Groove
11		Irregular shape
12		Shaped surface
13		Cyclic surface
14	Internal surface	General internal surface
15		Curved internal shape
16		Internal plane and cyclic surface
17	End surface	
18	Nonconcentric hole	Pattern of hole
19		Special hole
20	Non machining operation	
21	accuracy	

Figure 14: KK-3 Code structure

3.2 Opitz code

3.2.1 Definition

One of the most famous classification part codes in Group Technology is the Opitz code. This code system was initially proposed by Herwart Opitz in 1970 at Aachen Technology University in Germany [Opit70]. The code has a maximum of 14 digits and each digit may contain 10 different values (attributes). The first five digits are the form code that describe the primary design attributes of the part, then the second four numbers are the supplementary code that describe manufacturing related attributes and finally there are a non really common part that is called secondary code that give more detail of manufacturing attributes.

FORM CODE	SUPLEMENTARY CODE	SECONDARY CODE
1 2 3 4 5	6 7 8 9 10	A B C D

Figure 15: Opitz Code values

Some of the advantages of this code are that it is not proprietary, it is widely used, provides a basic framework for understanding the classification and coding process, it can be applied to machined parts, non-machined parts and purchased parts, it considers both design and manufacturing information.

The applications of the Opitz classification code system are showed in the following table [Hawo68]:

CONCERNED DEPARTMENT	FACILITIES PROVIDED
Design	Variety reduction Recognition of repeat or similar parts
Standards	Standard components easily identified Uniformity of characteristics
Production planning	Use of repeat Grouping parts requiring same machines Use of standard times
Production control	Suitability for Data processing
Production	Parts family manufacture
Equipment	Adapting the machine tool to the workpieces required

Figure 16: Applications of the Opitz Code System

In the form code, the first digit is the one that makes the difference between rotational and non-rotational parts and in this digit it is used a dimensional ratio to evaluate the geometry of the shape. For rotational parts the code uses the length (L) and the diameter

(D) of the part to classify it and for non-rotational parts the code uses the edge lengths of the components in decreasing order of magnitude (A, B and C). Then the second digit is for external shapes and relevant forms, these features are recognized as stepped, conical or straight contours. Threads and grooves are also important. The third digit is for internal shapes, features are solid, bored, straight or bored in a stepped diameter, threads and grooves are integral part. The fourth digit is for the surface plane machining, such as internal or external curved surfaces, slots ar splines. And finally the fifth digit is for auxiliary holes and gear teeth.

In the supplementary code there are four digits, the first one is for diameter or length of the workpiece, the second one is for material used, the third one is for raw materials like round bar, sheet metal, casting or tubing and the fourth digit is for the accuracy of the workpiece.

In the next table, the general structure of the code is showed [Opit70]:

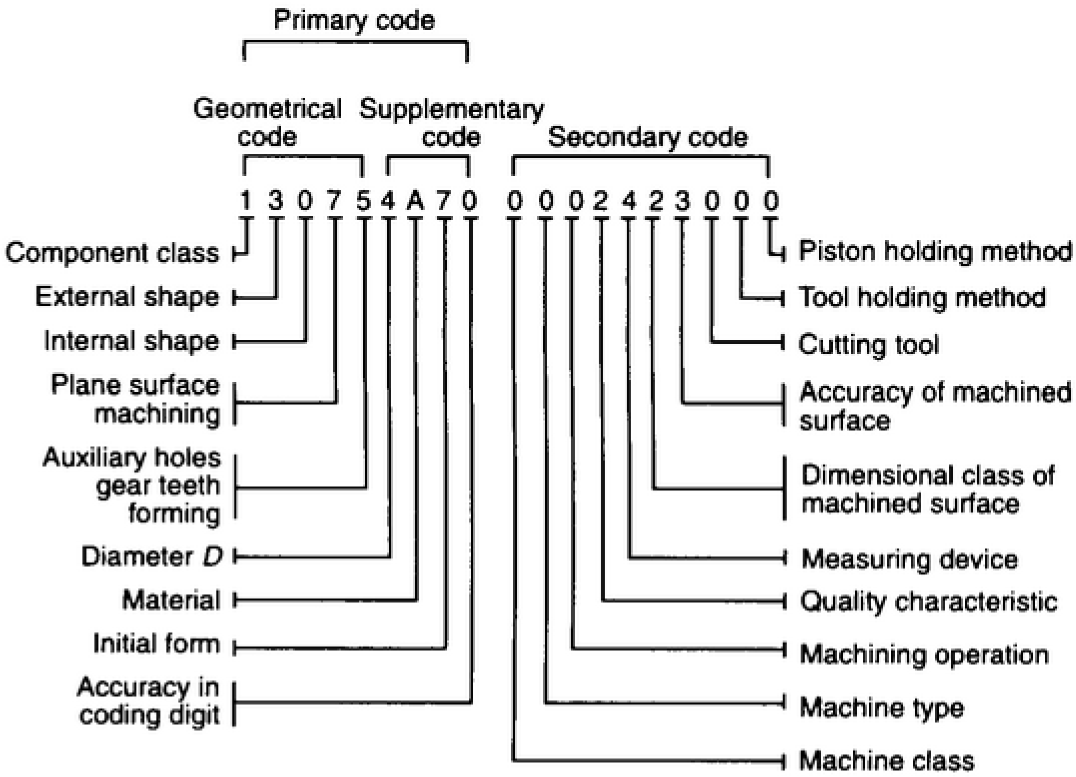


Figure 17: Basic structure of the Opitz Code System

3.2.2. Classification of the digits

The first digit of the Opitz Code is the most important digit that classifies the parts in rotational and non-rotational; due to this Thesis that is focused on non-rotational parts, in this section is going to be explained all the attributes of the five digits of the form code that classify the non-rotational parts. In the following tables the specification of each attribute of the digits is presented. The first table (figure 18) shows all the possible attributes of the five digits of the Opitz Code for the flat parts, when the digit one is six. The second table (figure 19) represents all the attributes of the Opitz Code that classify the long parts, when the digit one of the code is seven and the third table (figure 20) represents the attributes that Opitz Code System has to classify the cubic parts, when the digit one is eight.

DIGIT 2 MAIN FORM			DIGIT 3 Main bore and rotational machining		DIGIT 4 Machining of plane surfaces		DIGIT 5 Other holes, teeths and forming			
0	Plane/flat	Rectangular plane	0	No features	0	Without surface machining	0	Without features		
1	Plane/flat	Right-angled triangle plane	1	One smooth bore	1	Chamfers	1	Without transformation /without gearing		One bore direction
2	Plane/flat	Angularly	2	One bore multiple ascending	2	A flat surface	2	Without transformation /without gearing		Several bore directions
3	Plane/flat	Circular and rectangular	3	One main bore with all form elements	3	Stepped surface	3	Without transformation /without gearing	With hole	One bore direction
4	Plane/flat	Other	4	Two main bores parallels	4	Stepped surface vertically inclined and/or opposed	4	Without transformation /without gearing	With hole	Several bore directions
5	Flat part rectangular or orthogonal with small deviations		5	More than two main bores paralells	5	Groove and/or slot	5	Tranformation /without gearing		Formed without drilling
6	Flat part round or any other shape with small deviations		6	Many main bored perpendicular	6	Groove and/or slot and 4	6	Tranformation /without gearing		Formed with drilling
7	Flat part with regularly arched form		7	Ring groove machining surfaces	7	Curved surface	7	Gearing		
8	Flat part with irregularly arched form		8	7 + main bore	8	Guided surface	8	Gearing with hole		
9	Other		9	Other	9	Other	9	Other		

Figure 18: Attributes of the digits that classify non-rotational flat parts [Opit68]

DIGIT 2 MAIN FORM				DIGIT 3 Main bore and rotational machining		DIGIT 4 Machining of plane surfaces		DIGIT 5 Other holes, teeths and forming			
0	Straight form	Same cross-section	Rectangular cross-section	0	No features	0	Without surface machining	0	Without features		
1			Ortogonal cross-section	1	One smooth bore	1	Chamfers	1	Without transformation /without gearing		One bore direction
2			Any cross-section	2	One bore multiple ascending	2	A flat surface	2			Several bore directions
3		Rectangular cross-section		3	One main bore with all form elements	3	Stepped surface	3		With hole	One bore direction
4		Rectangular and ortogonal cross-section		4	Two main bores parallels	4	Stepped surface vertically inclined and/or opposed	4			Several bore directions
5		Other		5	More than two main bores paralells	5	Groove and/or slot	5	Tranformation /without gearing		Formed without drilling
6	Curved form	Rectangular, angular arbitrary cross-section		6	Many main bored perpendicular	6	Groove and/or slot and 4	6			Formed with drilling
7		Shaped part		7	Ring groove machining surfaces	7	Curved surface	7	Gearing		
8		Shaped part with deviations in the axis		8	7 + main bore	8	Guided surface	8	Gearing with hole		
9		Other		9	Other	9	Other	9	Other		

Figure 19: Attributes of the digits that classify non-rotational long parts [Opit68]

DIGIT 2 MAIN FORM				DIGIT 3 Main bore and rotational machining		DIGIT 4 Machining of plane surfaces		DIGIT 5 Other holes, teeths and forming			
0	Block-like parts		Cuboid	0	No features	0	Without surface machining	0			Without features
1			Ortogonal parts	1	One smooth bore	1	Chamfers	1	Without transformation /without gearing		One bore direction
2			Composite parallelepiped	2	One bore multiple ascending	2	A flat surface	2			Several bore directions
3			Parts mit mounting surfaces and main bore	3	One main bore with all form elements	3	Stepped surface	3		With hole pattern	One bore direction
4			Parts with mounting surfaces and main bore with distribution area	4	Two main bores parallels	4	Stepped surface vertically inclined and/or opposed	4			Several bore directions
5			Other	5	More than two main bores paralells	5	Groove and/or slot	5	Tranformation /without gearing		Formed without drilling
6	Case-like parts	Not shared case		6	Many main bored perpendicular	6	Groove and/or slot and 4	6			Formed with drilling
7			Any form	7	Ring groove machining surfaces	7	Curved surface	7			Gearing
8		Shared case		8	7 + main bore	8	Guided surface	8			Gearing with hole
9			Any form	9	Other	9	Other	9			Other

Figure 20: Attributes of the digits that classify non-rotational cubic parts [Opit68]

3.2.3 Examples

For the end of this part, a couple of examples of rotational and non-rotational part are introduced to facilitate the compression of the functionality of the code:

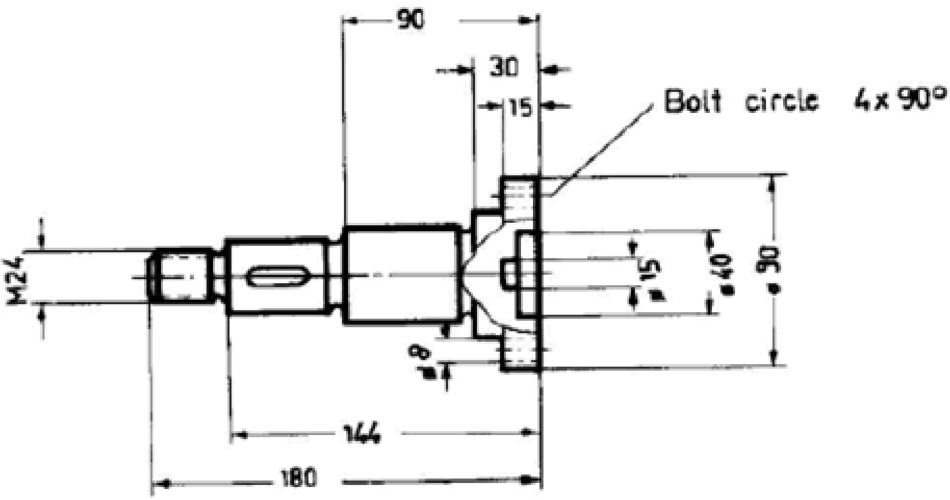


Figure 21: Rotational 3D model

The figure 21 shows a rotational part; the Opitz Code of this 3D model is 12131. The first digit of the code means the part class, in this case the number 1 says that the part is rotational with L/D=2. The second number of the code is 2, this number means that the shape is asymmetrically and has an ascending stepped thread. The third digit of the code is 1; this value signifies that the internal shape has asymmetrically ascending steps. The forth digit in the code is 3 and this means that the surface machining is an external slot. Finally, the fifth digit of the code is 2, this digit classify the auxiliary holes and means that the shape has auxiliary holes evenly spaced along the bolt circle.

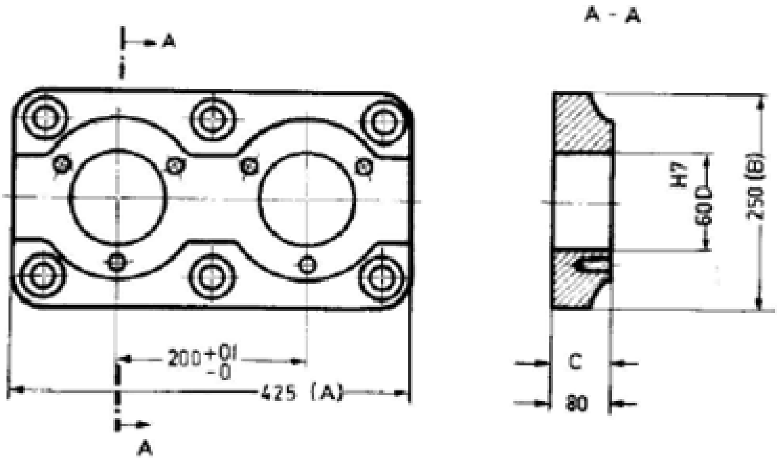


Figure 22: Non-rotational 3D model

The figure 22 is presented a non-rotational part; the Opitz Code value of this shape is 65443. The first digit that classifies the size of the shape and show that it is a non-rotational part is 6; in this case it is represented a flat part. The second digit that is 5 gives the information of the main shape; the part has small deviations in the surface.

The third digit of the code classifies the main holes and in this case it is 4 that mean that the shape has two main holes. The fourth digit of this part is 4 again and signifies that the surface is a stepped plane. Finally, the fifth digit that classifies the auxiliary holes is 3; in this case, there is a pattern for auxiliary holes in one direction.

Chapter 4

FEATURE TECHNOLOGY

4.1 Introduction

Feature technology is a concept that includes all the techniques that work with features. Nowadays the use of features is seen by many researches as the key to get a good integration between the design and the manufacture planning of shapes in the industry area.

The word “feature” signifies different meaning in different contexts depending on the specific domain. A feature, in computer-aided design (CAD), can be different depending if it is referred to design, machining or manufacturing information. For example, in design, feature is used as a net or a notch section, in manufacturing it is used for features as holes, slots, bosses and pockets, a machining feature can be regarded as the volume swept by a cutting tool. Feature data in a CAD model can be represented either as a collection of surfaces or volumetrically, surface features are usually used for example to describe manufacturing tolerances or locating surfaces in design.

Although there are lots of meaning for feature, the basic of these definitions is that features represent the engineering meaning of the geometry of a part, assembly, or other manufacturing activity. In this Thesis, features are seen with their manufacturing point of view, features such as holes, slots and pockets which the designed part should possess.

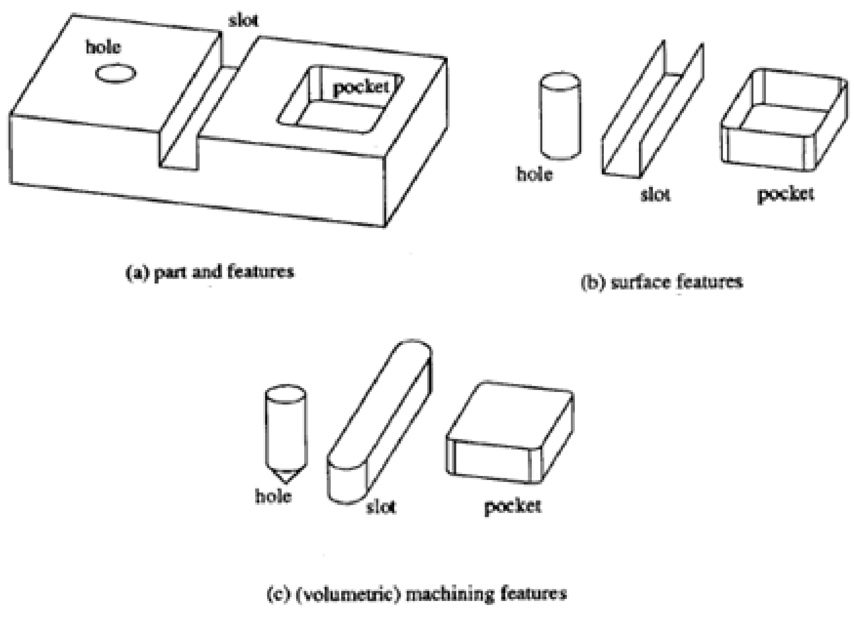


Figure 23: different types of features [HPR00]

4.2 Design by features vs. feature recognition

Feature technology is divided in two different approaches, design by features and feature recognition. Design by features that can be called feature-based design (FBD) is the way of using manufacturing features to accomplish the construction of parts and feature recognition (FR) is a method to algorithmically extract higher level entities (holes, loops, pockets, etc.) from lower level elements (curves, lines, surfaces, edges, etc.) [HPR00].

The advantage of design by features is that design with pre-defined form feature can make the process more efficient and features can serve as functional elements for designers. But the way to design parts with manufacturing features can be a problem because it forces the designer to think in terms of manufacturing operations during the design process which often is not natural to do properly the design.

Automatic Feature Recognition (AFR) is regarded as an ideal solution to automate design and manufacturing processes. Computer Aided Process Planning (CAPP) is the way that CAD and CAM can be integrated, the link between CAD and CAM in that it provides for the planning of the process to be used in manufacturing a designed part [AK06].

Feature Recognition is a discipline of solid modeling that focus on the design and implementation of algorithm for detecting manufacturing information (holes, slots, pockets, etc.) from solid models produced by CAD systems. One example of an integration system of Feature Recognition is proposed by Jung Hyun Han [HHLY01].

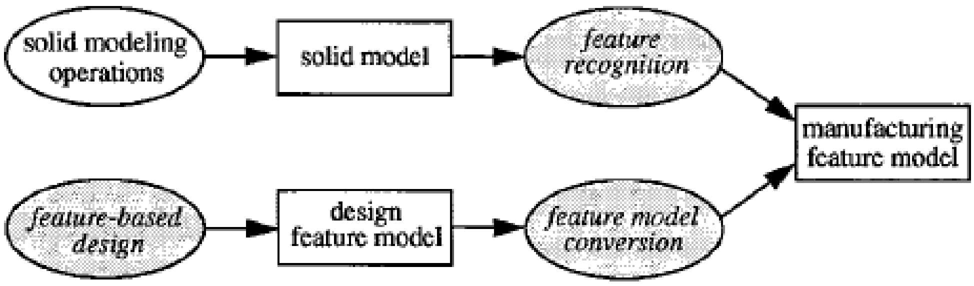


Figure 24: Feature model generation

There is a big among of different Feature Recognition techniques that have been done for CAD/CAM integration. Inside these techniques there are different approaches to perform the recognition of the features. The most important techniques are: the graph based approach, the hint based approach and the volumetric decomposition approach [HPR00].

In the graph based feature recognition that was first normalized by Joshi with the Graph Pattern Analysis approach [JC88], a model of parts is converted into a graph. Then this graph can be compared to other graph stored in a database and the largest common subgraph between them has to be determined in order to assess similarity. After performing the feature extraction, the graph representing the features and their interactions is defined. Then the nodes of this graph correspond to features and store attributes of the features and the arcs represent edges. In [ENR97] a graph representation of the input 3D models is represented as the shape signature for the model.

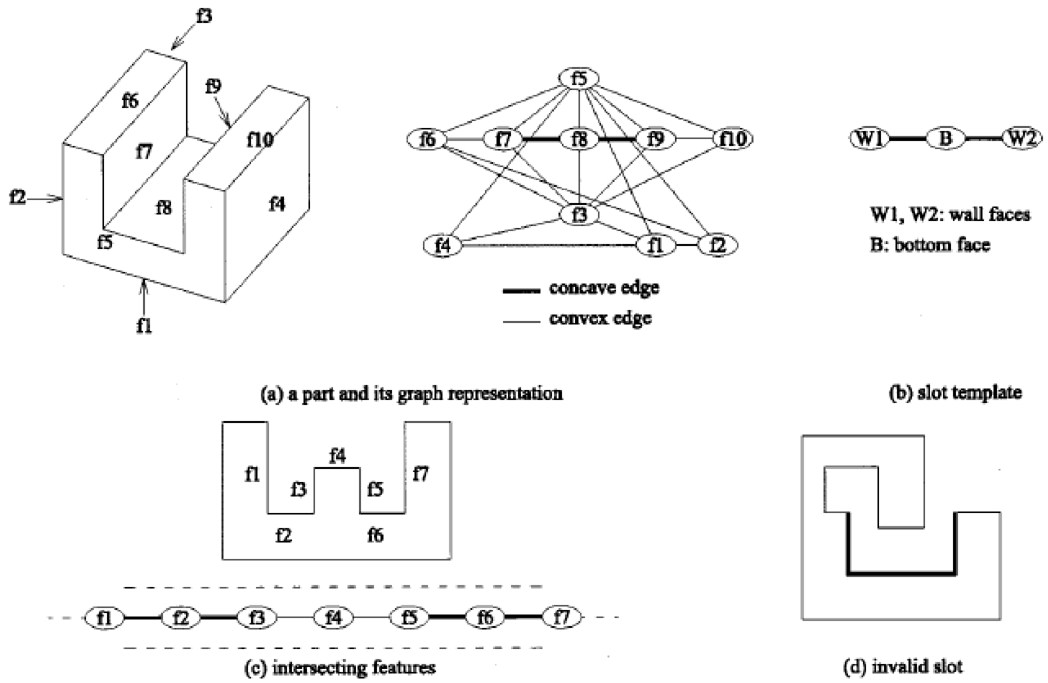


Figure 25: Graph pattern analysis [ENR97]

In the volumetric decomposition approach, the similarity assessment is reached by the decomposition of the input object into a set of volumes and then by manipulating the volumes to produce features. There are two important algorithms that use this approach, Convex Hull Decomposition and Cell-based Decomposition. The convex Hull Decomposition was investigated by Kim [WK98] and consists in four steps: Alternating Sum of Volumes with Partitioning Decomposition (ASVPD), recognition/generation of form features, generation of primitive machining features and machining feature aggregation. The Cell-based Decomposition essentially consists in three steps: delta volume decomposition into cells, cells composition and feature classification.

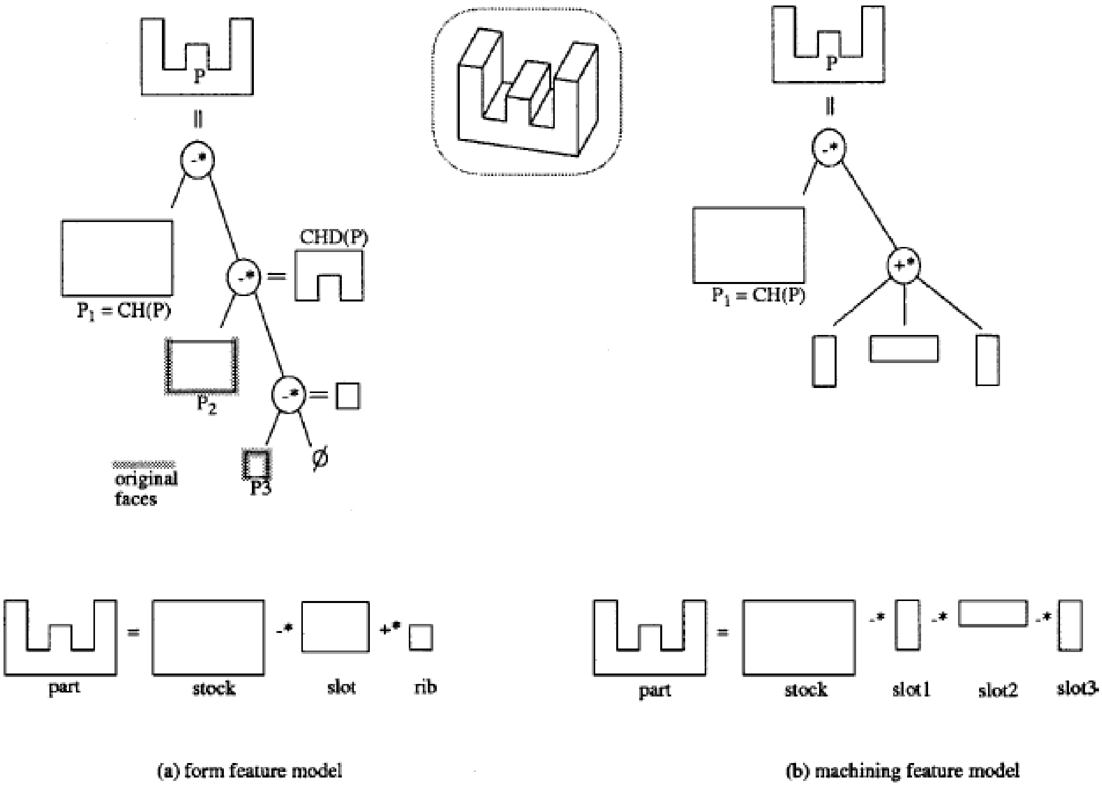


Figure 26: Convex Hull Decomposition

The Hint based approach is not only based on the recognition of machinable features from solid models of parts but also from additional data such design features, tolerances or surfaces attributes.

4.3 Feature extraction

The link between Computer-aided Design (CAD) systems and Computer-aided Manufacturing (CAM) systems is done though Computer-aided Process Planning (CAPP) systems. Feature Recognition is one of the most important parts in CAPP that join the design of a solid model with the manufacturing. Feature Recognition uses some specifics tools to extract the features, all these tools that are used to extract features form the concept of Feature Extraction. One of the most famous tools for extracting features is IGES file format [MKJ08].

4.3.1 Boundary representation (B-rep)

Boundary representation (B-rep) is one of the solid modeling methods that is used to create a solid model of a physical object. Boundary representation describes the geometry of an object in terms of its boundaries, namely the vertices, edges and surfaces which represent entities. The boundary (B-rep) geometrical information of the part design is analyzed by the feature recognition system that is created to extract the manufacturing features of the designed figure [Requ80].

4.3.2 Constructive solid geometry (CSG)

Constructive solid geometry (CSG) is a technique that is used to represent solid models. Constructive models represent a solid as a combination of primitive solids using Booleans operations. These primitive models can be cone, cylinder, sphere, torus, block, etc. and the Booleans operations are union, intersection and difference. The part design that is usually represented through CAD software should represent a solid model by using CSG technique as a design tool [Requ80].

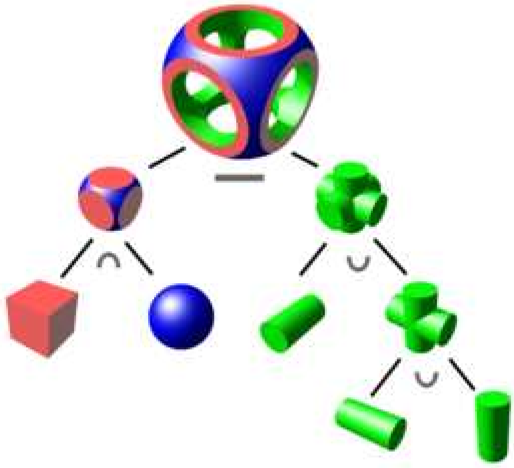


Figure 27: Solid modeling by CSG technique

4.3.3 Initial Graphic Exchange Specification (IGES)

IGES is a standard format that is used to define the data of the object drawing in solid modeling CAD systems in B-rep structure. The geometry and topology information of a solid model can be represented by the entry fields of the IGES file. The lower level of the format is the entity. The entities are classified in some classes; the most important classes are geometry entities, topological entities and non-geometry entities. The geometry entities represent the definition of the physical shape; the principal entities are points, curves, planes or circles. The topology entities defines the relationship between the object´s geometric parts, these entities should be loops, edges or vertices. The non-geometry entities provide a viewing perspective and appropriate dimension to the drawing; this class includes view, dimensions, text or notation [IGES06].

The IGES file consists of 80 column lines. Lines are grouped into sections and there are five or six sections. IGES data can be represented in either ASCII or binary format. Each line contains the specific data of each section in columns 1-72, an indentifying letter code in column 73 and an ascending sequence number in columns 74-80. In each section the sequence number start in 1 and is incremented in 1 by each line.

Section name	Col. 73 Letter Code
Start	S
Global	G
Directory Entry	D
Parameter Data	P
Terminate	T

Figure 28: IGES file sections [IGES06]

In the following steps an explanation of the different sections is going to be presented, but first in the next figure is presented a full example of the structure of an IGES file.

```
                                                                        S      1
1H,,1H;,4HSLOT,37H$1$DUA2:[IGESLIB.BDRAFT.B2I]SLOT.IGS;,                G      1
17HBravo3 BravoDRAFT,31HBravo3->IGES V3.002 (02-Oct-87),32,38,6,38,15,  G      2
4HSLOT,1.,1,4HINCH,8,0.08,13H871006.192927,1.E-06,6.,                   G      3
31HD. A. Harrod, Tel. 313/995-6333,24HAPPLICON - Ann Arbor, MI,4,0;     G      4
     116       1       0       1       0       0       0       0       1D      1
     116       1       5       1       0                               0D      2
     116       2       0       1       0       0       0       0       1D      3
     116       1       5       1       0                               0D      4
     100       3       0       1       0       0       0       0       1D      5
     100       1       2       1       0                               0D      6
     100       4       0       1       0       0       0       0       1D      7
     100       1       2       1       0                               0D      8
     110       5       0       1       0       0       0       0       1D      9
     110       1       3       1       0                               0D     10
     110       6       0       1       0       0       0       0       1D     11
     110       1       3       1       0                               0D     12
116,0.,0.,0.,0,0,0;                                                    1P      1
116,5.,0.,0.,0,0,0;                                                    3P      2
100,0.,0.,0.,0.,1.,0.,-1.,0,0;                                         5P      3
100,0.,5.,0.,5.,-1.,5.,1.,0,0;                                         7P      4
110,0.,-1.,0.,5.,-1.,0.,0,0;                                           9P      5
110,0.,1.,0.,5.,1.,0.,0,0;                                            11P      6
S      1G      4D      12P      6                                       T      1
```

Figure 29: Example of an IGES file format [IGES06]

4.3.3.1 Flag section

The Flag Section is an optional section that is used to indicate if ASCII format or Binary format.

4.3.3.2 Start section

The Start Section provides information about the file that can be readable for the human. This section is represented with an S in the column 73 and a sequence numbers in the columns 74-80. This section contains the names of the sending or receiving CAD/CAM systems and a short description of the product that is converted.

4.3.3.3 Global section

The required Global Section contains information describing the preprocessor and information needed by postprocessor to handle the file. This section is identified with the letter code G in the column 73 and a sequence number in columns 74-80. The first two global parameters define the parameter delimiter and record delimiter characters if the default values (comma and semicolon) are not used. Some of the most important parameters defined in this section are File Name, Preprocessor Version, Model Space Scale, Unit Flag, Unit Name, Version Flag, etc.

4.3.3.4 Directory Entry section

The Directory Entry Section or DE Section contains a list of all the entities that form the design solid. The DE is fixed in size and contains 20 fields of 8 characters each, in two consecutives 80 characters lines. The purposes of the Directory Entry Section are provided an index for the file and to contain attribute information for each entity. Some of the field in the DE sections may contain either an attribute value or a pointer to other part of the file. As is represented in the next figure, the first and eleventh attributes are the Entity type number, for example 100 for circle or 110 for lines; the second attribute is the parameter data that is a pointer to the entity in the Parameter Data Section. There are more attributes defined that are explained in [IGES06].

1 8	9 16	17 24	25 32	33 40	41 48	49 56	57 64	65 72	73 80
(1) Entity Type Number #	(2) Para-meter Data ⇒	(3) Structure #, ⇒	(4) Line Font Pattern #, ⇒	(5) Level #, ⇒	(6) View 0, ⇒	(7) Transfor-mation Matrix 0, ⇒	(8) Label Display Assoc. 0, ⇒	(9) Status Number #	(10) Sequence Number D #
(11) Entity Type Number #	(12) Line Weight Number #	(13) Color Number #, ⇒	(14) Para-meter Line Count #	(15) Form Number #	(16) Reserved	(17) Reserved	(18) Entity Label	(19) Entity Subscript Number #	(20) Sequence Number D # + 1

Nomenclature:

- (n) - Field number n
- # - Integer
- ⇒ - Pointer
- #, ⇒ - Integer or pointer (pointer is negated)
- 0, ⇒ - Zero or pointer

Figure 30: Attributes of an entity in DE section [IGES06]

In the next table is represented the main basic IGES entities:

ENTITY NUMBER	ENTITY TYPE
110	Line
100	Circular arc
124	Transformation matrix
120	Surface revolution
116	Point
123	Direction
190	Plane surface
502	Vertex list
404	Edge list
508	Loop
510	Face
514	Shell
192	Right circular cylindrical surface

Figure 31: Basic IGES entities [IGES06]

4.3.3.5 Parameter Data section

This section defines the data that correspond to the entity defined in the Directory Entry Section. Each line represents the parameters of one entity, the first value is the entity type and then the parameters of the entity are placed in free format from column 2 to 64. Column 65 shall contain a space character. Columns 66 though 72 shall contain the sequence number of the first line in the DE of this entity. Then finally, column 73 will contain the word P and the columns 74 though 80 will contain the sequence number of this section [IGES06].

1 ... 64	66 ... 72	73 ... 80
Entity type number followed by parameter delimiter followed by parameters separated by parameter delimiters	DE Pointer	P0000001
Parameters separated by parameter delimiters followed by record delimiter	Pointer	P0000002
⋮	⋮	⋮

Note: The DE pointer is the sequence number of the first Directory Entry line for this entity

Figure 32: Parameter Data Section [IGES06]

4.3.3.6 Terminate section

The Terminate Section consists in only one line that resumes the amount of lines that is formed the rest of the other sections. This section shall be the last sequence line of the file. The Terminate Section has a T in the column 73 and a number 1 in the columns 74-80. In the next table we can check how the information of the amount of lines of the others sections are situated in the columns of the line of the Terminate Section [IGES06].

Field	Columns	Section
1	1 - 8	Start
2	9 - 16	Global
3	17-24	Directory Entry
4	25-32	Parameter Data
5-9	33-72	(not used)
10	73-80	Terminate

Figure 33: Structure of the Terminate Section [IGES06]

1 8	9 16	17 24	25 32	33 40 41 48 49 56 57 64 65 72	73 80
S0000020	G0000003	D0000500	P0000261	Not Used	T0000001

Figure 34: Terminate Section example [IGES06]

Chapter 5

SIMILARITY MEASURES

5.1 Introduction

According to the Merriam-Webster Dictionary [MW12] the adjective similar is defined as “having characteristics in common: strictly comparable” or “not differing in shape but only in size or position”.

In all the field of shape retrieval information, the similarity measures play an important roll. Nowadays the industry companies realize that they can save costs and time using retrieval knowledge; in this Thesis is presented one way based on design with CAD software and classification of parts. When there is a database of classified parts, it is simple to find similarities between two parts, and then when a part is going to be designed, it is possible to search a similar one to use retrieval information and facilitate the design of this part. The way to find similarities between parts is with a similarity measure that is a function for quantifying the similarity between two models [SJ99].

A shape similarity measure useful for shape retrieval information in shape databases should be in accord with our visual perception. A similarity measure should follow the next requirements to get this basic property:

- A shape similarity measure should permit recognition of perceptually similar objects that are not mathematically identical.
- It should abstract from distortions.
- It should respect significant visual parts of objects.
- It should not depend on scale, orientation and position of objects.
- It is universal in the sense that it allows us to identify or distinguish objects of arbitrary shapes, i.e., no restrictions on shapes are assumed.

Normally the process to work with shape retrieval information starts with first to convert the 3D shape model in a shape signature, graph, vector, algorithm, etc. and then the similarity between two shape signatures is reflected in the distance between corresponding points of each.

5.2 Properties

The similarity is measured with distance functions. This distance functions must follow some properties that are explained in this section. It is normal that a distance function don´t follow all the properties, sometime some properties are desirable for some distance functions and some others not, sometimes even the combination of properties could be contradictory. To explain the properties, it is presented three shapes A, B and

C in a space S with the function $d: S \times S \rightarrow \mathbb{R}$. Here are presented the most important properties [VH05]:

1. No negativity: $d(A,B) \geq 0$.
2. Identity: $d(A,A) = 0$.
3. Uniqueness: $d(A,B) = 0$ implies A = B.
4. Strong triangle inequality: $d(A,B) + d(A,C) \geq d(B,C)$.
5. Triangle inequality: $d(A,B) + d(B,C) \geq d(A,C)$.
6. Relaxed triangle inequality: $c\big(d(A,B) + d(B,C)\big) \geq d(A,C)$, for some constant $c \geq 1$.
7. Symmetry: $d(A,B) = d(B,A)$.
8. Invariance: d is invariant under a chosen group of transformations G if for all $g \in G$, $d\big(g(A) + g(B)\big) = d(A,B)$.

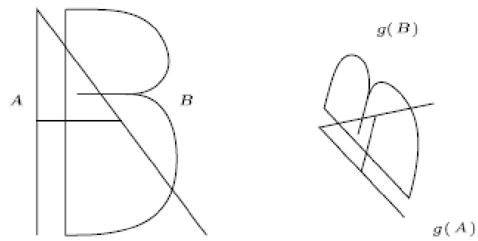


Figure 35: Invariance property

Finally, in one hand it is possible to find more properties that are focused in comparison of image or other fields, these ones could be about robustness, like perturbation robustness, crack robustness, blur robustness or noise robustness. These properties are useful to be robust against the effects of discretization. In the other hand other properties could be distributivity, endlessness, discernment, sensitivity, proporcionality and monotonicity [VH05].

5.3 Types of similarity measures

In this section a number of the most important similarity measures with their respective distance function that usually are used to compare graphs, image, polygons or vectors of digits are defined [Vert06].

5.3.1 Discrete metric

The discrete metric is the easy way to compare two shapes, the models should have a shape signature, this means for example one vector of numbers or one graph and the comparison between us is just done with the following distance function [PW12]:

$$d(A,B) = \begin{Bmatrix} 0 \ \ if\ A\ equal\ B \\ 1 \ \ in\ other\ case \end{Bmatrix}$$

In this function the terms A and B mean the shape signature of the models. The main problem of this metric is that when two shape signatures are difference it is only possible to obtain the maximum distance between them, this metric only gives the possibility of two results, totally different or totally equal, so the most disadvantage is that it don't have accuracy for shape signatures.

5.3.2 Minkowski distance L_p

Minkowski distance is actually a class of distance measurement methods since it looks and works differently for different values of the p coefficient. There are lots of similarity measures that use the Minkowski distance as a distance function between two points [Vand04]. It can be call L_p norm too. For two points $x, y\ in\ \mathbb{R}^k$, the Minkowski distance is defined as:

$$L_p = \left(\sum_{i=0}^{k} |x_i - y_i|^p \right)^{1/p}$$

P must be bigger than zero; so, for $p = 1$, the distance metric is called *Manhattan distance* (also known as city block distance, taxicab geometry or rectilinear distance); this distance was defined with this name because remains the pattern of the streets in Manhattan, all the streets are perpendiculars and parallels and the distance between two points is the same distance than a car cover going from one point to the other one. For *p=2,* this representation define the *Euclidean distance* that it is the normal distance between two points, a straight-line distance; for given two point *x, y* and one distance *d*, one of the points will be the center of a circle, the distance will be the radius and the other point will be any point of the circle. In the limit case, for $p = \infty$, it is obtained the *Chebyshev distance,* also known as Maximum distance, that it is defined as, $max|x_i - y_i|$.

In the next figures is presented a comparison between different Minkowski distances. The figure 36 shows a comparison between the Manhattan distance, lines red, blue and yellow, and the Euclidean distance, line green. The figure 37 presents a comparison between the main values that the *p* coefficient can take.

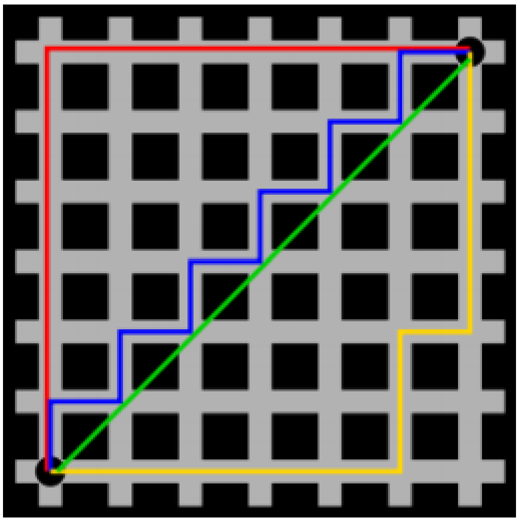


Figure 36: Comparison between Manhattan distance and Euclidean distance

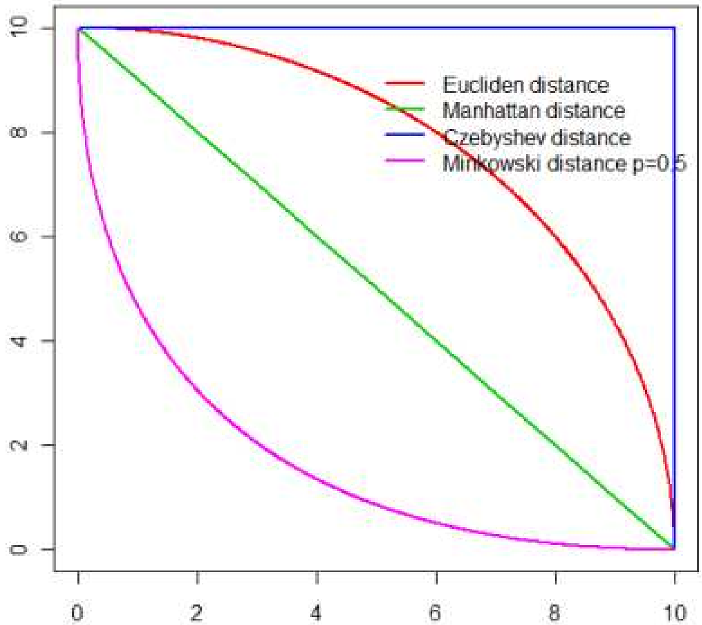


Figure 37: Comparison of the difference Minkowski distances

5.3.3 Hausdorff distance

The Hausdorff distance is a distance defined between two point sets; this metric calculate the longest distance between two point sets. This distance tolerates errors in the positions of the points, as well as the presence of extra points and missing points. This is important in cases of stereo or images matching that are signals with occlusion and noise [HKR93].

The Hausdorff distance between two point sets M *and* I is defined as:

$$H(M, I) = max(h(M, I), h(I, M))$$

Where

$$h(M,I) = \max_{m \in M} \min_{i \in I} \|m - i\|$$

Where $\|.\|$ is some norm in the plane, it should be L_2.

$h(M,I)$ is computed by taking each point of *M*, computing the distance from that point to the nearest point of *I*, and reporting the largest distance; $h(I,M)$ is computed similarly; and the largest of these two distance is $H(M,I)$. $h(M,I)$ is the directed distance from *M* to *I* and it is small exactly when every point of *M* is near some point of *I*. Similarity, $h(I,M)$ is small when every point of *I* is near some point of *M*, and the undirected distance $H(M,I)$ is small when both of these are true.

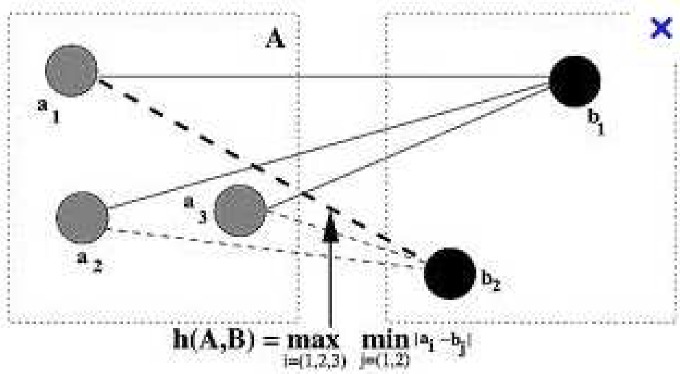


Figure 38: Example of Hausdorff distance

5.3.4 Bottleneck distance

Given two point sets *A* and *B* and a distance *d(a,b)* between two points in each group set, the Bottleneck distance *F(A,B)* [EI96] is the minimum over all 1-1 correspondences f between *A* and *B* of the maximum distance *d(a, f(a)).* For de distance between two points we can use a Minkowski distance.

$$F(A,B) = \min_{f \in F} \max_{a \in A} d\big(a, f(a)\big)$$

If *d(a,b)* is the Euclidean distance, the Bottleneck distance between A and B can be computed in time $O(n^{1.5} \log n)$. It is computed using a technique called parametric search.

There is another possibility to calculate the distance between two points which is to compute an approximation Bottleneck function to the real one. The approximate matching can be computed in time $O(n^{1.5} \log n)$ and it will be defined as:

$$d'(A,B) < (1 + \epsilon) d(A,B)$$

5.3.5 Turning function distance

The cumulative angle function or turning function, $\Theta_A(s)$ of a polygon *A* gives the angle between the counterclockwise tangent and the x-axis as a function of the arc length s. The turning function increases with left hand turns and decrease with right hand turns. This function is invariant under translation of the polyline. The turning function is a piecewise constant function, increasing or decreasing at the vertices, and constant between two consecutive vertices [Mitc10].

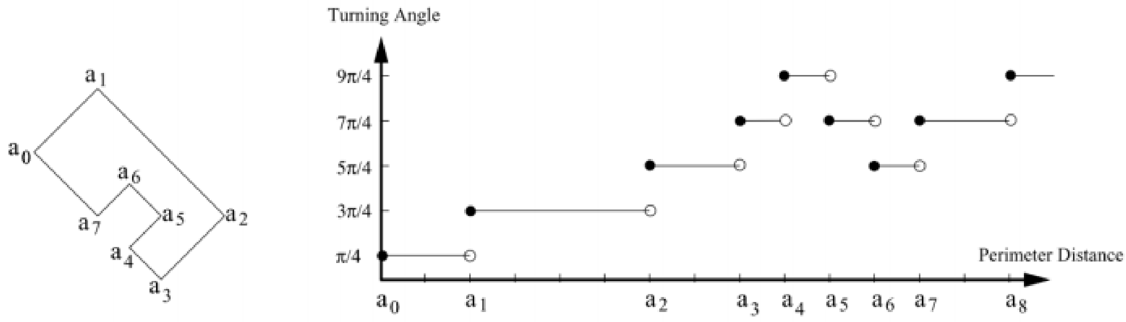


Figure 39: The turning function

The turning function distance is used to match polygons. Given two polygons A and B with turning functions $\Theta_A(s)$ and $\Theta_B(s)$, the definition of the L_p distance between the two functions is [Zhan00]:

$$d(A,B) = \left(\int_0^1 |\Theta_A(s) - \Theta_B(s)|^p ds\right)^{1/p}$$

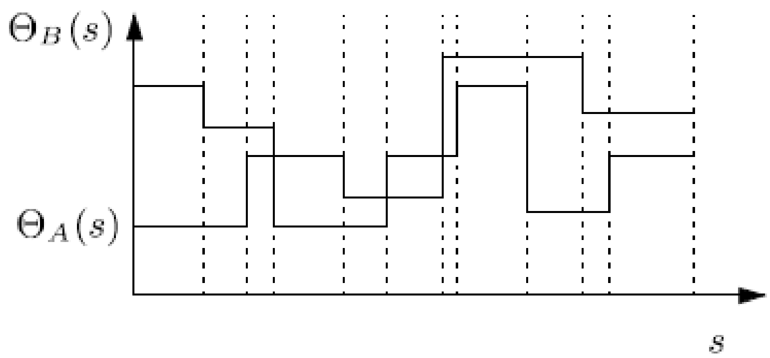


Figure 40: Comparison between two turning functions

5.3.6 Signature function distance

The signature function is a less discriminative function than the turning one. The signature function value is the arc length of the curve to the left or on the tangent line at that point. It is invariant under translation, rotation and scaling. For convex curve, the signature function has value one everywhere. For matching two signatures functions it is used “time warps”, pairing elements of one function with elements of the other one [Orou85].

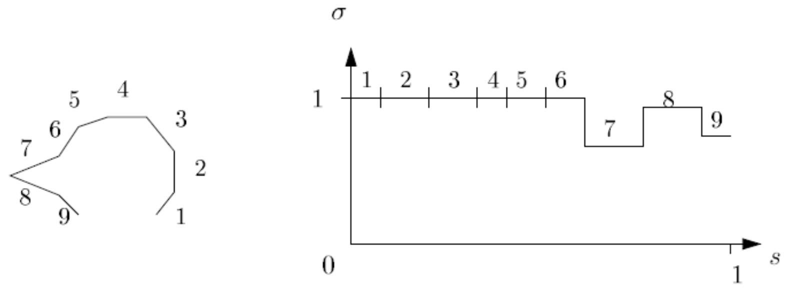


Figure 41: Signature function

5.3.7 Fréchet distance

The Fréchet distance is a measure of similarity between curves that takes into account the location and ordering of the points along the curves. For curves, it is often better than the Hausdorff distance [EM94]. A simple example to understand better this distance between two curves is explained; suppose a man walking his dog with a belt, the man is walking in one curve and the dog is walking in another one, they can adjust their speeds but they are not allow to move backwards. So the Fréchet distance of this two curves that are generated by the man and the dog is done by the belt of the dog.

In the next figure, two curves are illustrated; it is possible to see the different between the Hausdorff distance that aims to calculate the minimum distance between two points, each one of each curve and the Fréchet distance that takes into account the locations of the points in the curve.

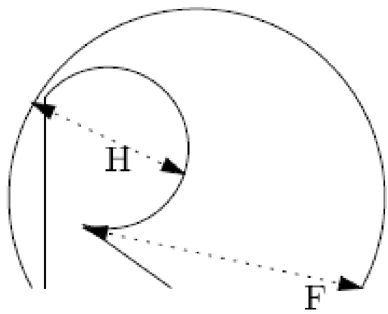


Figure 42: Comparison between Hausdorff distance (H) and Fréchet distance (F)

Given two curves, the Fréchet distance is defined as follows:

$$F_{r(P,Q)} = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\| P\big(\alpha(t)\big) - Q\big(\beta(t)\big) \right\|$$

Where $P, Q: [0,1] \rightarrow \mathbb{R}^2$ are parameterizations of the two curves and $\alpha, \beta: [0,1] \rightarrow [0,1]$ are the range over all continuous and monotone increasing functions.

5.4 Similarity measure for the Opitz Code System

One of the major objectives of this Master Thesis is development of a distance function to compare Opitz Codes. To achieve this objective at the first step, many different distance functions has been studied and evaluated. As it was mentioned in the previous chapter, distance functions are developed usually to compare graphs, vectors, digits, images, points, lines, etc.; the main characteristic of them is that they can be continuous or discrete. In this project, the first five digit of the Opitz Code System, called Opitz Form Code, are implemented. So it is clear that the developed system works with a discrete vector of numbers. Starting with this premise, the distance function for Opitz Code is going to be an improvement of the discrete metric.

The discrete metric just does a total comparison between two shape signatures and gives only two results; total equality if the shape signatures are the same or any equality if there is some difference between them. This means that this distance function is not able to characterized different types of dissimilarities. Due to this disadvantage of the discrete metric, a variation of the distance metric that aims to get a distance function capable of differentiate dissimilarities was developed. To obtain this, the distance function is going to compare the digits of the code in an individual way to fix an appropriate algorithm that calculates different kinds of dissimilarity. In the following paragraphs, this algorithm is going to be explained.

The algorithm starts with the comparison of the first digit; the first digit classifies the non-rotational part in flat parts, long parts or cubic parts; if the comparison of the first digit of the two codes is different means that the parts are totally different, there is no similarity. In the next figure it is possible to see the difference between these parts.

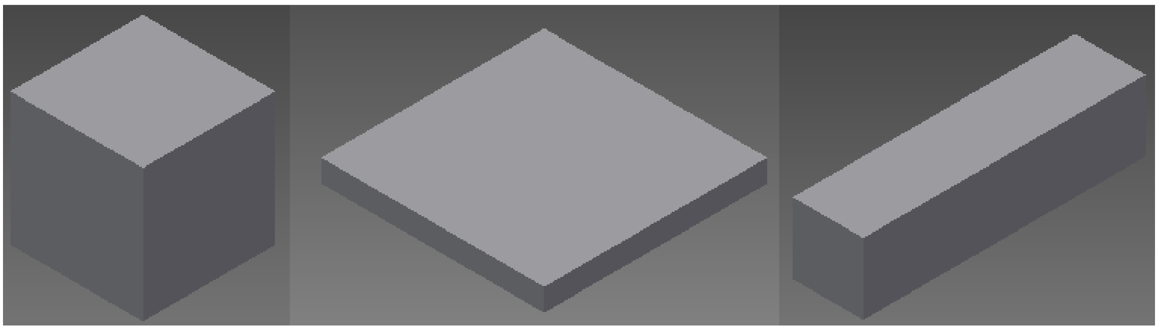


Figure 43: Three different possibilities of the first digit of the Opitz Code

The next step of the algorithm is when the comparison of the first digits of the codes is equal; for example, both are flat parts, and then start being some similarity. Following that the first digit of the codes are equals, the algorithm is going to focus in the second digit that classifies the main form of the part. When the algorithm detects that the first digit is equal in the two codes and the second digit of the codes are different means that there is a similarity of twenty percent. The next figure shows an example of this similarity with flat parts.

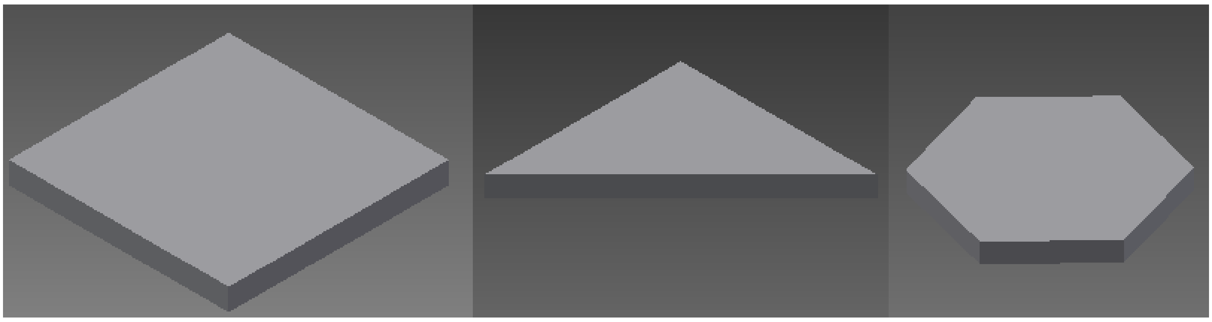


Figure 44: Parts with twenty percent of similarity

Following last step, the next possibility is that the digit one and two in the comparison of the codes are equals, for example a flat part with rectangular main form. Then the algorithm focuses in the fourth digit of the codes that are being compared; this digit signifies the machining of the surface of the part. If both codes have different values in the digit four and the first and the second ones are equal, the distance function gives a similarity of forty percent. In the following figure is presented an example of this similarity.

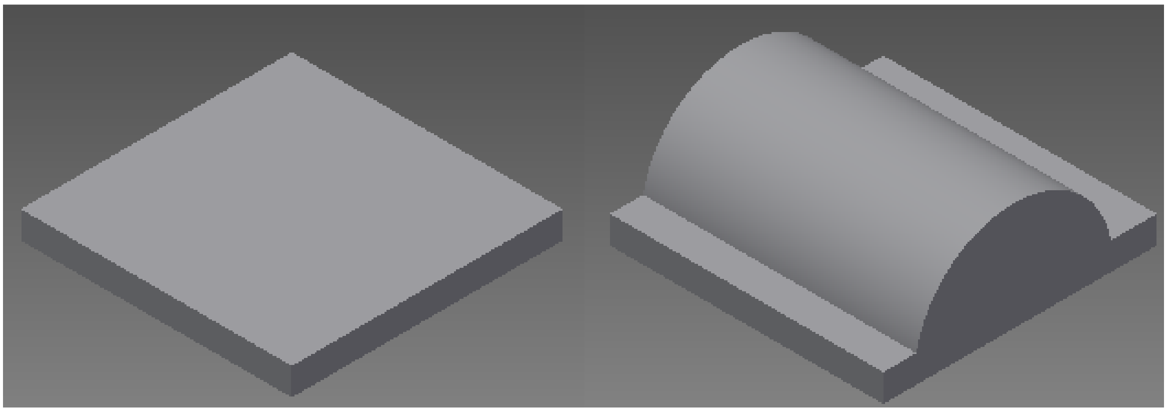


Figure 45: Parts with forty percent of similarity

The next possibility in the detection of similarity is that the first, the second and fourth digits of the two codes are equals. Then the distance function is going to focus in the third digit of the codes. This digit classifies the main holes in a part. So if the distance function notifies the digit one, two and four equals in the codes, and a different value comparing the digit three of the codes, it means that the parts have a similarity of sixty percent. In the next figure is presented an example of sixty percent similarity.

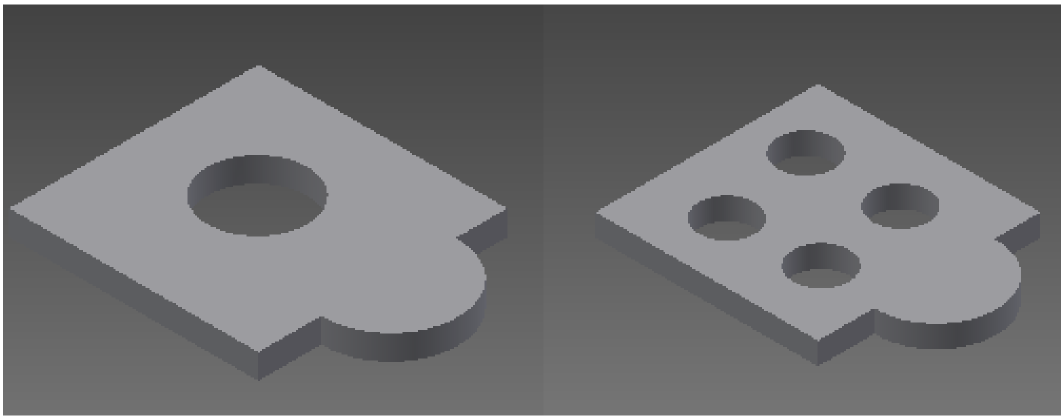


Figure 46: Parts with sixty percent of similarity

On the other hand, if the third digits of the two codes are equal means that the first four digits of the codes are the same. So, the distance function will focus in the fifth digit. The digit five classifies the auxiliary holes of the parts. When the similarity measure compares the first four digits and gives the result that they are equals and the comparison of the last one is different signify that the parts have a similarity of eighty percent. In the next figure is presented an example of this similarity.

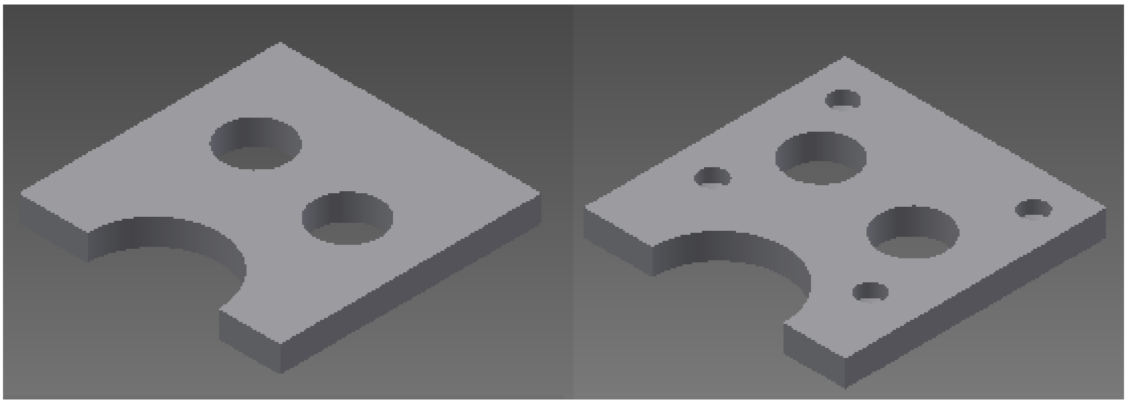


Figure 47: Parts with eighty percent of similarity

Finally, the last possibility of comparison is the complete similarity. This situation appears when the five digits of the two codes are equals. The figure 48 shows an example.

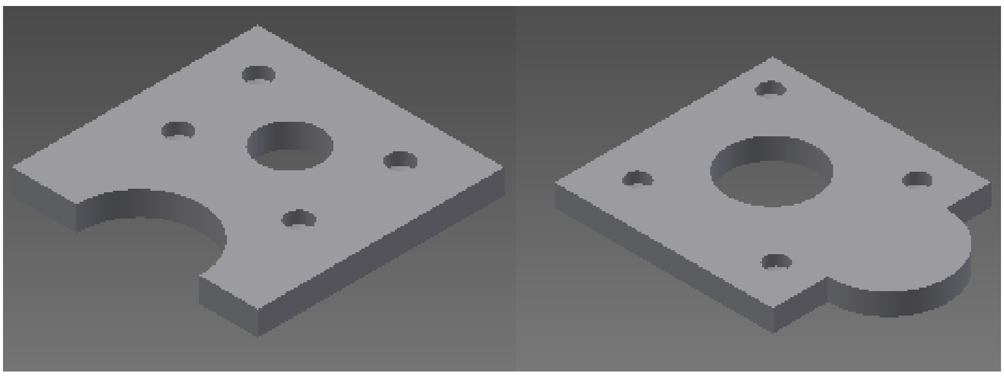


Figure 48: Complete similarity between parts

Chapter 6

THE DEVELOPED TECHNIQUE

6.1 Introduction

In this chapter, the followed methodology for the developed system is presented and explained. The main objective of this Thesis is the implementation of a classification and comparison system for non-rotational parts. For the classification, a shape signature method based on feature recognition has been developed; this shape signature is a five digit vector for de Opitz Code System that characterizes the features of a 3D model. For the comparison, a distance function of similarity measure has been implemented to compare the shape signatures of two codes.

A part is designed with CAD software and it is represented as a solid model by using Constructive Solid Geometry (CSG) technique as a design tool. The CAD software generates an IGES file of the designed part; the IGES file format represents the boundary geometrical information of the 3D model (arcs, lines, points, etc.). With a feature recognition program, the boundary information in the IGES file is extracted and managed to identify features of the part. The Opitz Code system is used to classify these features and generate the shape signature, a vector of five digits, and finally, the shape signature of a part is saved in a fold creating a database of codes. One of the most important achieves is to develop a system that is able to detect part features as much as possible. The implemented feature recognition program recognizes the following features:

- Non-rotational form; flat part, long part and cubic part.
- Main form of the part; rectangular form, triangular form, circular and rectangular form, angularly form, etc.
- Main bores; one bore, two bores, several bores in same direction and several bores in different directions.
- Machining of surface; plane surface and curve surface.
- Auxiliary holes; auxiliary holes in one direction and auxiliary holes in different directions.

The comparison part of the implemented system takes two codes of the database and makes a comparison with a developed distance function for Opitz Code. The similarity results between the codes can be zero percent, twenty percent, forty percent, sixty percent, eighty percent or hundred percent of similarity. Chapter 5 explains the functionality of this implemented similarity measure.

In the next figure a flowchart with all the steps followed in the developed system is presented.

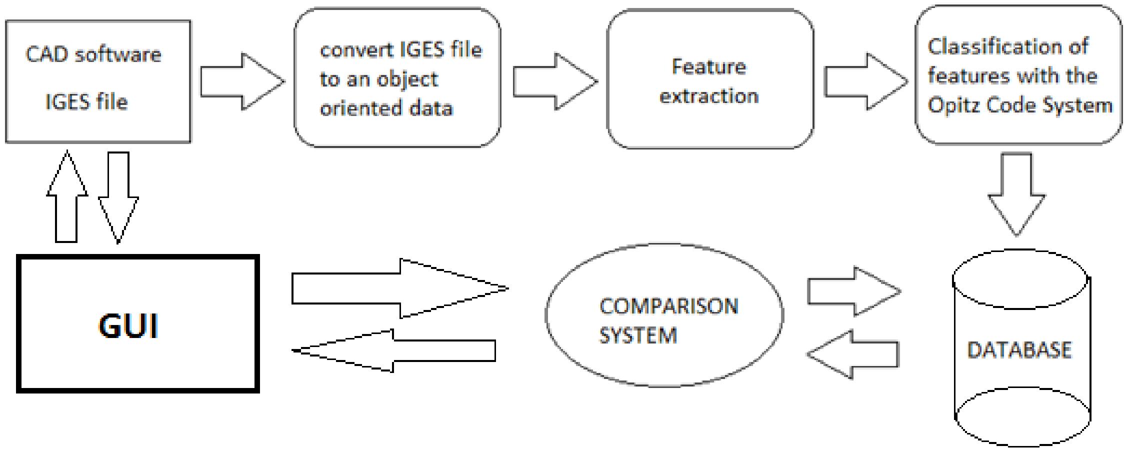


Figure 49: Flowchart implemented procedure to retrieval of similar shapes

Based on the presented flowchart in Figure 49, in the next sections, all the process involved in Shape Similarity Retrieval with a request model in CAD software is explained.

6.2 Producing IGES file from a CAD file

As it was mentioned, the beginning of the implemented technique starts with designing 3D models with CAD software and generating the IGES file. IGES is a standard file format that defines the boundary information of a 3D model drawn with CAD software. The lower level of the format is the entities; the entities of the 3D model are defined in the Directory entry section of the IGES file and the values of each entity are inside the Parameter Data section of the file.

IGES file format defines a big amount of entities to characterize all the boundary information. In this research, features include lines, arcs, holes, curve surface, auxiliary holes, etc. In this thesis, the developed system works only with line entities, circular arc entities and transformation matrix entities to detect features of the part.

In the next figures a flat part with a main hole is presented as an example. The first figure shows the CAD software representation of the 3D model and the second figure presents the IGES file of this representation. In this file, there are twelve lines of the flat part in blue, two circular arcs of the main hole in green and two transformation matrixes in red.

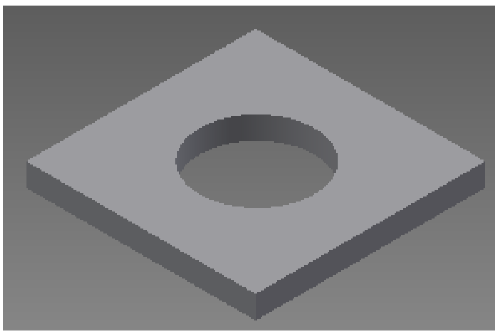


Figure 50: Flat part with main hole

```
                                                                        S      1
,,30Hflat_part_rectangular_mainhole,27HNo Iges File Name specified,22HAu  G      1
todesk Inventor 2013,37HAutoDesk Inventor Iges Exporter R2013,32,38,6,    G      2
99,15,30Hflat_part_rectangular_mainhole,1.0D0,1,2HIN,1,.08D0,15H20121204  G      3
.160805,3.93700787401575D-4,10000.0D0,6HJavier,,11,0,,;                   G      4
406     1     0     0     1     0     0     000000000                     D      1
406     0     0     1     3                 0                             D      2
100     2     0     0     0     0     5     000000000                     D      3
100     0     0     2     0                 0                             D      4
124     4     0     0     0     0     0     000000000                     D      5
124     0     0     2     0                 0                             D      6
100     6     0     0     0     0     9     000000000                     D      7
100     0     0     1     0                 0                             D      8
124     7     0     0     0     0     0     000000000                     D      9
124     0     0     2     0                 0                             D     10
110     9     0     0     0     0     0     000000000                     D     11
110     0     0     1     0                 0                             D     12
110    10     0     0     0     0     0     000000000                     D     13
110     0     0     1     0                 0                             D     14
110    11     0     0     0     0     0     000000000                     D     15
110     0     0     1     0                 0                             D     16
110    12     0     0     0     0     0     000000000                     D     17
110     0     0     1     0                 0                             D     18
110    13     0     0     0     0     0     000000000                     D     19
110     0     0     1     0                 0                             D     20
110    14     0     0     0     0     0     000000000                     D     21
110     0     0     1     0                 0                             D     22
110    15     0     0     0     0     0     000000000                     D     23
110     0     0     1     0                 0                             D     24
110    16     0     0     0     0     0     000000000                     D     25
110     0     0     1     0                 0                             D     26
110    17     0     0     0     0     0     000000000                     D     27
110     0     0     1     0                 0                             D     28
110    18     0     0     0     0     0     000000000                     D     29
110     0     0     1     0                 0                             D     30
110    19     0     0     0     0     0     000000000                     D     31
110     0     0     1     0                 0                             D     32
110    20     0     0     0     0     0     000000000                     D     33
110     0     0     1     0                 0                             D     34
406,2,0,30Hflat_part_rectangular_mainhole;                               1P      1
100,0.0D0,0.0D0,0.0D0,-.25D0,3.06161699786838D-17,-.25D0,                3P      2
3.06161699786838D-17;                                                    3P      3
124,-1.0D0,0.0D0,0.0D0,.5024984D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,         5P      4
1.0D0,0.0D0,.4967035D0;                                                  5P      5
100,0.0D0,0.0D0,0.0D0,.25D0,0.0D0,.25D0,0.0D0;                           7P      6
124,-1.0D0,0.0D0,0.0D0,.5024984D0,0.0D0,0.0D0,-1.0D0,.1D0,0.0D0,         9P      7
-1.0D0,0.0D0,.4967035D0;                                                 9P      8
110,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,0.0D0;                                11P      9
```

```
110,0.0D0,0.0D0,1.0D0,0.0D0,.1D0,1.0D0;                        13P    10
110,0.0D0,.1D0,0.0D0,0.0D0,.1D0,1.0D0;                         15P    11
110,0.0D0,0.0D0,0.0D0,0.0D0,.1D0,0.0D0;                        17P    12
110,0.0D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0;                       19P    13
110,1.0D0,.1D0,0.0D0,0.0D0,.1D0,0.0D0;                         21P    14
110,1.0D0,0.0D0,0.0D0,1.0D0,.1D0,0.0D0;                        23P    15
110,1.0D0,0.0D0,0.0D0,1.0D0,0.0D0,1.0D0;                       25P    16
110,1.0D0,.1D0,1.0D0,1.0D0,.1D0,0.0D0;                         27P    17
110,1.0D0,0.0D0,1.0D0,1.0D0,.1D0,1.0D0;                        29P    18
110,1.0D0,0.0D0,1.0D0,0.0D0,0.0D0,1.0D0;                       31P    19
110,0.0D0,.1D0,1.0D0,1.0D0,.1D0,1.0D0;                         33P    20
S     1G     4D     34P     20                                   T     1
```

Figure 51: IGES file format of a flat part with main hole

According to the entities that the developed system uses to find features and classify parts, in the appendix A is presented a literature review of the line entity, the circular arc entity and the transformation matrix entity.

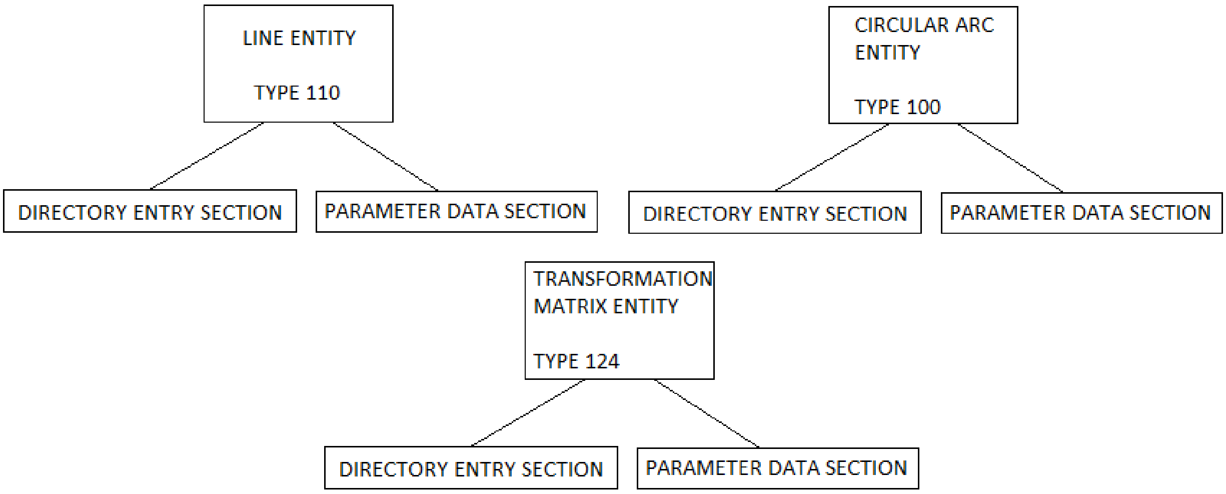


Figure 52: Flowchart of the used entities

6.3 Conversion of an IGES file to an object oriented data structure

The IGES file is generated with CAD software. The next step is to develop a method to extract the important information of the file that is used to detect the features of the part. In this case, a function implemented in Matlab is the solution to extract the information of each entity. The information and values of the entities in the IGES file is saved in a cell array structure called *ParameterData*.

This Matlab function has one input, the IGES file. The file is parsed by this function; first the function detects the entities in the Directory Entry section, then this section readdresses each entity to the Parameter Data section, where the information of the entities is located, and finally, the function saves the values and the information of each entity of the IGES file in the *ParameterData*. Hence, the output of this function is this cell array to which is joined three arrays more; the *EntityType* array that gives the information of all the different entities that the IGES file has, the *numEntityType* array that contain the amount of each entities and the *unknownEntityType* that store entities unknown for the Matlab function. In the next figure an example of a parsed IGES file is shown.

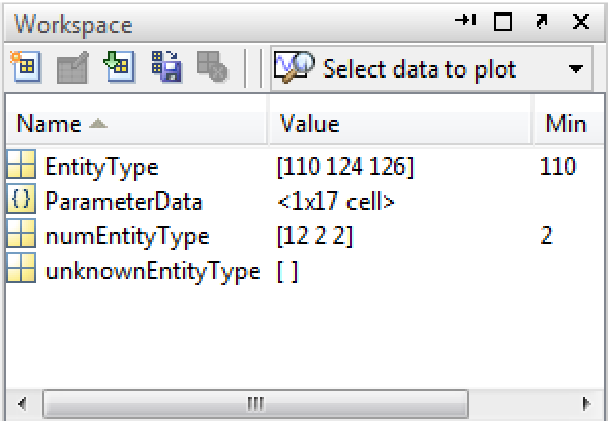


Figure 53: Example of the outputs of an IGES file

The outputs of the figure 53 belong to a cube part with a main hole (Figure 54). The boundary information of this 3D model consists in twelve lines that delimit the cube, two circular arcs that define the main hole and two transformation matrixes to fix the coordinate system of the arcs. The *EntityType* array contains the numbers 110, 124 and 126, the values of the line, transformation matrix and circular arc respectively; the *numEntityType* array reports that there are twelve lines, two circular arcs and two transformation matrixes; the *unknownEntityType* array is empty due to the program is able to identify all the entities of the IGES file and the *ParameterData* contain all the information of each entities mentioned. In the next figure, the entities of the part are presented in yellow to facilitate the explanation.

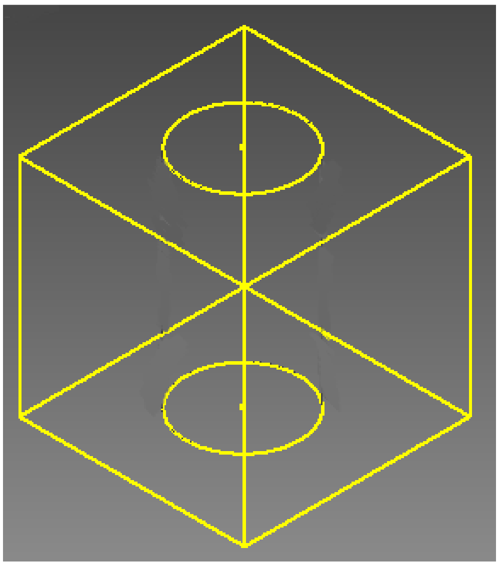


Figure 54: Entities in yellow of the 3D model

6.4 Main algorithm for the feature extraction

Once the information of the entities of the IGES file are extracted and stored in the *ParameterData*, the implementation of algorithms that are able to characterize and extract features of the 3D model is required. The complete knowledge of the Opitz Code system is needed to develop these algorithms according to the features that are managed by the classification system.

6.4.1 First steps

The first digit of the Opitz Code system characterizes the size and the class of the part, the digit has the possibility to classify rotational and non-rotational parts. This thesis is focused on non-rotational ones, so this first digit recognizes three different classes of parts; flat part, long part and cubic part. In Figure 55, an example of the three different classes of non-rotational parts is presented.

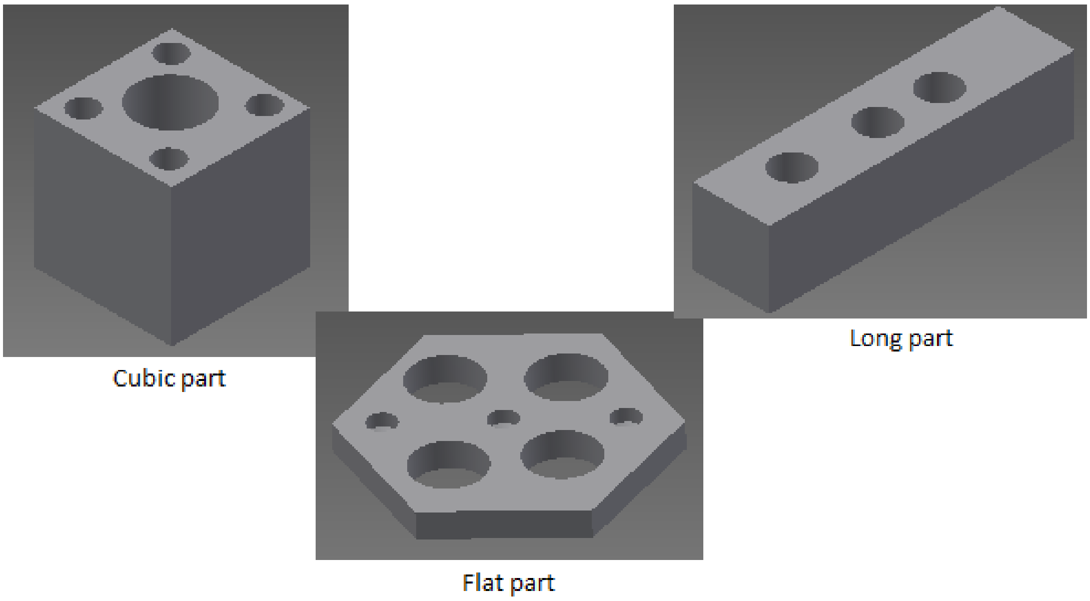


Figure 55: Example of the three different classes of non-rotational parts

The performance of this digit is done by characterizing the length (A), width (B) and height (C) of the designed model. It is important to mention that the value A always has to be the largest, the value B is the second largest one and the value C is the smallest one; this three values follow this simple relationship, $A \geq B \geq C$. In Figure 56, an example of a properly characterization is presented.

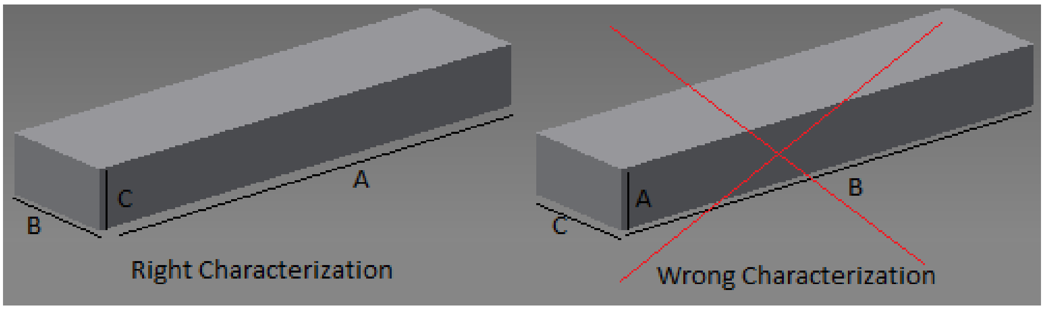


Figure 56: Characterization of the part

The recognition of the three different classes is done by simple relationships between the values A, B and C. In the next table is shown the premises that the developed system follows to recognize flat, long and cubic parts.

Flat part	Long part	Cubic part
$\boldsymbol{A/B \leq 3}$ $\boldsymbol{A/C \geq 4}$	$A/B > 3$	$A/B \leq 3$ $A/B < 4$

Figure 57: Premises to recognize class parts

There are some rules that the developed system follows; when a part is designed by the CAD software. The parts that are designed have to be positioned in the coordinate axis in the properly way to be analyzed. It is obligatory to design the part with the length (A) positioned in the axis *Z*, the width (B) in the axis *X* and the height (C) in the axis *Y*. In the next figures some examples are presented to leave clear how the 3D models have to be positioned when they are designed in the CAD software.

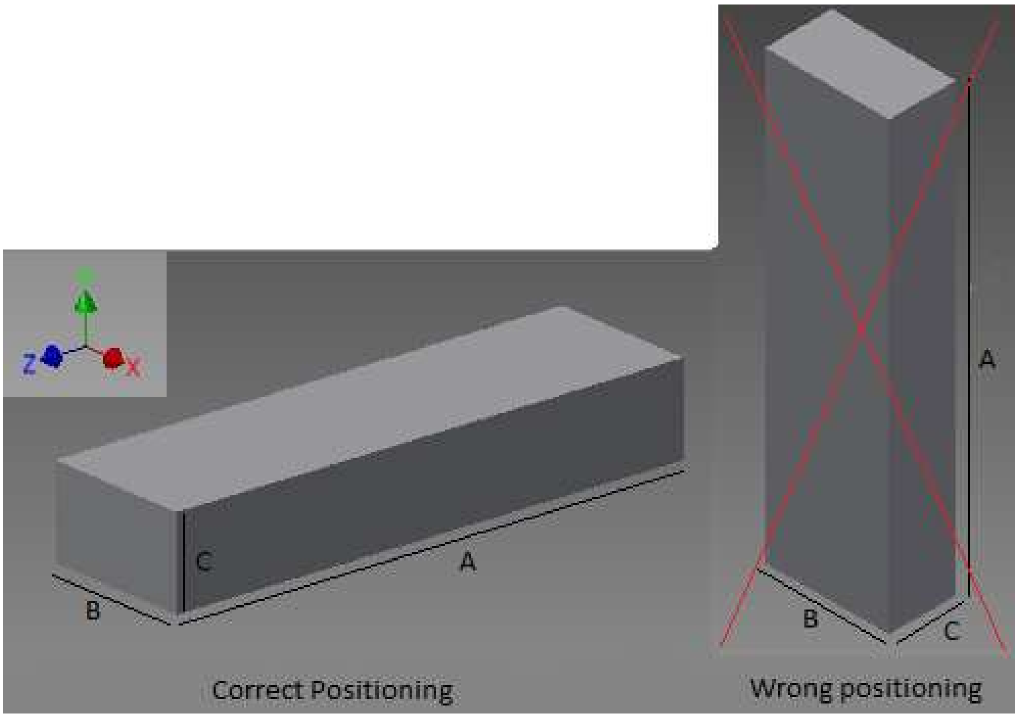


Figure 58: Example of positioning

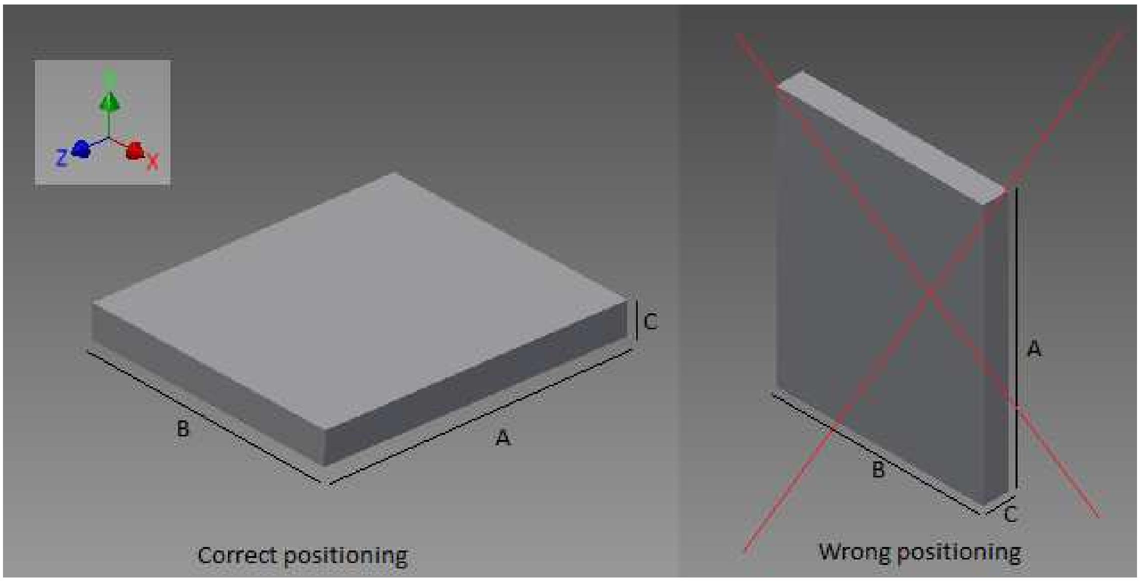


Figure 59: Example of positioning

According to these premises about the characterization and the position of the designed parts, in the next step the main algorithm of the developed program is presented; the algorithm works with lines and circular arcs. This algorithm helps and facilitates information to the rest of the methods for the feature extraction of the parts.

6.4.2 Main algorithm

The main algorithm is designed to aid the rest of the system code to recognize and extract features. It works with lines and circular arcs and the functionality is to organize the lines and the arcs along the axis *X*, *Y* and *Z*.

6.4.2.1Lines

The entity of a line is registered in the *ParameterData* and characterized by its start point and finish point. This is all the information that the developed program has to work with a line. Otherwise, an algorithm that works with lines has to be designed to characterize properly the part. The best idea is to organize the lines along each axis and these organized lines have to be positioned in perpendicular planes to each axis. That means that along the axis *Z*, the algorithm groups lines that are situated in planes *XY*; for the axis *X*, the system groups lines situated in planes *YZ* and for the axis *Y*, the system groups lines situated in planes *XZ*. A better explanation of this idea is presented with a clear example.

A simple cube part is analyzed as an example to facilitate the understanding of the performance of the algorithm. In the Figure 60, the extraction of the eight perpendicular lines along the axis *Z* is shown. Below the figure is explained the functionality of the algorithm.

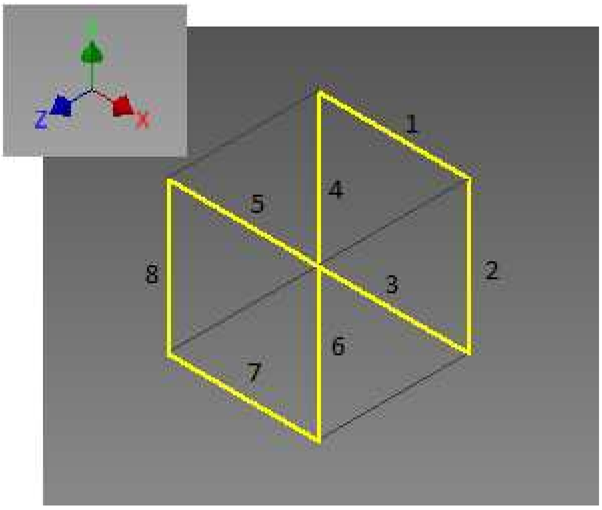


Figure 60: Lines perpendicular to the axis Z

1. Extraction of all the lines and their values of the *ParameterData* array.
2. Search lines with the same value of *Z* in the start and terminate points; this means that the line is perpendicular to the axis *Z* and it is inside a plane *XY*.
3. Group lines with the same value *Z* in the start and end point along the axis *Z*; in this example there are two groups of lines.
4. Repeat the steps 2 and 3 for the axis *X*.
5. Repeat the steps 2 and 3 for the axis *Y*.

As it is shown in the Figure 60, the developed algorithm returns two groups of lines along the axis *Z*, one group with lines 1, 2, 3 and 4 in the position *Z=0* and the other one with lines 5, 6, 7 and 8 in the position *Z=1*. To show clear the process of the algorithm, the next figures present the recognition of lines along axis *X* and *Y*.

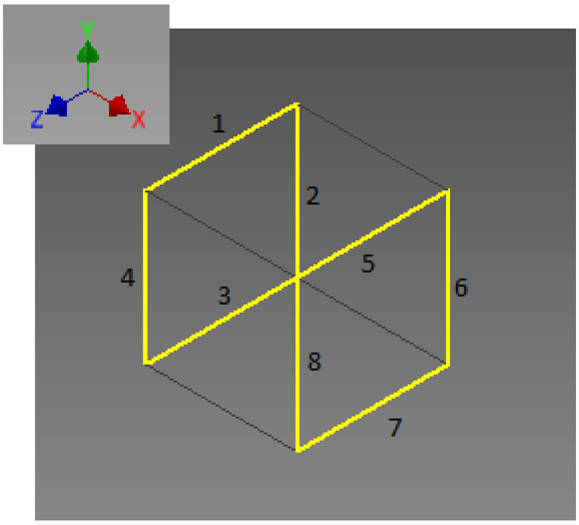


Figure 61: Lines perpendicular to the axis X

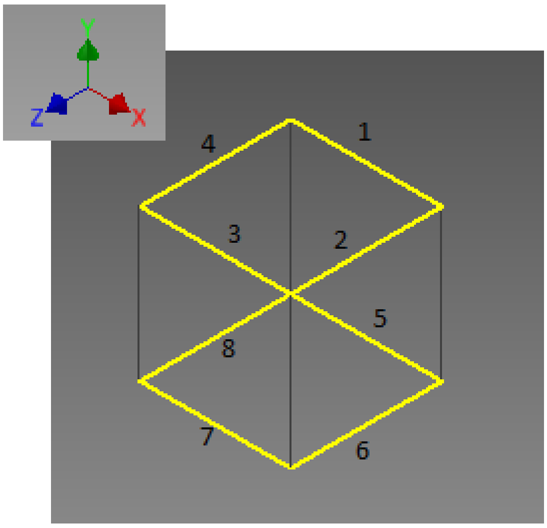


Figure 62: Lines perpendicular to the axis Y

As it is shown in Figure 61, the algorithm detects eight lines that are perpendicular to the axis X and this lines are grouped in two groups; in Z=0 lines 1, 2, 3 and 4 and in Z=1 lines 5, 6, 7 and 8. In the figure 62 it is presented the analogue for the axis Y.

6.4.2.2 Circular arcs

The circular arcs are defined with the center point, the initial point and the terminate point of the arc and these values are stored in the *ParameterData*. The main algorithm extracts all the circular arcs stored in the *ParameterData* and organizes them along each axis. The returned information by the algorithm is an array with the arcs along the axis Z that are in planes XY, another array with arcs along the axis X that are in planes YZ and one last array with arcs that are along the axis Y and in planes XZ. In Figure 63 an example is presented, the green arcs are the ones that belong to the axis Y, the blue ones belong to the axis Z and the red ones belong to the axis X. the algorithm group the arcs in different arrays depending the axis that belong.

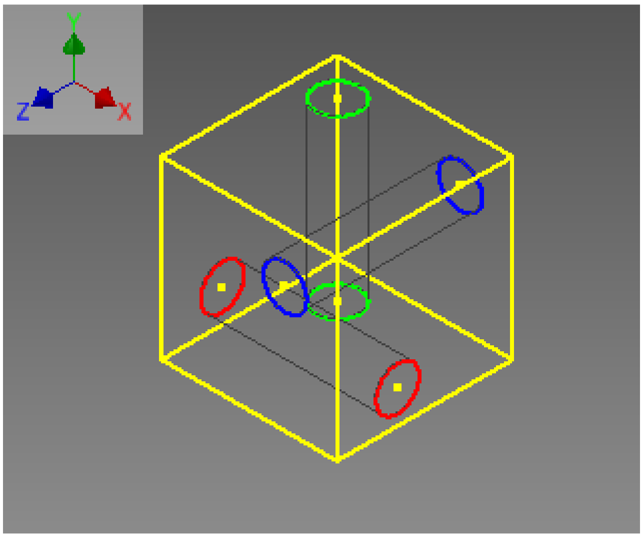


Figure 63: Example with arcs

6.5 Opitz code extraction

One of the objectives of the developed system is to generate an Opitz Code for each part that is designed. As it is known, each digit of the Opitz Code characterizes a type of features; the first digit characterizes the part class, the second digit the main shape, the third digit the main bores, the forth digit the machining surface and the fifth digit the auxiliary holes. Once the main algorithm is developed and tested, it is going to be implemented the rest of the program to generate Opitz Codes as a shape signature. The next steps explain the algorithms that the system follows to recognize the features of each digit.

6.5.1 Digit 1, part class

The digit one acquires the value 6 when the 3D model is a flat part; the value 7 if the model is a long part or the value 8 if the model is a cubic part. The developed system calls the main algorithm and gets all the lines along each axis and grouped in perpendicular planes respectively as it was explained; so it is easy to know the values of *A*, *B* and *C* that are needed to characterized the part. The value *A* is obtained searching the farthest perpendicular lines that are in a plane XY along the axis *Z* and for the value *B* and *C* it is the same process but along the axis *X* and *Y* respectively. Finally, to get the value of the first digit of the code it is used the relationships on the figure 57. In the next figure is presented an example of how the system obtain the value *A*; for the value *B* and *C* it is de same but working with the axis *X* and *Y*.

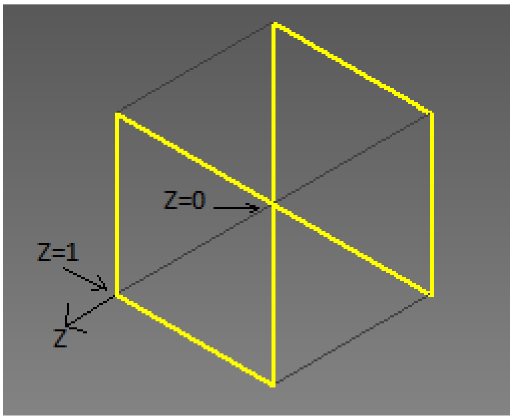


Figure 64: Example to obtain the value A

As it is shown in the figure 64, the main algorithm return two groups, one with the lines in *Z=0* and the other one with the lines in *Z=1*. The farthest lines is the part are the ones which are in *Z=1*, so the value of *A* is 1.

6.5.2 Digit 2, main shape

This digit defines the main shape of the part. The implemented algorithm that gets the value of the second digit works with the base of the parts; in flat part the base is in *Y=0* and in long and cubic parts the base is in *Z=0*. Depending on the part, the algorithm detects the number of lines of it base and calculates the amount of right-angled between lines; with this information the developed system is able to recognize and classify main shapes of the parts. For flat parts, the different main shapes that the algorithm is able to recognize are:

- Rectangular shape: the base of the part has four lines and four right angles.
- Right-angled triangle shape: the base of the part has three lines and one right angle or six lines and six right angles.
- Angularly shape: the base of the part has more than four lines and don't have any right angle.
- Angular and rectangular shape: the base of the part has right angles between lines and circular arcs that are not holes, with different start and terminate points.
- Rectangular shape with small deviations: the algorithm detects in the axis *Y* more than two groups if lines in planes *XZ*.

In the next figures, an example of each feature is presented.

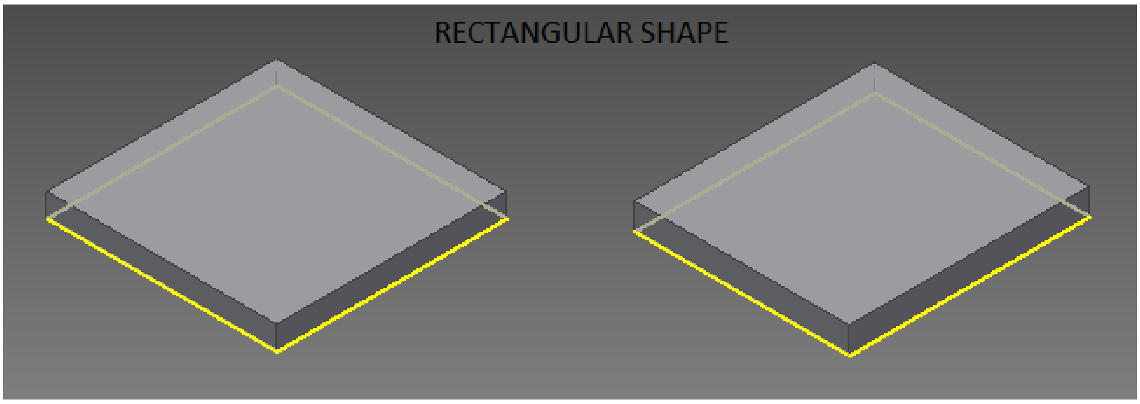


Figure 65: Example of rectangular shapes in flat parts

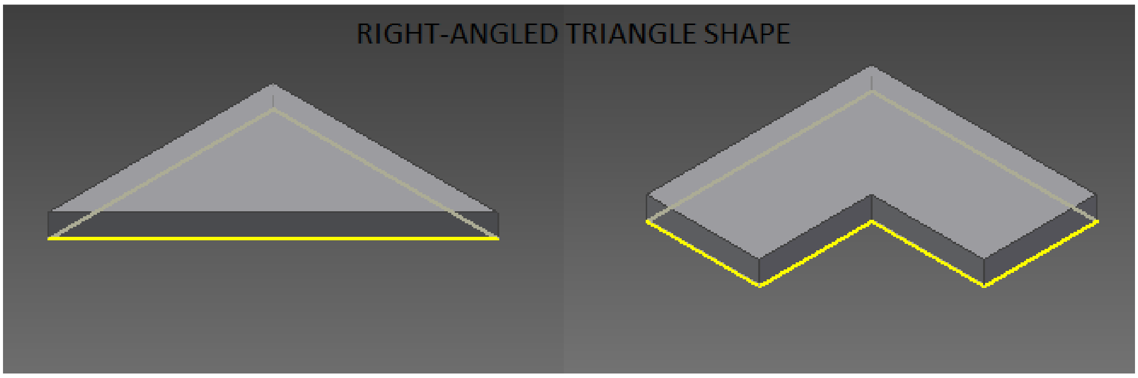


Figure 66: Example of right-angled triangle shapes in flat parts

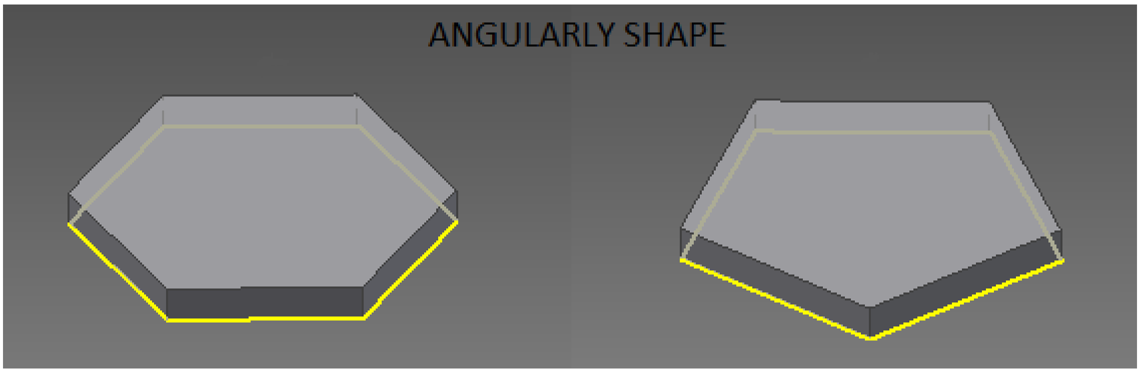


Figure 67: Example of angularly shapes in flat parts

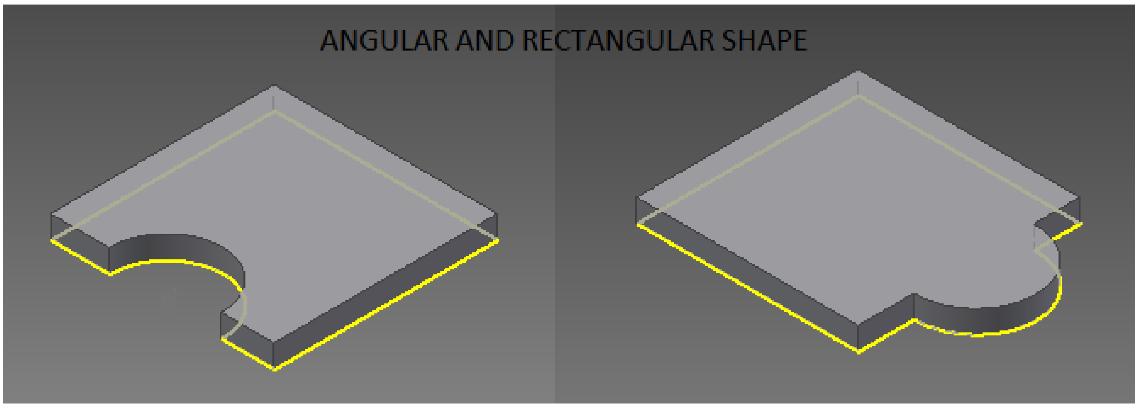


Figure 68: Example of angular and rectangular shapes in flat parts

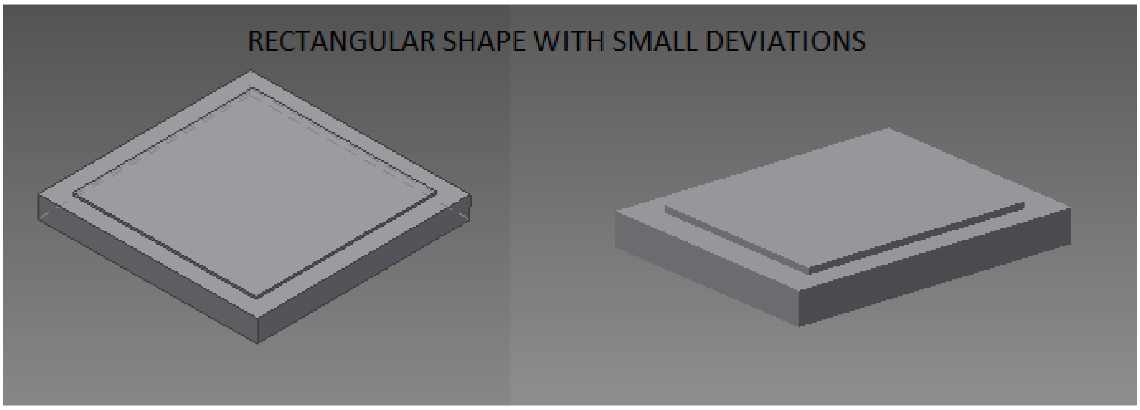


Figure 69: Example of rectangular shapes with small deviations in flat parts

For long parts, the main shapes that the developed system recognizes are presented as following:

- Rectangular cross-section: the cross-section of the part has four lines and four right angles.
- Orthogonal cross-section: the cross-section of the part has three lines and one right angle or six lines and six right angles.
- Any angular cross-section: the cross-section of the part has more than four lines and don't have any right angle.

The next figures present an example of each feature of the second digit for long parts.

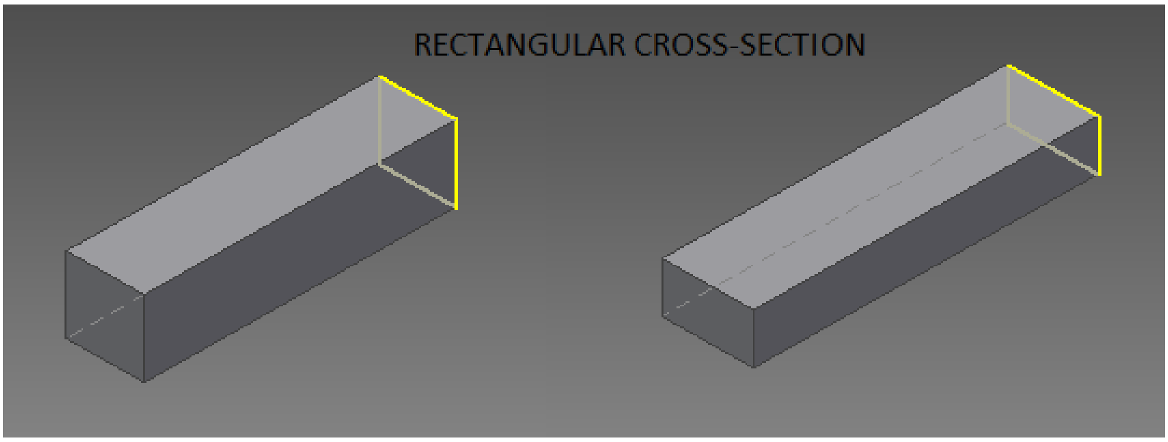


Figure 70: Example of rectangular cross-sections in long parts

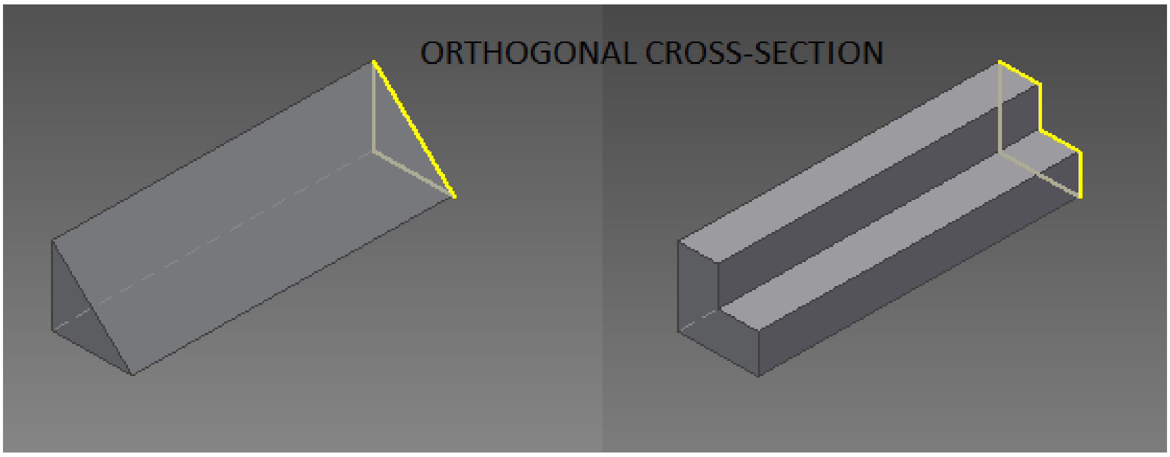


Figure 71: Example of orthogonal cross-sections in long parts

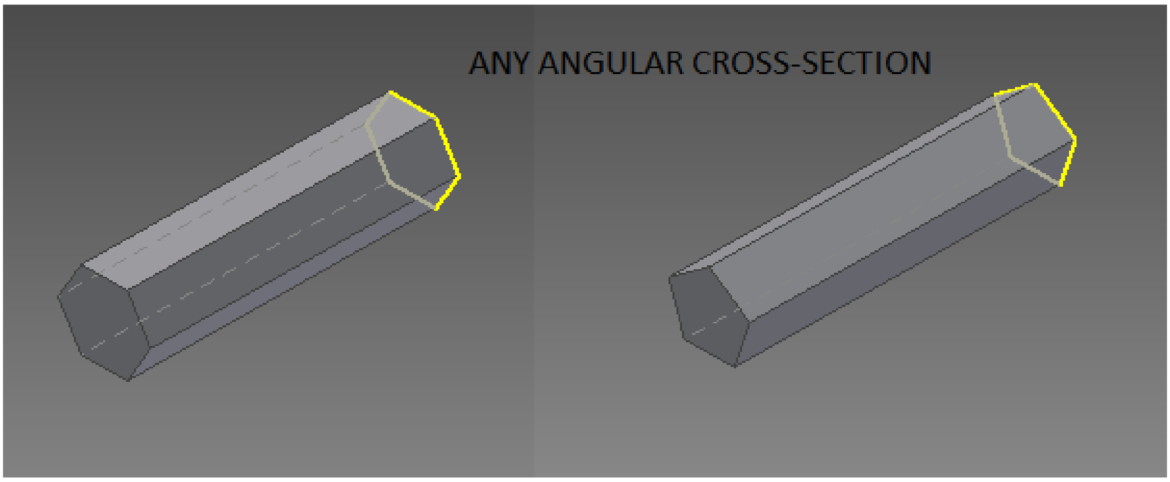


Figure 72: Example of any angular cross-sections in long parts

For cubic parts, the different main shapes that the developed system recognizes are:

- Cuboids shape: the base of the part has four lines and four right angles.
- Orthogonal shape: the base of the part has three lines and one right angle or six lines and six right angles.

- Composited parallelepiped: the algorithm detects in the axis *X*, *Y* and *Z* more than two groups of lines forming surfaces.

In the next figure several examples are presented to leave clear the classification of the second digit for cubic parts.

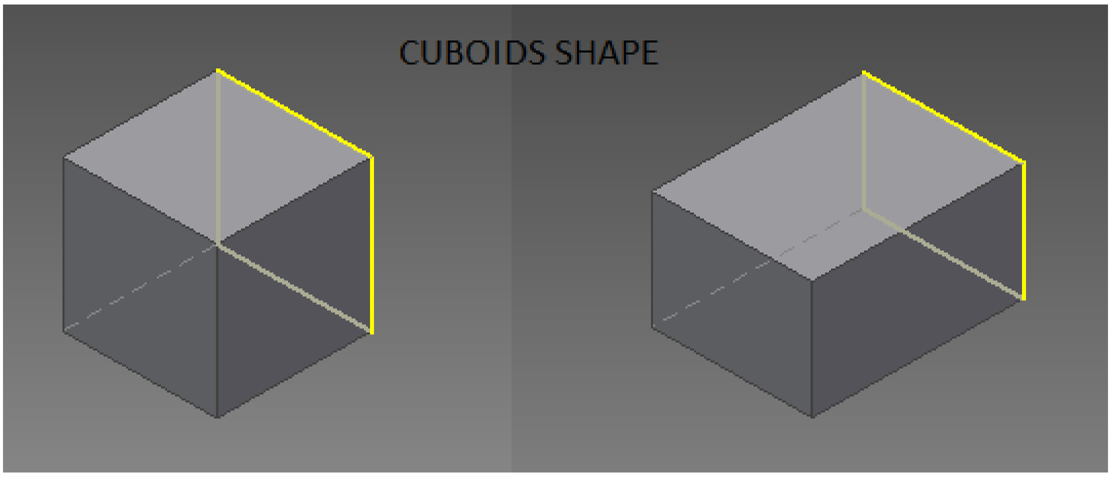


Figure 73: Example of cuboids shapes in cubic parts

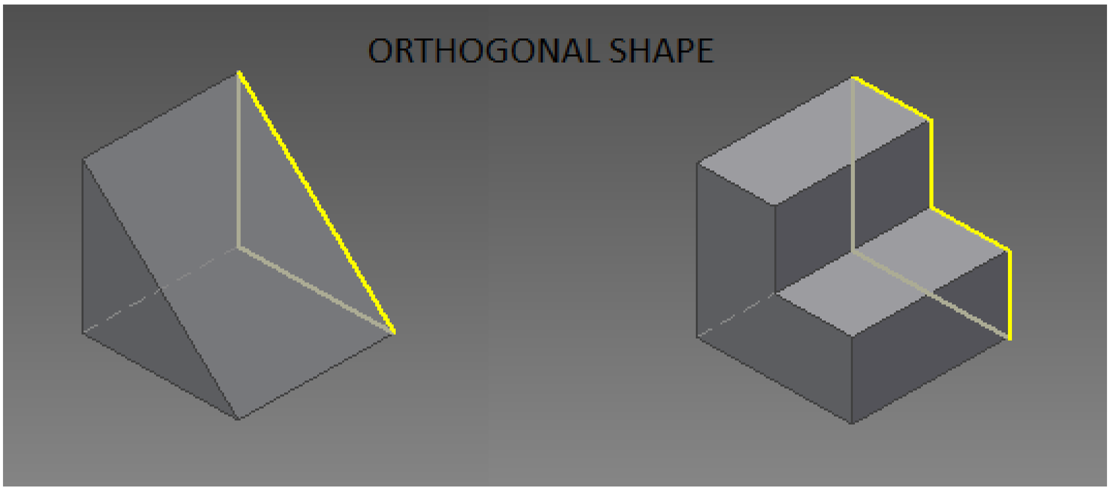


Figure 74: Example of orthogonal shapes in cubic parts

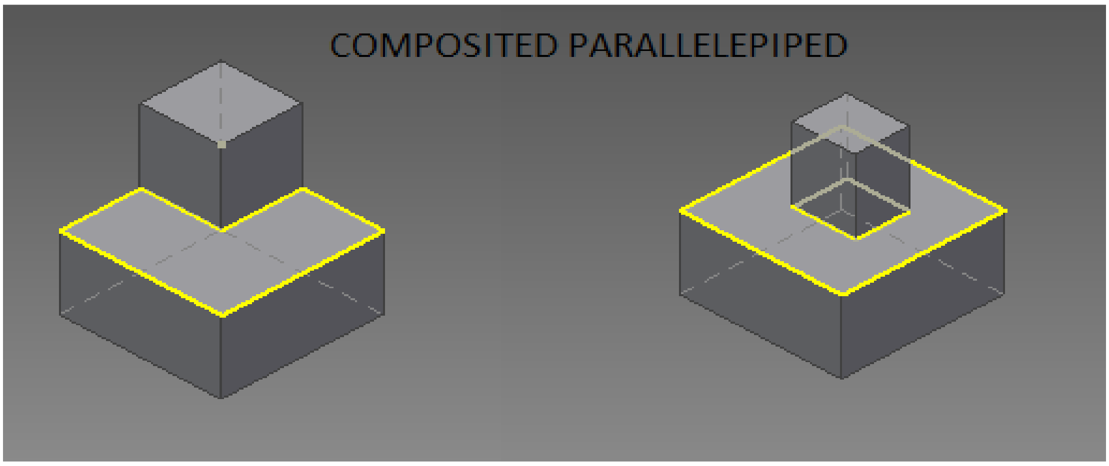


Figure 75: Example of composited parallelepipeds in cubic parts

6.5.3 Digit 3, main bores

According to the Opitz Code System, the digit three classifies main bores in the part. A bore is defined by two complete arcs with same start and end points and it is a main bore when the length of the bore is equal to the value *A*, *B* or *C* of the part if the bore is positioned in the direction of the axis *Z*, *X* or *Y* respectively. With the main algorithm all the arcs in each direction of the axis are extracted. Then the arcs that have the same start and end point are selected; with the properly length to find the correct amount of the main bores. The third digit of the Opitz Code is able to characterize the next features:

- No main bores.
- One main bore.
- Two main bores.
- More than two main bores.
- Bores in several directions (only for long and cubic parts).

In the next figures examples of the classification possibilities of the digit three are presented.

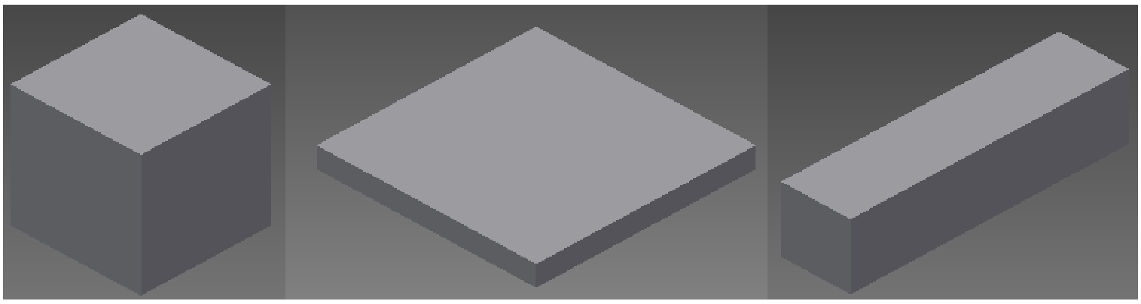


Figure 76: Cubic, flat and long parts without man bores

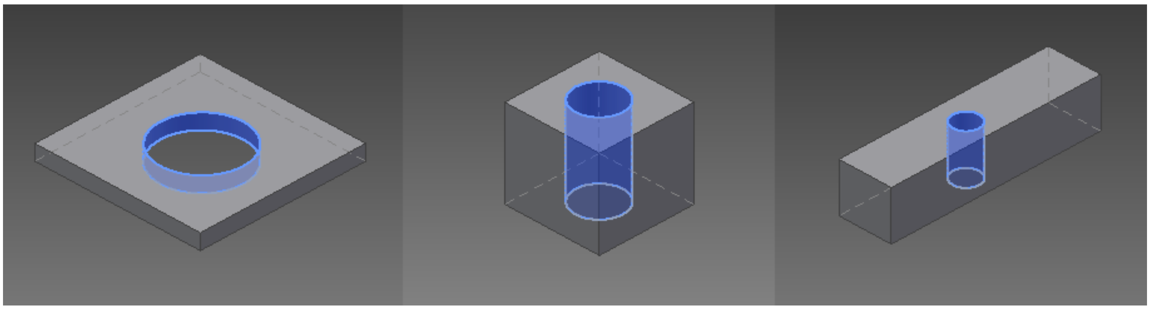


Figure 77: Flat, long and cubic parts with one main bore

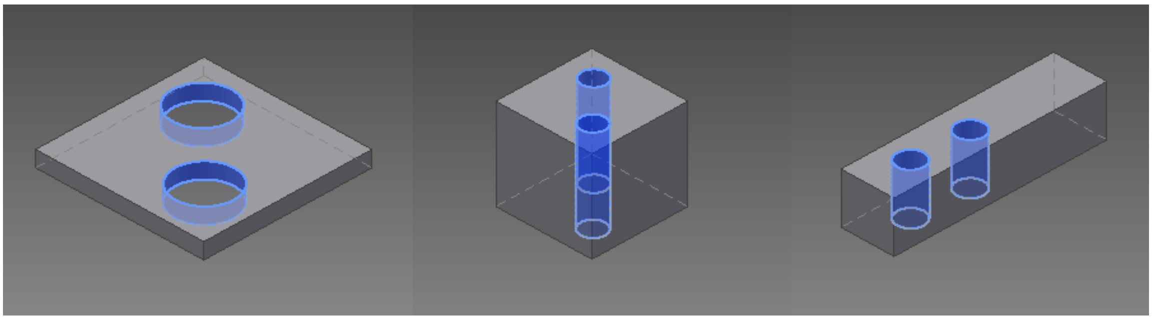


Figure 78: Flat, long and cubic parts with two main bores

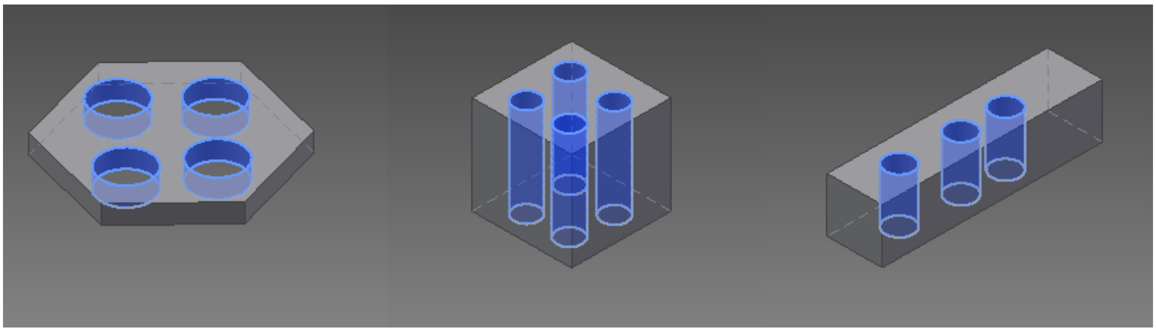


Figure 79: Flat, long and cubic parts with several main bores

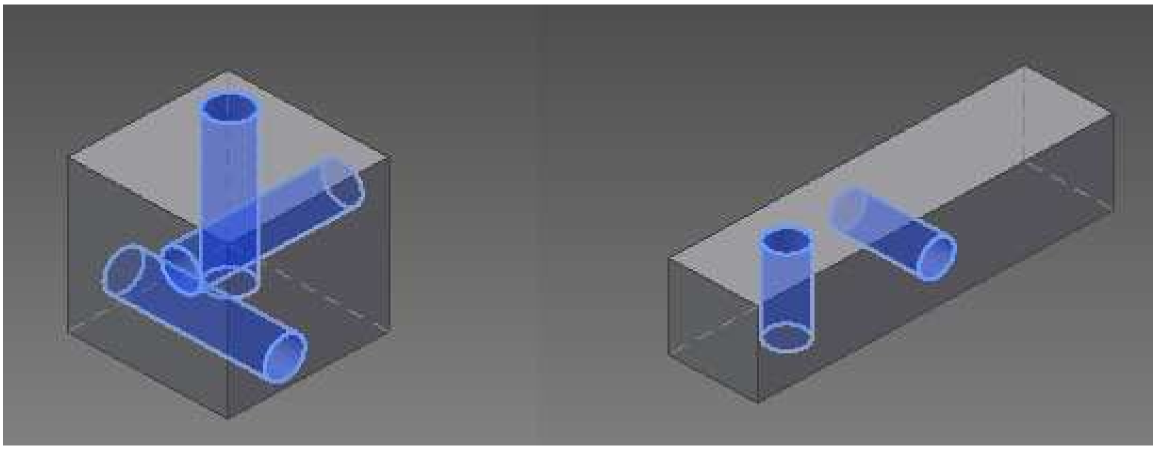


Figure 80: Cubic and long parts with bores in several directions

6.5.4 Digit 4, machining of plane surface

This digit classifies surface features and the developed program is able to recognize:

- No surface machining: the surface of the part doesn't have any variation and it is totally smooth.
- A flat surface: the surface of the part has a step (only for long and cubic parts).
- Curved surface: the surface of the part is curve.

In the next figures a clear example of each feature is shown.

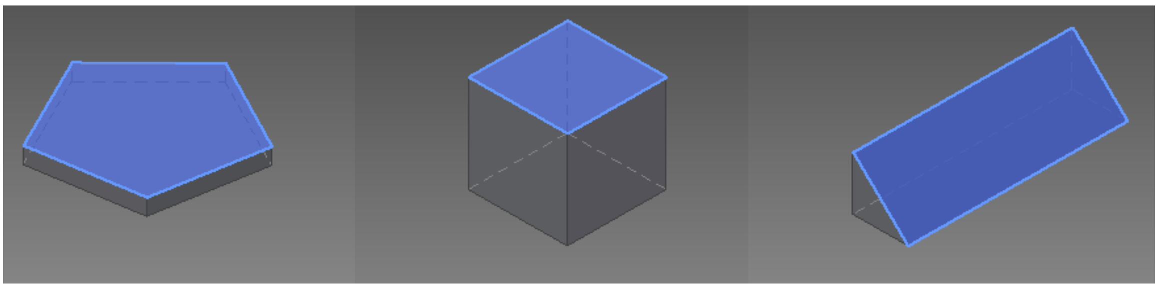


Figure 81: Flat, cubic and long parts with no machining surface

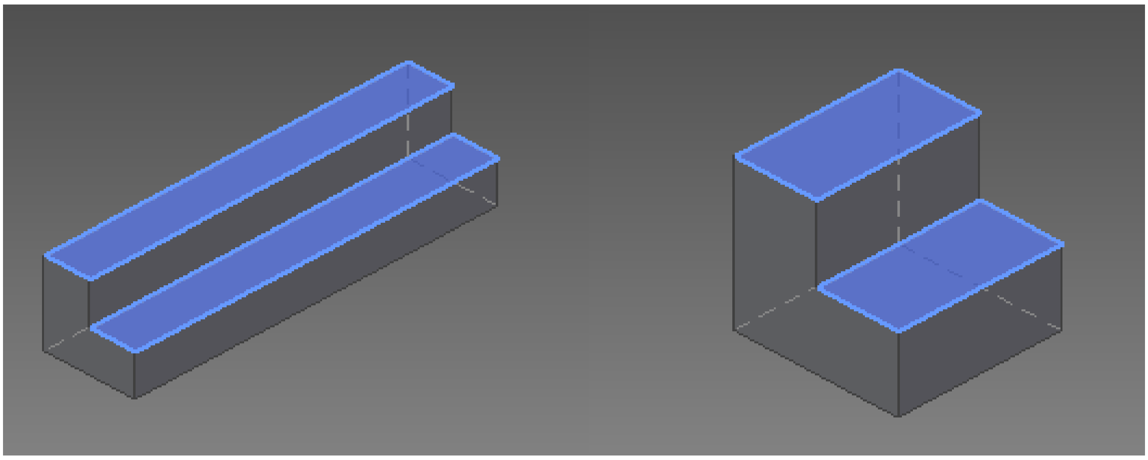


Figure 82: Long and cubic parts with a flat surface feature

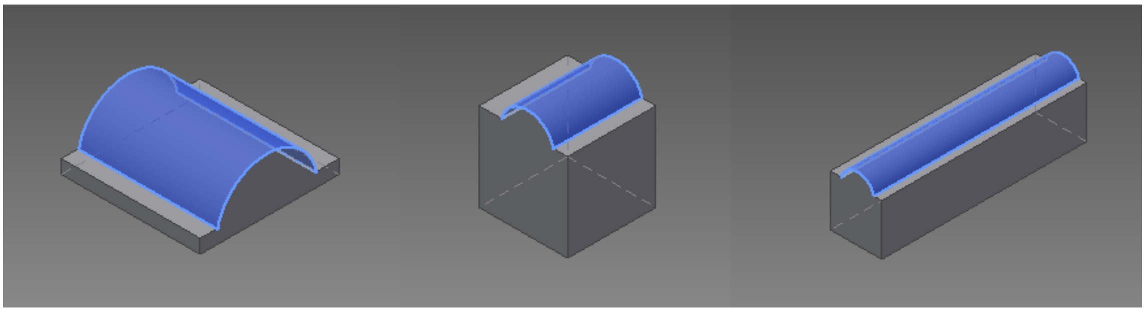


Figure 83: Flat, cubic and long parts with curved surface

6.5.5 Digit 5, auxiliary holes

The last digit of the Opitz Code Form classifies the auxiliary holes of the part. An auxiliary hole is two complete arcs with same start and end points but the distance between the arcs is smaller than the values A, B or C of the part if the auxiliary hole is in the direction of the axis Z, X or Y respectively. The main algorithm extracts the arcs along each axis and then another algorithm select the properly arcs to detect the amount of auxiliary holes in the part. The digit 5 is able to classify this several features:

- No auxiliary holes.
- Auxiliary holes in one direction.
- Auxiliary holes in several directions (only for long and cubic parts).

In the next figures several example of each possibility of auxiliary holes classification is presented.

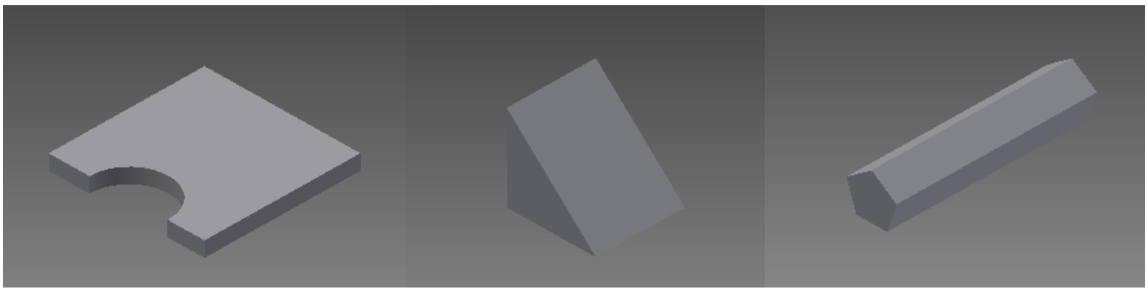


Figure 84: Flat, cubic and long parts without auxiliary holes

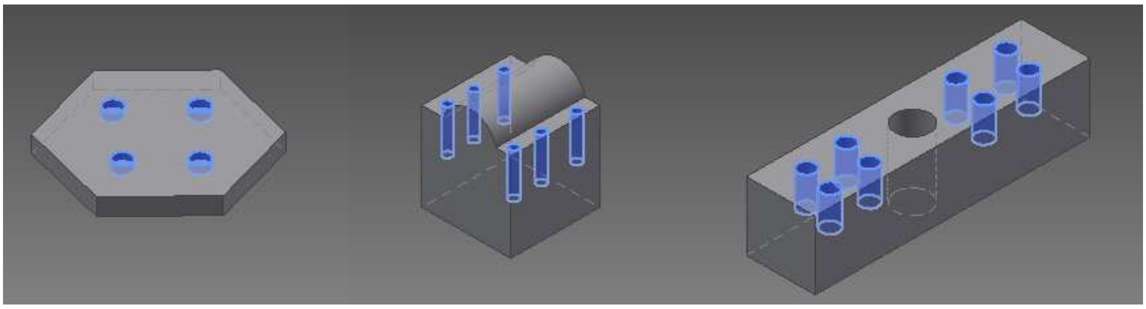


Figure 85: Flat, cubic and long parts with auxiliary holes in one direction

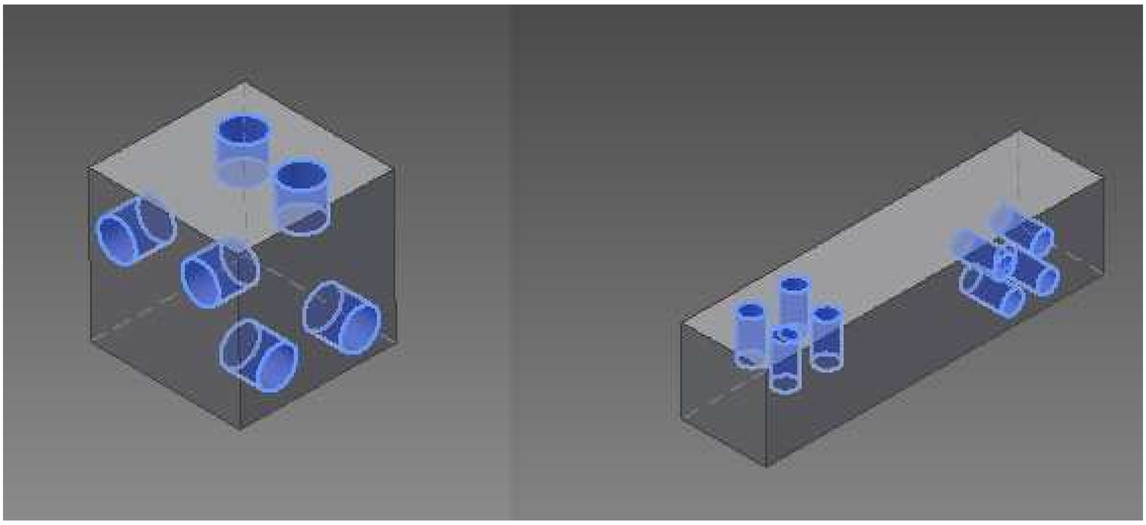


Figure 86: Cubic and long parts with auxiliary holes in several directions

6.6 Similarity measures

According to the objectives of this Thesis, a distance function to compare Opitz codes as a shape signature has been developed. The complete implementation of the distance function for this Thesis has been explained in chapter 5.

6.7 Graphic user interface (GUI)

To represent the result of this work and the developed system, a graphic interface is developed to manage properly all the program. This interface is implemented in MATLAB as it will be explained in the next chapter. The GUI consists in three part performed to achieve the best functionality of the developed system.

In figure 87 the first part of the GUI is presented. OPITZ CODE GENERATOR is the part that generates the Opitz code of an IGES file of a 3D model designed, there is a small space to introduce the name of the file and a bottom to generate the code, also if the IGES file does not exist an error pop up will be appeared behind the name of the file suggesting the user to try again another file; finally as it is presented, the save bottom will save in a folder a text file with the code inside and with the IGES file name as the name of the text file creating a database of parts.

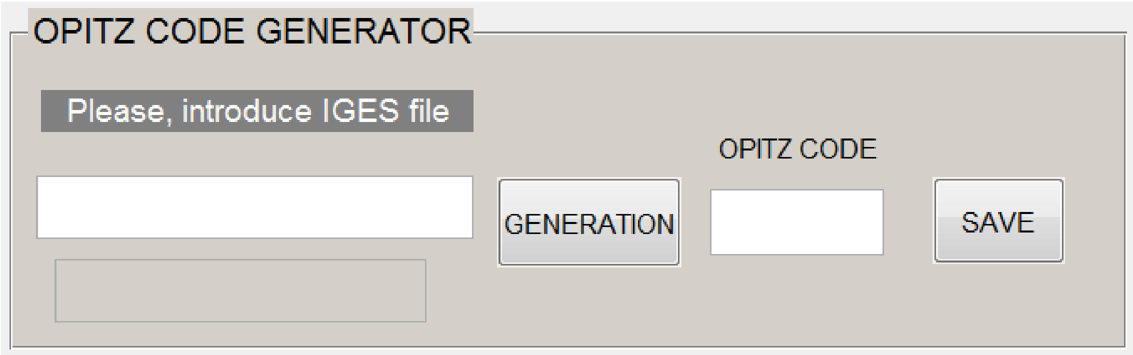


Figure 87: First part of the GUI, Opitz code generator

Moreover, the Figure 88 presents the second part of the GUI, The COMPARISON OF PARTS. Here there are two spaces to introduce the IGES file that the user want to compare, when the bottom is pressed, the result of the similarity measure will appear; also if the IGES files are not a correct ones, an error message will be appeared suggesting to introduce a new ones.

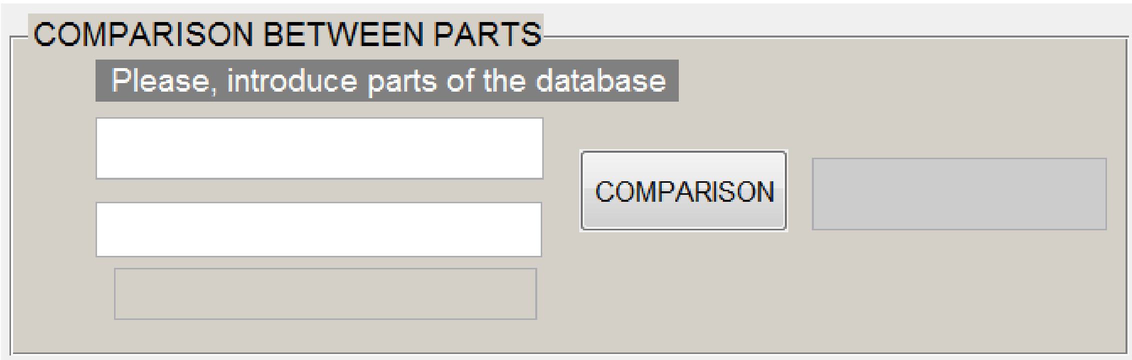


Figure 88: Second part of the GUI, comparison between parts

The last part of the GUI is presented in the Figure 89. The SIMILARITY MEASURE GUI has the functionality of searching in the database all the parts that have certain similarity with a new one designed. For the implementation of this function, the GUI has a first part to generate the Opitz code of the new designed part and then there are several bottoms with possible percentages of similarity. Pressing one on these bottoms, the GUI will return a list of parts that are this percentage similar regarding to the new designed one.

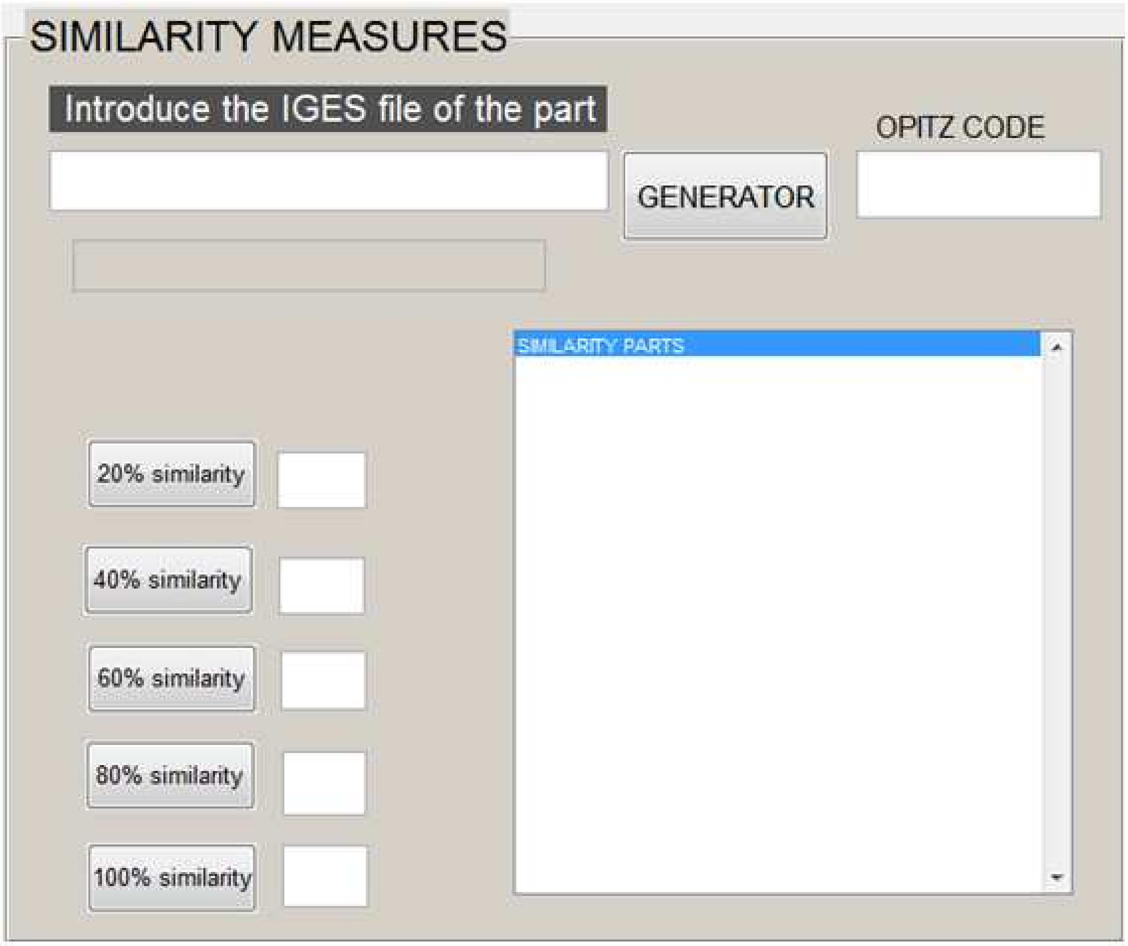


Figure 89: third part of the GUI, similarity measures

Chapter 7

VALIDATION, TESTING AND RESULTS

7.1 Implementation and development

Two software programs have been used in the developing Opitz code generation and similarity retrieval system. Autodesk Inventor version 2012 is a CAD software used to design parts and Matlab 8.0 (R2012a) is a programming environment with a programming language that has been used to implement and develop the classification and similarity algorithms and generate the graphic interface.

Autodesk Inventor is able to work with 2D or 3D parts. In this research, the parts designed are 3D model; these models are created using Constructive Solid Geometry (CSG). With Autodesk Inventor the database has been designed part by part, then this software allows generating an IGES file of each part as well. There are different types of IGES files, including output solid as surfaces, solids or wireframe. The developed system uses wireframe type as an output.

The IGES file of each part is parsed with a Matlab function to extract all the values of the entities as well as all the algorithms explained in the chapter 6 are implemented in Matlab to performance properly the system. Finally, a graphic interface developed in Matlab too is done to supply a good functionality of the system to the users.

7.2 Validation and testing

For the validation of the system a database of more than two hundreds parts has been created; but in addition the classification system is able to manage more three hundreds of different parts with different Opitz codes. A 3D model for each class, flat, long and cubic, has been designed following the ranges of the figure 62, and then it has been adding 3D models of each class trying to fulfill all the possible combinations of features. The database tries to give a global vision of all the different features in each class to validate the system.

In order to test the system, the shape signature of each part has been generated checking that the code is the correct one and has been saved in a folder to perform the database of codes. Then for the distance function, several proves has been done to see that identifies properly the dissimilarity between parts. The communication with the system is done through the GUI. Thus receiving correct results from the system is a proof for GUI implementation as well as for the algorithms.

7.3 Illustrative example

In the next section, an example is presented explaining the functionality of the developed system step by step. One common part is analyzed by the program generating its Opitz code and saving the value in the database. In the next figure the part and its Opitz code are presented.

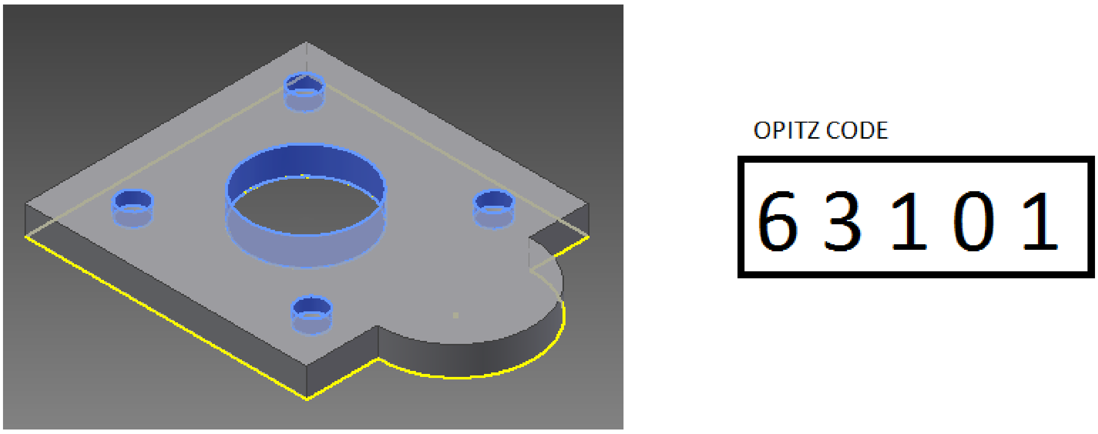


Figure 90: Flat part rectangular with one main hole and auxiliary holes

The Opitz code of the part shown in Figure 90 is 63101. Each digit is explained as following:

- Digit 1: the value 6 signifies that the part is flat.
- Digit 2: the value 3 signifies that the main shape of the part is circular and rectangular.
- Digit 3: the value 1 signifies that the part has a main bore.
- Digit 4: the value 0 signifies that the part does not have a machining surface.
- Digit 5: the value 1 signifies that the part has auxiliary holes in one direction.

The next step is to generate the Opitz code of the part. First the IGES file is generated with the CAD software and then the name of this file is transferred in the graphic interface and pressing the generator bottom the code is obtained as it is shown in the next figure.

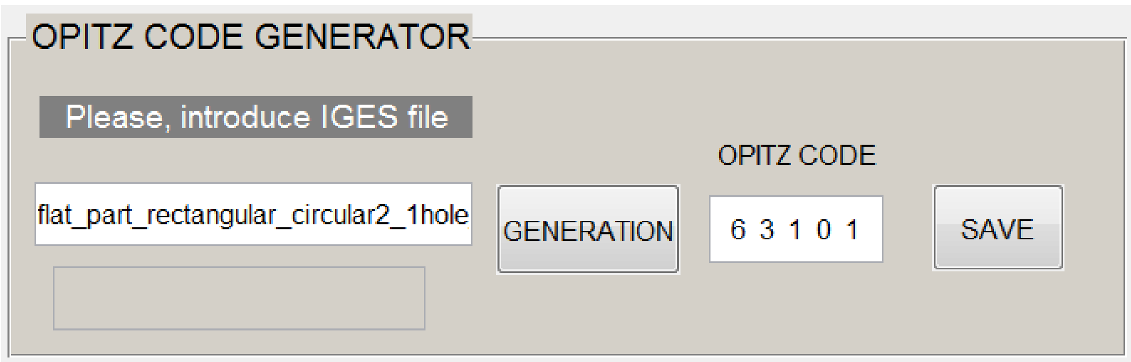


Figure 91: Generation of the Opitz code with the GUI

Finally, the save bottom store the value of the code in a text file with the name of the IGES file inside a folder. Pressing it, the part is added to the database. In the next figure a part of the database is presented.

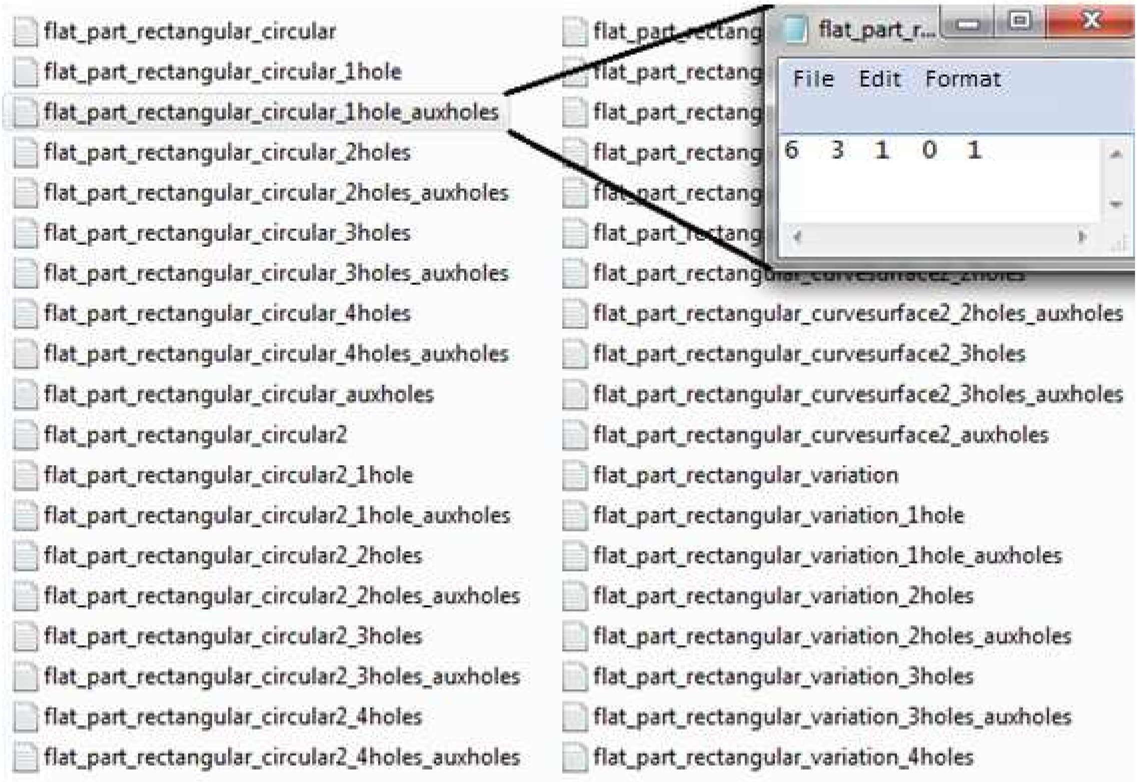


Figure 92: A part of the database

Regarding to the comparison of parts, the Figure 93 presents a part that is going to be compared with the previous one.

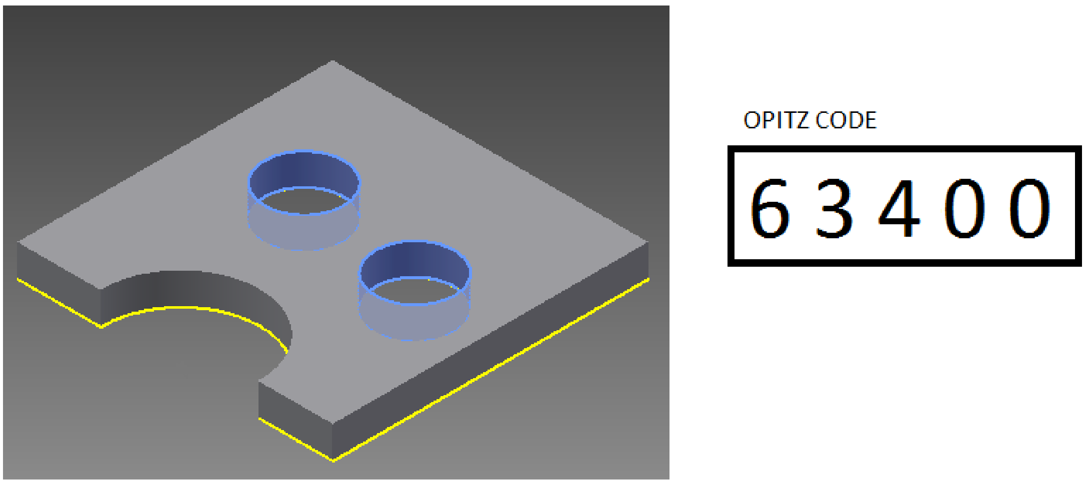


Figure 93: Flat part rectangular with two holes

In the Figure 94 the graphic user interface according to the comparison of parts is presented. The comparison between both parts presented above is done and as it is shown there is a similarity of sixty percent between parts, this means that both parts are flat parts, have the same main form and the same machining surfaces, but they are different in main bores.

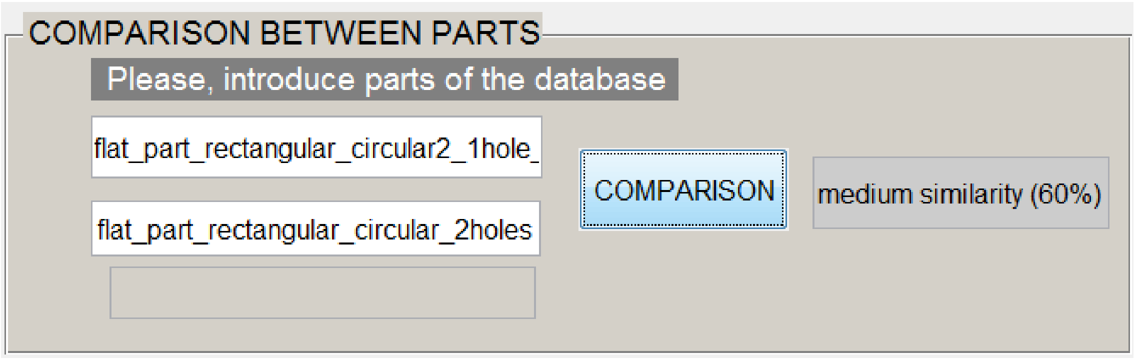


Figure 94: Comparison of parts with the GUI

To conclude this illustrative example, it is supposed that the part on the Figure 93 is going to be designed as a new one. The third part of the GUI brings the opportunity of searching parts in the database with certain percentage of similarity regarding to the new one and presenting them in a list. In the Figure 95 a designed part (Figure 93) is introduced and compared with all the parts of the database presenting in a list the ones that have a similarity of 60% with the introduced part.

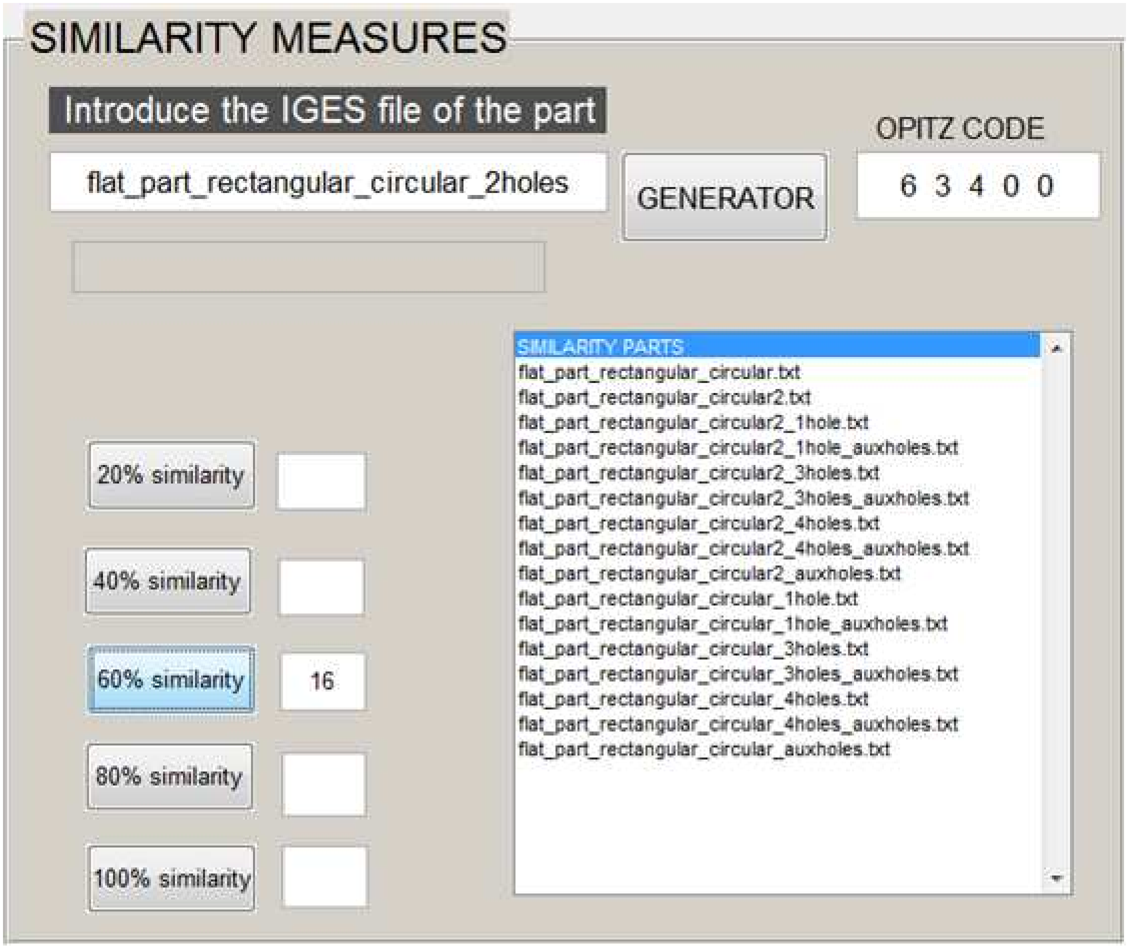
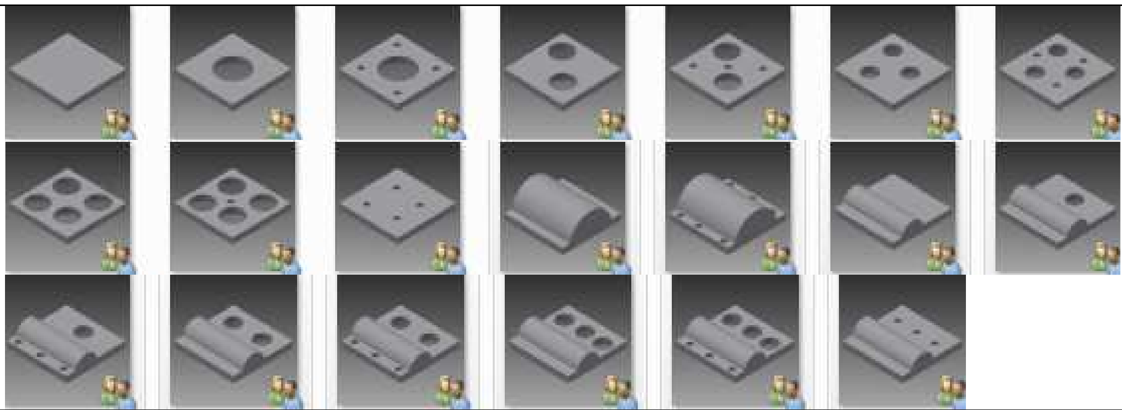
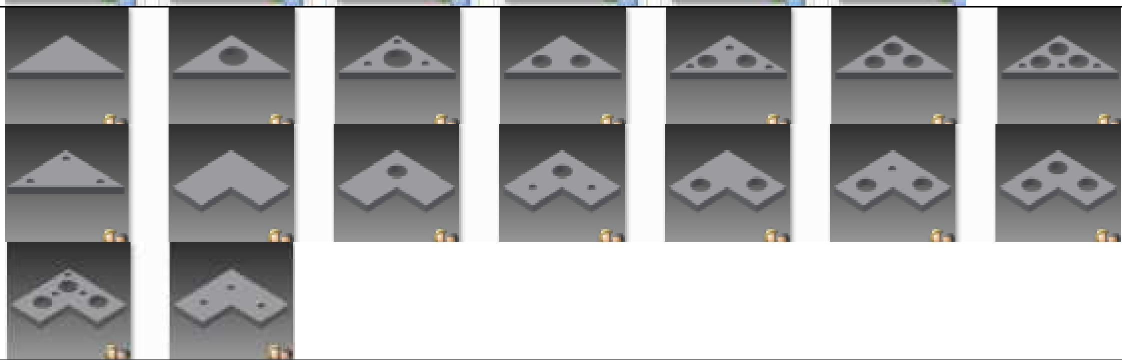
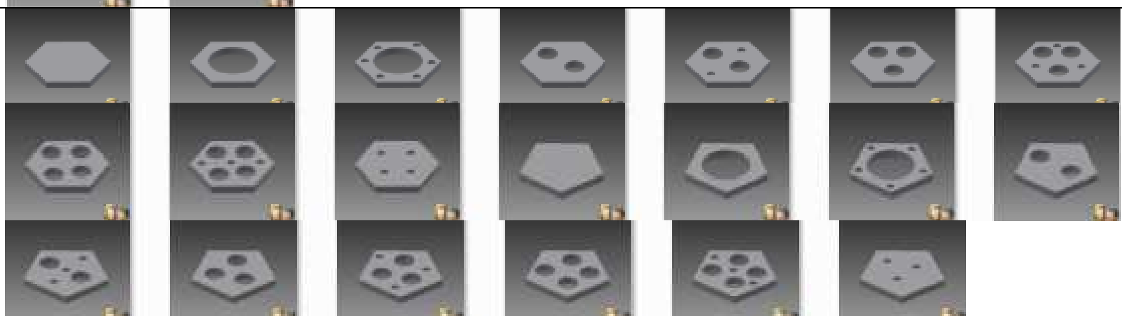
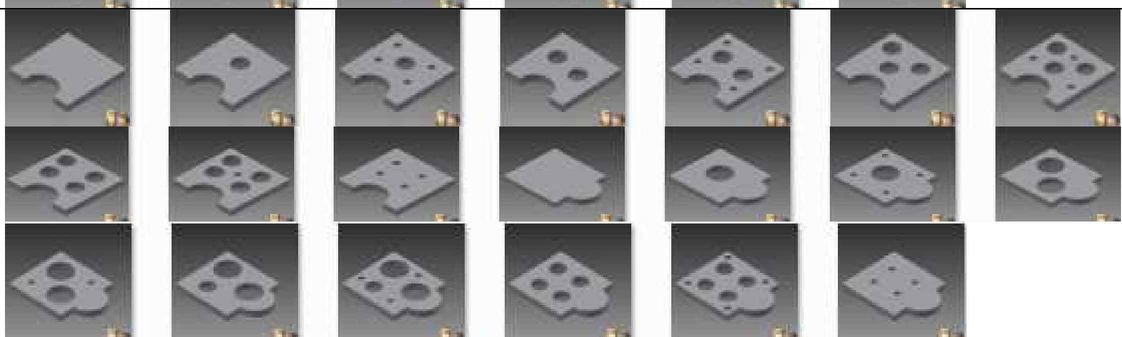
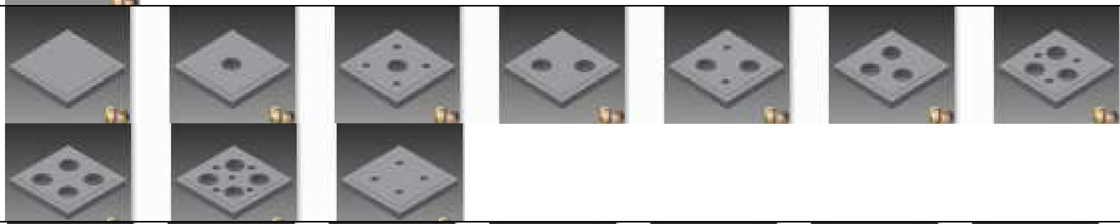
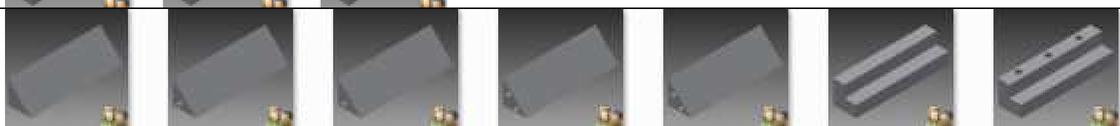
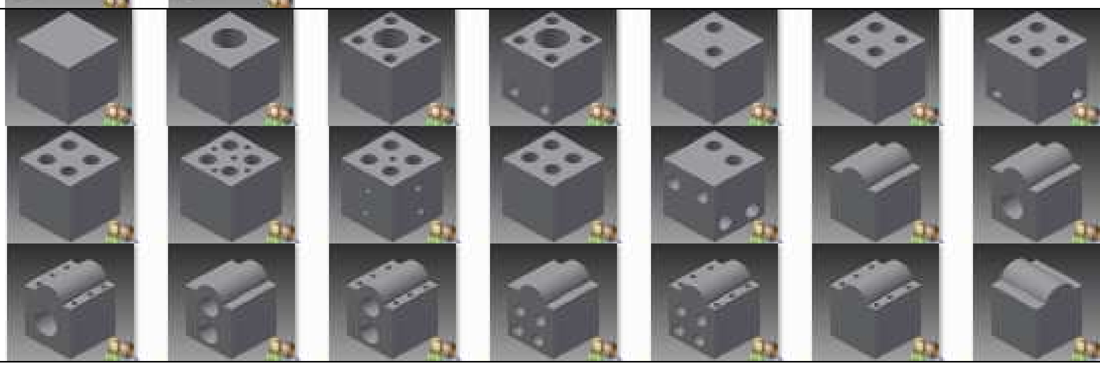


Figure 95: Example of the third part of the GUI

7.4 The generated database for evaluation

To evaluate and to test the developed system, a database of random parts has been created as Figure 96 presents. The database includes 201 parts in both very different and very similar forms to test and validate the algorithms and the system. In the database, the parts are grouped with a similarity of forty percent; this means that the Opitz code of both parts has the same value in the digit one and two.

	PARTS
1	
2	
3	
4	
5	

	
6	
7	
8	
9	
10	

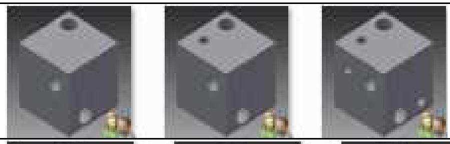
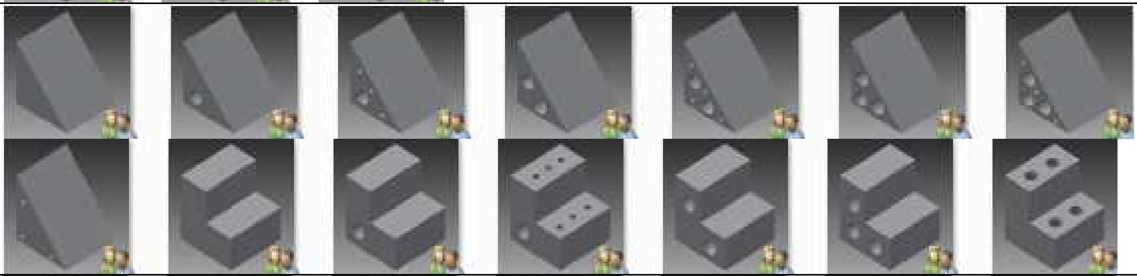
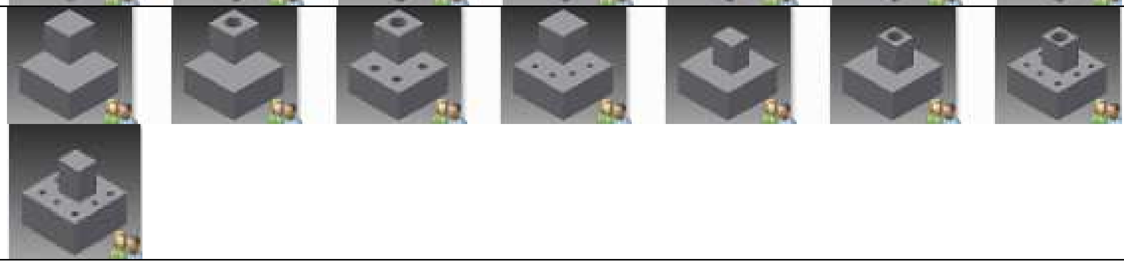
	
11	
12	

Figure 96: Classified database

Chapter 8

CONCLUSIONS AND FUTURE WORK

The ultimate objective of this research has been set as an investigation and realization of a novel technique to reuse knowledge information in the design of 3D solid models in order to optimize the design process of a new model. To achieve this goal, in this Thesis a classification and comparison system for non-rotational parts has been developed. The developed system uses two softwares, Autodesk Inventor to design 3D models and generate IGES file, and Matlab. The system developed in Matlab works with the IGES file to recognize and extract the features of the part. For the classification, Opitz Code System is implemented to classify the extracted features of the IGES files generating a shape signature. All the shape signatures, related CAD files and Opitz codes are stored in a database hosting more than two hundreds 3D models. An algorithm applied in the database allows the comparison between parts and the reusing of knowledge information with a developed distance function. Three Graphical User Interfaces have been developed to map and connect the database, the CAD software and the user requirements and orders in addition to the system management as is presented in Figure 49. As the final step, the correct functionality of the system has been validated and tested using database members.

The evaluation and testing of the system proves a properly functionality and a good implementation of the developed system. Classification and comparison of huge number of features in non-rotational parts has resulted to an acceptable outcome in retrieval of knowledge information from the database. However, the program classifies the designed features of the parts with Opitz Code System that is a classification system focused on classifying the features for manufacturing purposes. The repercussion of using Opitz code is presented in figure 97. Both parts have the same classification code but they do not have identical 3D shapes. This problem happens due to the fact, that Opitz Code is focused in the features for the manufacture of the part and not for the design. In the other hand, as it is shown in Figure 97, the both parts have one main bore but with different size or both parts have auxiliary holes but in different positions.

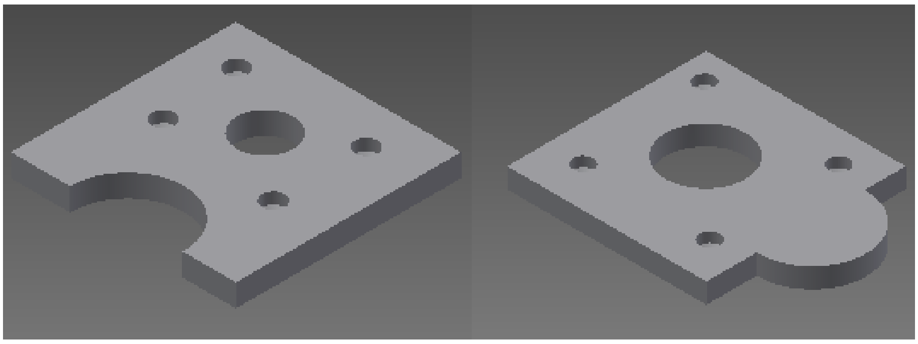


Figure 97: Equal shapes in manufacturing features

Another possible upgrade for this research refers to the amount of features in non-rotational parts which is really significant. A suggested improvement of the developed system always can be the recognition and extraction of more features focusing in more details of the designed parts. For the developed distance function, the possibility of comparing more specific features of the part or focusing in other similarities will improve the quality of the developed distance function as well.

APPENDIX A

In this appendix, a literature review is presented to aid to understand properly the functionality of each entity of the IGES file that is going to be used in the thesis [IGES06].

A.1 Line entity (type 110)

A line is a bounded, connected portion of straight line which has different start and terminate points.

A line is defined by its end points. Each end point is specified relative to definition space by triple coordinates. With respect to definition space, a direction is associated with the line by considering the start point to be listed first and the terminate point second.

A.1.1 Directory Entry section

In this section, all the lines of the 3D model are defined.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Entity Type Number	Parameter Data	Structure	Line Font Pattern	Level	View	Xformation Matrix	Label Display	Status Number	Sequence Number
110	⇒	< *n.a.* >	#, ⇒	#, ⇒	0, ⇒	0, ⇒	0, ⇒	??????**	D #
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Entity Type Number	Line Weight	Color Number	Parameter Line Count	Form Number	Reserved	Reserved	Entity Label	Entity Subscript	Sequence Number
110	#	#, ⇒	#	0				#	D # + 1

Figure 98: Directory Entry section of the line entity [IGES06]

A.1.2 Parameter Data section

The Parameter Data section is where the values and the information of the entity is registered.

Index	Name	Type	Description
1	X1	Real	Start Point *P1*
2	Y1	Real	
3	Z1	Real	
4	X2	Real	Terminate Point *P2*
5	Y2	Real	
6	Z2	Real	

Figure 99: Parameter Data section of the line entity [IGES06]

A.2 Circular arc entity (type 100)

A circular arc is a connected portion of a circle that has different start and terminate points. The definition space coordinate system is always chosen so that the circular arc lies in a plane either coincident with, or parallel to, the XT, YT plane.

A circular arc determines unique arc endpoints and an arc center point (the center of the parent circle). By considering the arc end points to be enumerated and listed in an ordered manner, star point first, followed by the terminate point, a direction with respect to definition space can be associated with the arc. The ordering of the end points corresponds to the ordering necessary for the arc to be traced out in a counterclockwise direction. This convention serves to distinguish the desired circular arc from its complementary arc.

A.2.1 Directory Entry section

(1) Entity Type Number	(2) Parameter Data	(3) Structure	(4) Line Font Pattern	(5) Level	(6) View	(7) Xformation Matrix	(8) Label Display	(9) Status Number	(10) Sequence Number
100	⇒	< n.a. >	#,⇒	#,⇒	0,⇒	0,⇒	0,⇒	??????**	D #
(11) Entity Type Number	**(12) Line Weight**	**(13) Color Number**	**(14) Parameter Line Count**	**(15) Form Number**	**(16) Reserved**	**(17) Reserved**	**(18) Entity Label**	**(19) Entity Subscript**	**(20) Sequence Number**
100	#	#,⇒	#	0				#	D # + 1

Figure 100: Directory Entry section of the circular arc entity [IGES06]

A.2.2 Parameter Data section

Index	Name	Type	Description
1	ZT	Real	Parallel Z_T displacement of arc from X_T, Y_T plane
2	X1	Real	Arc center abscissa
3	Y1	Real	Arc center ordinate
4	X2	Real	Start point abscissa
5	Y2	Real	Start point ordinate
6	X3	Real	Terminate point abscissa
7	Y3	Real	Terminate point ordinate

Figure 101: Parameter Data section of the circular arc entity [IGES06]

A.3 Transformation matrix entity (type 124)

The transformation Matrix Entity transforms three-row column vectors by means of a matrix multiplication and then a vector addition. The notation for this transformation is:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \cdot \begin{bmatrix} X\ INPUT \\ Y\ INPUT \\ Z\ INPUT \end{bmatrix} + \begin{bmatrix} T1 \\ T2 \\ T3 \end{bmatrix} = \begin{bmatrix} X\ OUTPUT \\ Y\ OUTPUT \\ Z\ PUTPUT \end{bmatrix}$$

Here, column [X INPUT, Y INPUT, Z INPUT] is the vector being transformed, and column [X OUTPUT, Y PUTPUT, Z OUTPUT] is the column vector resulting from this transformation. R= [Rij] is a 3 row by 3 column matrix of real numbers, and T = column [T1, T2, T3] is a three-row column vector of real numbers. Thus, 12 real numbers are required for a Transformation Matrix Entity. This entity can be considered to be an "operator" entity in that it starts with the input vector, operates on it as described above, and produces the output vector.

Frequently, the input vector lists the coordinate of some point in one coordinate system, and the output vector lists the coordinates of that same point in a second coordinate system. The matrix R and the translation vector T express a general relationship between the two coordinate systems. By considering special input vector such as column [1,0,0], column [0,1,0] and column [0,0,1] and computing the corresponding output results, a geometric appreciation of the spatial relationship between the two coordinate systems can be obtained.

A.3.1 Directory Entry section

(1) Entity Type Number 124	(2) Parameter Data ⇒	(3) Structure < n.a. >	(4) Line Font Pattern < n.a. >	(5) Level < n.a. >	(6) View < n.a. >	(7) Xformation Matrix 0, ⇒	(8) Label Display < n.a. >	(9) Status Number ****??**	(10) Sequence Number D #
(11) Entity Type Number 124	(12) Line Weight < n.a. >	(13) Color Number < n.a. >	(14) Parameter Line Count #	(15) Form Number 0-1,10-12	(16) Reserved	(17) Reserved	(18) Entity Label	(19) Entity Subscript #	(20) Sequence Number D # + 1

Figure 102: Directory Entry section of the transformation matrix entity [IGES06]

A.3.2 Parameter Data section

Index	Name	Type	Description
1	R11	Real	Top Row
2	R12	Real	.
3	R13	Real	.
4	T1	Real	.
5	R21	Real	Second Row
6	R22	Real	.
7	R23	Real	.
8	T2	Real	.
9	R31	Real	Third Row
10	R32	Real	.
11	R33	Real	.
12	T3	Real	.

Figure 103: Parameter Data section of the transformation matrix entity [IGES06]

REFERENCES

[AK06] Emad S. Abouel Nasr, Ali K. Kamrani; A new methodology for extracting manufacturing features fron CAD system; 2006.

[AKKS99] Ankerst M., Kastenmüller G., Kriegel H-P., Seidl T.; 3D shape histograms for similarity search and classification in spatial databases. Lecture Notes in Computer Science, vol.1651; 1999.

[CBC97] Cybenko G., Bhasin A., Cohen K.; Pattern recognition of 3D CAD objects; Smart Eng Syst Des, 1997.

[CGK03] Antonio Cardone, Satyandra K. Gupta, Mukul Karnik; A Survey of Shape Similarity Asessment Algorithms for Product Design and Manufacturing Applications; Journal of Computing and Information Science in Engineering; June 2003.

[CK01] Cyr CM., Kimia BB.; 3D object recognition using shape similarity-based aspect graph; In: Proc. Inter. Conf. Comp. Vis.; 2001.

[CM12] CELLULAR MANUFACTURING UNIT 7; http://www.ignou.ac.in/upload/UNIT7-55.pdf; date of research 10/11/12.

[EI96] A. Efrat and A. Itai; Improvements on bottleneck matching and related problems using geometry; 1996.

[EM03] El-Mehalawi M., Miller R.; A database system of mechanical components based on geometric and topological similarity; Computer-Aided Des, 2003.

[EM94] T. Eiter and H. Mannila; Computing Discrete Fréchet Distance; Christian Doppler Labor für Expertensysteme, Technishe Universität Wien, Paniglgasse 16, 1040 Wien; 1994.

[ENR97] A. Elinson, D.S. Nau and W.C. Regli; Feature-based Similarity Assessment of Solid Models; 4th Symposium on Solid Modelling and Applications; 1997.

[ES07] C. Elanchezhan, G. Shanmuga Sundar; Computer Aided Manufacturing; Firewall Media; 2007.

[FCCL11] D. Francia, A. Ceruti, G. Caligiana, A. Liverani; Shape Similarity: Methods and Industrial Applications; Proceedings of the IMProVe 2011, International conference on Innovative Methods in Product Design, 2011.

[GT12] Chapter6: Group Technology http://me.emu.edu.tr/majid/IENG447/IE%20447/PDF%20FILES/CIM%20Lecture%20Notes%206.pdf; date of research: 20.10.12.

[Hawo68] E. A. Haworth; Group Technology – Using the Opitz System; The production Engineer; January 1968.

[HHLY01] Jung Hyun Han, Member, IEEE, Inho Han, Eunseok Lee, and Juneho Yi, Member, IEE; Manufacturing feature recognition toward integration with process planning; IEEE transactions on systems, man and cybernetics part B, Vol.31, No.3; 2001.

[HKR93] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge; Comparing Images Using the Hausdorff Distance; IEEE TRANSACTIONS ON PATERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.15, NO.9; 1993.

[HPR00] Jung Hyun Han, Member, IEEE, Mike Pratt and William C. Regli, Member, IEE; Manufacturing Feature Recognition from Solid Models: A Status Report; IEEE transactions on robotic and automation, vol. 16, no. 6; 2000.

[HSKK01] Hilaga M., Shinagawa Y., kohmura T., Kunii TL.; Topology matching for fully automatic similarity estimation of 3D shapes; In: SIGGRAPH.: ACM Press, 2001.

[IGES06] Initial Graphics Exchange Specification 5.3; The U.S. Product Data Association; 2006.

[IJLKR04] Natraj Iyer, Subramaniam Jayanti, Kuijang Lou, Yagnanarayanan Kalyanaraman, Karthik Ramani; Three-dimensional shape searching: state-of-the-art review and future trends; Computer-Aided Design 37 (2005) pages 509-530, 2004.

[JC88] S. Joshi and T. Chang; Graph-based heuristics for recognition and machined features from a 3D solid model; computer-aided design, vol.20, No.1, pp. 58-66; 1988.

[JS12] Jacopo Aleotti, Stefano Caselli; A 3D shape segmentation approach for robot grasping by parts; Robotics and Autonomous Systems, vol.60, Issue 3, March 2012.

[KLM97] Kalyanapasupathy V., Lin E., Minis I.; Group Technology code generation over the internet ETC97/CIE-4290; 1997.

[LMS08] K. Lalit Narayan, K. Mallikarjuna Rao, M.M.M. Sarcar; Computer Aided Design and Manufacturing; 2008.

[MKJ08] Y. Mingqiang, K. Kidiyo, R. Joseph; A survey of shape feature extraction techniques; Pattern Recognition, Peng-Yeng Yin, Ed.2008, 43-90; 2008.

[Mitc10] P. Mitchell; Polygon Matching, Measuring similarity between polygons; 2010.

[MPRS01] McWherter D., Peabody M., Regli WC., Shoukofandeh A.; Solid model databases: techniques and empirical results; ASME J Comput Inf Sci Eng, 2001.

[MS12] MANUFACTURING SYSTEM. Wwwme.nchu.edu.tw; date of research: 23.10.12.

[MW12] http://www.m-w.com; date of research: 23.10.12.

[OFCD01] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin; Matching 3D Models with Shape Distributions; International Conference on Shape Modeling and Applications, Genova; 2001.

[Opit68] H. Opitz; Verschlüsselungsrichtlinien und Definitionen zum werkstückbeschreibenden Klassifizierungssystem; 1968.

[Opit70] Herwart Opitz; A Classification System to Describe Workpieces; 1970.

[Orou85] Joseph O´Rourke; Curve similarity via signatures; In G:T: Toussaint, editor, Computational Geometry, pager 295-317; 1985.

[Requ80] Aristides A. G. Requicha; Representations for Rigid Solids: Theory, Methods and Systems; Computing Surveys, vol.12, no.4; December 1980.

[PW12] http://www.proofwiki.org/wiki/Definition:Discrete_Metric; date of research: 15.11.12.

[RYD01] Ramesh M., Yip-Hoi D., Dutta D.; Feature based shape similarity measurement for mechanical parts; ASME J Comput Inf Sci, 2001.

[SHI96] Shum H., Hebert M., Ikeuchi K.; On 3D shape similarity; Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1996.

[SJ99] S. Santini, Member, IEEE, and R. Jain, Fellow, IEEE; Similarity Measures; IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.21, no.9; 1999.

[SSGD03] Sundar H., Silver D., Gagvani N., Dickinson S.; Skeleton based shape matching and retrieval; In: Proceedings of Shape Modeling and Applications, 2003.

[Sund94] Sunderesh S. Heragu; Group Technology and Cellular Manufacturing; IEEE Transactions on Systems Man and Cybernetics, vol.24, no.2; February 1994.

[Tolo10] M. Tolouei-Rad; Integration of part classification, cell formation and capacity adjustment; Journal of Achievements in Materials and Manufacturing Engineering, vol.39, issue 2; 2010.

[TR84] Timothy J. Greene, Randall P. Sadowski; A review of Cellular Manufacturing Assumptions, Advantages and Design Techniques; Journal of Operations, Management vol.4, no.2; February 1984.

[Vand04] John P. Van de Geer; Some Aspects of Minkowski Distance; Department of Data Theory; Leiden University; 2004.

[Vert06] Remco C. Vertkamp; Shape Matching: Similarity Measures and Algorithms; Department of Computer Science, Utrecht University Padualaan 14, 3584 CH, Utrecht, The Netherlands.

[VH05] Remo C. Veltkamp and Michiel Hagedoorn; Shape Similarity Measures, Properties, and Constructions; Department of Computer Science, Utrecht University Padualaan 14, 3584 CH, Utrecht, The Netherlands.

[WK98] E. Wang and Y.S. Kim; Form feature recognition using convex decomposition: results presented at the 1997 ASME CIE feature panel session; Computer Aided Design 30, 13; 1998

[WR97] Wolfson H., Rigoutsos I.; Geometric Hashing: an overview; IEEE Comput Sci Eng; 1997.

[Zhan00] M. Zhang; Geometric Modeling; spring 2000.

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

Unterschrift:

Stuttgart, den 16.01.2013