Journal of Achievements in Materials and Manufacturing Engineering
VOLUME 26
ISSUE 1
January
2008

Basic solutions on shape complexity evaluation of STL data

B. Valentan*, T. Brajlih, I. Drstvensek, J. Balic
Faculty of Mechanical Engineering, University of Maribor,
Smetanova 17, 2000 Maribor, Slovenia
* Corresponding author: E-mail address: bogdan.valentan@uni-mb.si

Received 19.02.2007; published in revised form 01.01.2008

Analysis and modelling

ABSTRACT

Purpose: Purpose of this paper is to present basic solutions on shape complexity, based on basic information of the STL data.
Design/methodology/approach: Paper presents a few methods of mathematically evaluating the complexity of the shape. Methods vary from very simple based on the number of triangles in STL file, STL file size and the parts volume, to the more complex mathematical evaluation based on the basic relations of the STL data.
Findings: We discovered that evaluation of shape complexity based only on basic data of STL data gives us some basic results on part complexity and can be used for further researches.
Research limitations/implications: For parts with large block volume/part volume ratio and thinner parts with free form surfaces only the first method is suitable and gives suitable results.
Practical implications: In a rapidly developing field of manufacturing technologies choosing the optimal manufacturing procedure is a difficult and crucial decision. Usually the decision is based on experience evaluation that is fast and can be optimal. Usually, this method produces goods results, but in some cases this method can lead to cost increases and reduced economic efficiency without us even knowing that. Therefore, it is crucial, that a fast and simple solution is developed, by which the optimal way of manufacturing can be determined.
Originality/value: Choosing maximum efficient manufacturing processes on base of part complexity is a new perspective in manufacturing, which, properly evolved and complied can cause revolution in manufacturing optimization, especially in hybrid manufacturing processes.
Keywords: Engineering design; Shape complexity; STL file; STL file parameters

1. Introduction

With rapid development of additive fabrication technologies, the importance of shape complexity as an influence factor is decreasing. However, there is still major need for evaluation of models shape complexity in the field of classical – conventional machining, where geometrical properties can have a great impact on production costs. Those costs can be drastically decreased by determining the optimal way of manufacturing [1-3]. Additionally, in the field of rapid tooling, the parts complexity represents the parting plane layout and eventual tool construction (inserts, cores, etc.) [4-8]. Even when using rapid prototyping procedure [9,10], the support material consumption depends greatly on the complexity of the part and can, together with the problem of optimal orientation and position of the part, significantly influence the manufacturing costs.

Basic solutions of shape complexity evaluation can be made in several ways [11,12]. Usually, most efficient way is evaluation based on previous experiences regarding a certain manufacturing procedure.

However, such an estimate is very subjective and largely depends on the person that made it (in a case of highly experienced person, estimation is extremely fast and accurate, but in a case of inexperienced persons, estimation can be completely false and non optimized and expensive solution can be accepted). This paper presents basic possibilities of evaluating shape complexity based on STL CAD file format data. The methods presented are simple mathematical equations based on fundamental information that can be acquired from STL data.

© Copyright by International OCSCO World Press. All rights reserved. 2008

2. STL file format

Research is based on the three-dimensional CAD model in the STL file format [13,14], that was originally developed for Stereo Lithography [1,15] rapid prototyping procedure. STL file format is generally accepted in the field of rapid prototyping and is supported by all major CAD software packages. STL is a facet based representation that approximates surface and solid entities only. Entities such as points, lines, curves, and attributes like layers, colour, are ignored during the output of the STL from the CAD systems. An STL file consists of a list of facet data. Each facet is uniquely identified by a unit normal (a line perpendicular to the triangle and with a length of 1.0) and by three vertices (corners). The normal and each vertex are specified by three coordinates each, so there is a total of 12 numbers stored for each facet.

2.1. Facet orientation

The facets define the surface of a 3-dimensional object. As such, each facet is part of the boundary between the interior and the exterior of the object. The orientation of the facets (which way is "out" and which way is "in") is specified redundantly in two ways which should be consistent. First, the direction of the normal is outward. Second, which is most commonly used now-a-day, list the facet vertexes in counter-clockwise order when looking at the object from the outside (*right-hand rule*) (Figure 1).

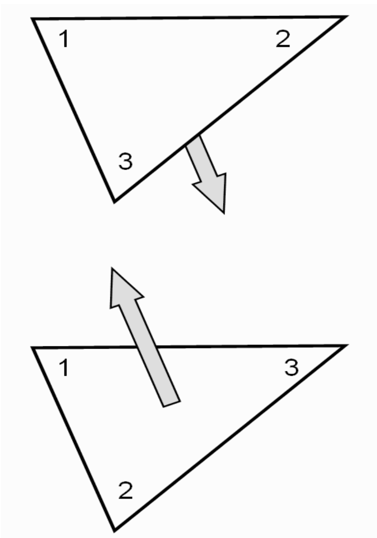

Fig. 1. Counter clockwise order of STL facets. The arrows point toward outside of the object

2.2. Vertex-to-vertex rule

Each triangle must share two vertices with each of its adjacent triangles (Figure 2). In other words, a vertex of one triangle cannot lie on the side of another (Figure 3).

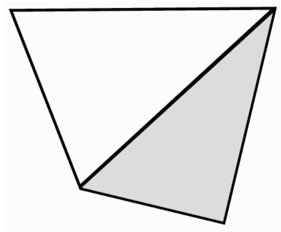

Fig. 2. Adjacent triangles

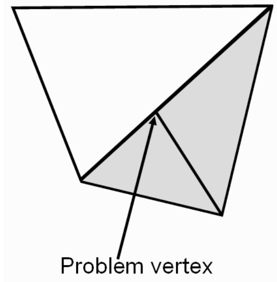

Fig. 3. A violation of the vertex-to-vertex rule

Because of the vertex-to-vertex rule, we know that a legal solid will have (3/2) edges for each face. This gives us three consistency rules against which to check:

1. number of faces (F) must be even
2. number of edges (E) must be a multiple of three
3. 2*E must equal 3*F

2.3. Axis and units

The format specified that the object represented must be located in the all-positive octant. In other words, all vertex coordinates must be positive-definite (nonnegative and nonzero) numbers. However, with a few exceptions most software used today allow the facets in arbitrary location. The STL file does not contain any scale information; the coordinates are in arbitrary units. In many RP pre-processing software, the program will try to determine the unit of the part by the magnitude of the dimension. For example if the X/Y/Z size of the part is below 10, it is very likely that it is an inch part.

2.4. Storage formats

There are two storage formats available for STL files, which are ASCII and BINARY. ASCII file is human-readable and can be modified by a text editor if required. The ASCII format is used for debugging, or when one has to transfer the file over a 7-bit channel.

STL ASCII file format

Here is a typical ASCII STL file that defines tetrahedron (figure 4):

```
solid tetrahedron
  facet normal -5.773503e-001 5.773503e-001 -5.773503e-001
    outer loop
      vertex 1.000000e+001 1.000000e+001 1.000000e+001
      vertex 1.000000e+001 -3.469447e-015 0.000000e+000
      vertex 4.336809e-015 -1.734723e-015 1.000000e+001
    endloop
  endfacet
  facet normal 0.000000e+000 0.000000e+000 1.000000e+000
    outer loop
      vertex 1.000000e+001 1.000000e+001 1.000000e+001
      vertex 6.938894e-015 0.000000e+000 1.000000e+001
      vertex 1.000000e+001 0.000000e+000 1.000000e+001
    endloop
  endfacet
  facet normal 1.000000e+000 0.000000e+000 0.000000e+000
    outer loop
      vertex 1.000000e+001 0.000000e+000 0.000000e+000
      vertex 1.000000e+001 1.000000e+001 1.000000e+001
      vertex 1.000000e+001 0.000000e+000 1.000000e+001
    endloop
  endfacet
  facet normal 0.000000e+000 -1.000000e+000 0.000000e+000
    outer loop
      vertex 1.000000e+001 0.000000e+000 0.000000e+000
      vertex 1.000000e+001 0.000000e+000 1.000000e+001
      vertex 6.938894e-015 0.000000e+000 1.000000e+001
    endloop
  endfacet
endsolid
```

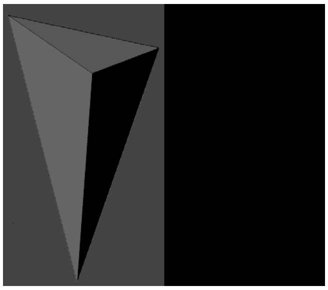

Fig. 4. tetrahedron is the simplest solid part with only 4 triangles

STL binary file format

The binary format uses the IEEE integer and floating point numerical representation. Binary (.STL) files are organized as an 84 byte header followed by 50-byte records each of which describes one triangle facet:

# of bytes	*Description*		
80	Any text such as the model name		
4	int equal to the number of facets in file		
	facet 1		
4	float	normal	x
4	float	normal	y
4	float	normal	z
4	float	vertex1	x
4	float	vertex1	y
4	float	vertex1	z
4	float	vertex2	x
4	float	vertex2	y
4	float	vertex2	z
4	float	vertex3	x
4	float	vertex3	y
4	float	vertex3	z
2	unused (padding to make 50-bytes)		
	facet 2		
4	float	normal	x
4	float	normal	y
4	float	normal	z
4	float	vertex1	x
4	float	vertex1	y
4	float	vertex1	z
4	float	vertex2	x
4	float	vertex2	y
4	float	vertex2	z
4	float	vertex3	x
4	float	vertex3	y
4	float	vertex3	z
2	unused (padding to make 50-bytes)		
	facet 3		
	...		

A facet entry begins with the x, y, and z components of the triangle's face normal vector. The normal vector points in a direction away from the surface and it should be normalized to unit length. The x, y, z coordinates of the triangle's three vertices come next. They are stored in CCW order when viewing the facet from outside the surface. The direction of the normal vector follows the "right-hand-rule" when traversing the triangle vertices from 1 to 3, i.e., with the fingers of your right hand curled in the direction of vertex 1 to 2 to 3, your thumb points in the direction of the surface normal.

Notice that each facet entry is 50 bytes. So adding the 84 bytes in the header space, a binary file should have a size in bytes = 84 + (number of facets) * 50. Notice the 2 extra bytes thrown in at the end of each entry to make it a nice even 50. 50 is a nice number for people, but not for most 32-bit computers because they store values on 4-byte boundaries. Therefore, when writing programs to read and write .STL files the programmer has to take care to design data structures that accommodate this problem.

The recent introduction of colour RP introduced the need for a 'colour STL' extension. One of these proposed extension make use of the 2 padding bytes to store the RGB data.

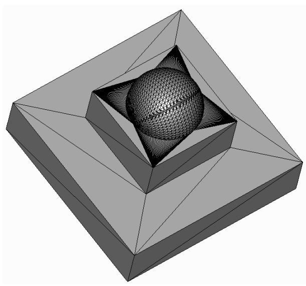

Fig. 5. Different sizes of triangles regarding to the complexity of different areas of the part

When exporting the three-dimensional CAD model to STL format the resolution of the file is selected. This is done by selecting desired deviation and angle tolerances. It is important to know, that by choosing the resolution the size of triangles is not fixed. Resolution only determines the minimal size of triangles used in more complex (curved) areas of the part. The simpler areas will be described with larger triangles (Figure 5). The whole part will be described with minimal possible number of triangles regarding to the chosen STL resolution.

The complexity of the part in general depends largely on the chosen manufacturing technology and the parts size. For example, a part with very small details can be very difficult to manufacture by a certain procedure, but using some other procedure can be made easily. On the other hand a part can be to large for certain machine, but can be easily manufactured on a machine with larger workspace. Furthermore, a certain freeform part is very complex for manufacturing by conventional machining, but its shape is not a problem for rapid prototyping. Taking this into consideration, it can be seen, that the general complexity evaluation should be made with consideration regarding the manufacturing procedure used to produce a certain part.

Evaluation of shape complexity is based on determining the correlations between adjacent triangles, triangle surfaces, normals and the volume of the part. Those simple evaluations cannot be taken universally, because they do not take the parts size (either very small ore very large) as a contributing factor to its complexity.

3. Test parts

For evaluation of the shape complexity seven various models (Figure 6-11) were designed. They vary from very simple shapes with only a couple of building triangles to highly complex parts that are described with over 500 000 triangles in the STL file. Table 1 shows information about test parts that are acquired from STL file. Block volume represent the minimal block volume that the parts fits into.

4. Evaluation of shape complexity based on number of triangles

STL data type describes the parts shell with triangles and accordingly the STL file structure is build. This means that the file size is in direct correlation with number of triangles. Therefore, basic shape evaluation [16] can be made by counting the number of triangles or examining the file size. When creating STL file, the number of triangles can not be directly set (that number is automatically set to the smallest possible value regarding to the overall resolution) therefore, this is a good method, but it should still be considered, that the optional overall resolution can be applied (resolution is therefore also directly connected with file size). The number of triangles can be directly determined with appropriate software (BIN to ASCII STL converter [17] or by examining the part properties in STL related CAD software (like Magics,…).

Figure 12 shows how STL file size and the number of triangles increases with the parts complexity. All parts were exported to STL format with the same resolution.

Table 1.
Information about test parts

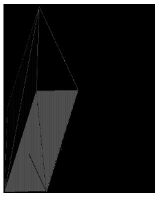
Fig. 6. prism

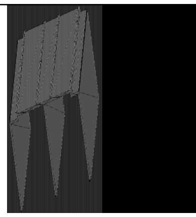
Fig. 7. rib

	Fig. 6. prism	Fig. 7. rib
Size BIN (KB)	0,484	16,6
Size ASCII (KB)	2,32	63,2
Number of triangles	8	340
Volume (mm^3)	63	79042
Surface (mm^2)	110	94353
Outer dimensions (mm)	5x5x5	83x60x180
Block Volume (mm^3)	125	8964000


Fig. 8. plug

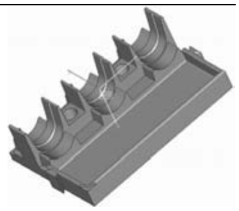
Fig. 9. housing

	Fig. 8. plug	Fig. 9. housing
Size BIN (KB)	165	503
Size ASCII (KB)	628	1847
Number of triangles	3372	10302
Volume (mm^3)	27056	1833
Surface (mm^2)	12850	2486
Outer dimensions (mm)	35,5x62,3x35,3	33x10x21
Block Volume (mm3)	78071	6930


Fig. 10. holder

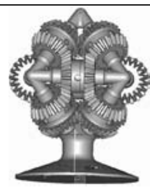
Fig. 11. wheels

	Fig. 10. holder	Fig. 11. wheels
Size BIN (KB)	1642	28563
Size ASCII (KB)	6502	108216
Number of triangles	33622	584962
Volume (mm^3)	89138	168157
Surface (mm^2)	51103	96585
Outer dimensions (mm)	213x180x57	93x111x93
Block Volume (mm^3)	2216160	960039

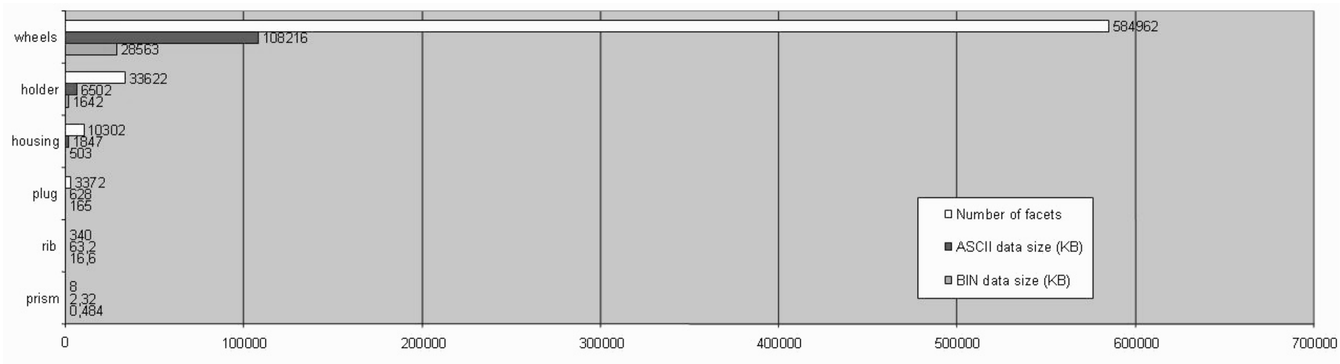

Fig. 12. Comparison between the BIN file size, ASCII file size and number of triangles

5. Evaluation of shape complexity based on ratio between volume and number of triangles

In order to acquire the ratio between volume and number of triangles, the volume data in needed. There are several ways to determine the parts volume. The determination of volume can be made according to the known algorithm [18], or by examining the parts volume information in our CAD software. For very fast but not so efficient way of volume determination the minimal block volume that can accommodate the part can also be used. Error is smaller; when the part is optimally orientated. The minimal block volume can be determined by searching for the maximal and minimal values of the triangle coordinates in each axis, so the vertexes of the block volume can be determined (Table 2).

Table 2.
Vertexes of the block volume

1	X_{min}	Y_{min}	Z_{min}
2	X_{min}	Y_{min}	Z_{max}
3	X_{min}	Y_{max}	Z_{min}
4	X_{min}	Y_{max}	Z_{max}
5	X_{max}	Y_{min}	Z_{min}
6	X_{max}	Y_{min}	Z_{max}
7	X_{max}	Y_{max}	Z_{min}
8	X_{max}	Y_{max}	Z_{max}

This method is much simpler and faster then calculating the real volume, but the estimate is very rough and can lead to significant errors.

Evaluation of shape complexity is:

$$\frac{volume}{number\_of\_facets} \quad (1)$$

Table 3 shows the calculated ratios for test parts. The calculations were made with exact and roughly determined volume. Figure 13 represents the ratio between exact and roughly determined volumes.

Table 3.
Comparison between evaluations of the shape complexity based on the volume and the number of triangles ratio

Part	Evaluation of complexity (exact volume)	Evaluation of complexity (block volume)
Prism	7,88	15,63
Rib	232,48	2636,47
Plug	8,02	23,15
Housing	0,18	0,67
Holder	2,65	65,91
Wheels	0,29	1,64

Observing Figure 13 and 14, it can be established, that the increase in parts complexity causes the volume/number of triangles ratio to decrease. The ratio difference between the exact and rough volume depends on the shape of the part and increases with thinner parts and free form surfaces.

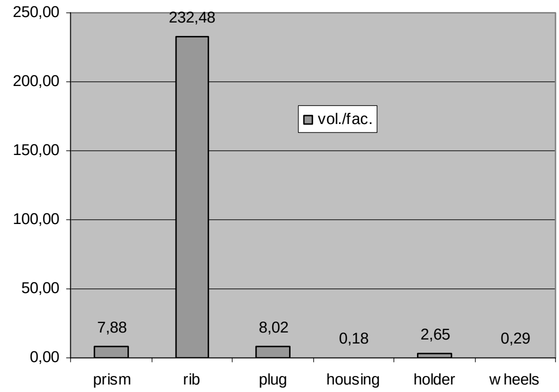

Fig. 13. Comparison between evaluations of shape complexity based on exact volume/number of triangles ratio

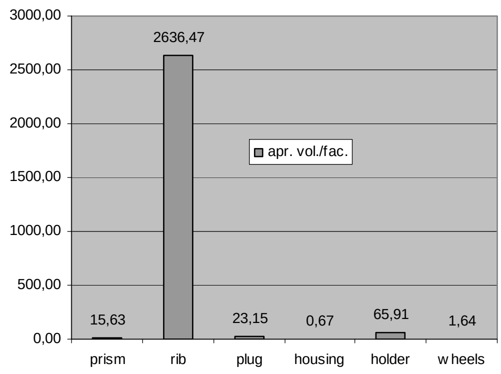

Fig. 14. Comparison between evaluations of shape complexity based on rough volume/number of triangles ratio

6. Evaluation of shape complexity based on ratio between parts volume and surface

Another possibility to get basic evaluation of shape complexity can be made on the volume/surface ratio. Calculated ratios for the test parts are presented in Table 4. Calculations were made both for exact volume and block volume (Figure 15 and 16). Evaluation of shape complexity is:

$$\frac{volume}{area} \qquad (2)$$

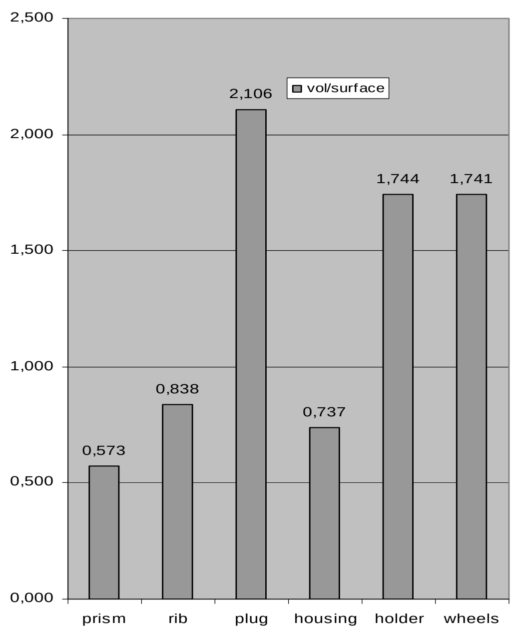

Fig. 15. Comparison between evaluations of the shape complexity based on the volume and surface ratio

Table 4.
Comparison between evaluations of the shape complexity based on the volume and surface ratio

MODEL	Evaluation of complexity (exact volume)	Evaluation of complexity (block volume)
Prism	0,573	1,136
Rib	0,838	9,500
Plug	2,106	6,076
Housing	0,737	2,788
Holder	1,744	43,367
Wheels	1,741	9,940

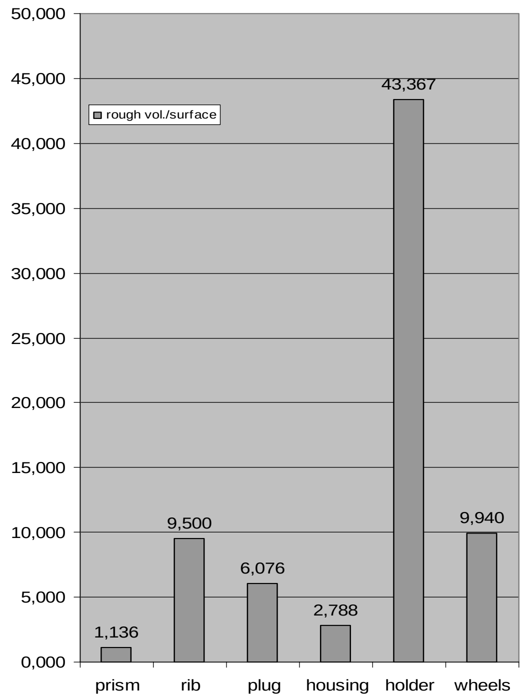

Fig. 16. Comparison between evaluations of the shape complexity based on the rough volume and surface ratio

Volume/surface ratio basically describes the quantity of curved or free form surfaces in a part. It also points the difference between thin walled and bulk parts.

7. Evaluation of shape complexity based on ratio between a minimal block volume and a parts volume

Block volume/volume ration also shows a difference between simple bulk parts and more complex free form surface parts (Table 5) and can be comparable to volume/surface ration (Figure 17).

Table 5.
Comparison between evaluations of the shape complexity based on the minimal bulk volume and volume ratio

MODEL	VOLUME RATIO
Prism	1,98
Rib	11,34
Plug	2,89
Housing	3,78
Holder	24,86
Wheels	5,71

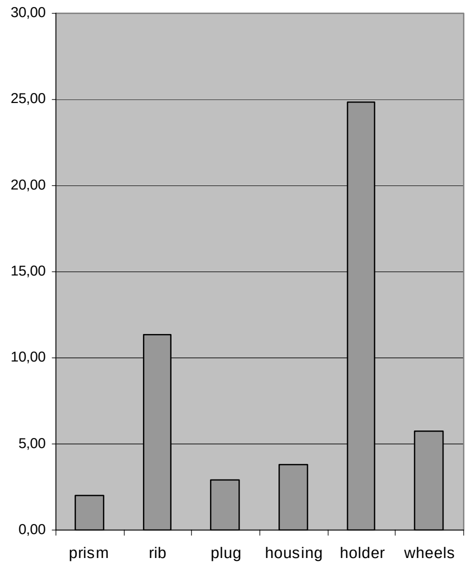

Fig. 17. Comparison between evaluations of the shape complexity based on the minimal bulk volume and volume ratio

8. Results commentary

Observing the results, the thesis presumed at the beginning established itself as being accurate. From all parts two can be pointed out that present the worst evaluation in most cases. The rib and the holder are both thin walled, with very high minimal block volume/exact volume ratio. The results of those two parts greatly differ from other results for every method except the "number of triangles method", which is volume independent. In a case of calculations based on minimal block volume the biggest error appears for rib and holder results because those parts have very large minimal block volume with extremely small actual part volume.

The experience based evaluation of shape complexity (Table 6) (purely geometrical without considering the manufacturing procedures) based on scale from 1 to 10 would be:

For basic shape evaluation four simple methods were based on simple and easily available information acquired from STL CAD data format. The subject of complexity has been investigated by many researchers [19-22] but with more complex approach, that is not suitable for STL files. The methods presented here use basic geometrical data values available from STL files and the ratios between those values.

Table 6.
Experience based evaluation of test models

MODEL	ESTIMATION
Prism	1
Rib	3
Plug	6
Housing	7
Holder	9
Wheels	10

First and the simplest method, based only on number of triangles, can be a very objective estimate, because the number of triangles in a parts STL file is not dependant on parts volume but only on the export resolution. A very accurate estimate can by made by volume/number of triangles method described in second method. Last two methods are very similar and the results are very close to experience based evaluation presented in Table 6.

But all methods presented are unable to determine the convexity or concavity and similar geometrical properties of parts that can be a key factor when considering conventional machining.

9. Application of methods

In a rapidly developing field of manufacturing technologies choosing the optimal manufacturing procedure is a difficult and crucial decision. Usually the decision is based on experience evaluation that is fast and can be optimal. Usually, this method produces goods results, but in some cases this method can lead to cost increases and reduced economic efficiency without us even knowing that. Therefore, it is crucial, that a fast and simple solution is developed, by which the optimal way of manufacturing can be determined.

The complexity from the manufacturing point of view depends largely on the manufacturing procedure used. Every manufacturing procedure has its properties and limitations that must be taken into consideration, when estimating parts complexity.

Use of method for basic shape evaluation can also be spread on other non conventional fields [23], where shape recognition is needed.

10. Conclusions

Four methods presented, make only small inroads into the subject of evaluating shape complexity. Their advantages lie in mathematical simplicity and intuitive use. However, the simplicity can also lead to some significant errors, especially when complex thin walled parts are in question.

Therefore, some new concepts of determining shape complexity are being developed that can lead to greater (perhaps even uniform) objectivity. One of these methods is based in the size of individual triangles of the parts STL file, enabling the slicing of certain part on layers of different complexity and using this in hybrid tools and products. The other method is based on the difference of angle between normals of adjacent triangles, greatly enhancing the possibility of determining edges, convexity and concavity. It also enables the evaluation of some small features in a certain part that can present a mayor problem for manufacturing. Combining the existing methods with those currently in development should give an accurate evaluation of shape complexity and should serve for determining appropriate manufacturing procedures regarding to the evaluation.

References

[1] F. Kimura, H. Suzuki, K. Takahashi, Product design evaluation based on effect of shape errors for part assembly, CIRP Annals 41/1 (1992) 193-196.

[2] T. Brajlih, I. Drstvensek, J. Balic, B. Katalinic, Casting of intake pipes by silicone rubber moulding, Annals of DAAAM for 2005 & Proceedings of the 16th International DAAAM Symposium, University of Rijeka, 2005, Opatija, Croatia.

[3] F. Cus, U. Zuperl, E. Kiker, M. Milferner, Adaptive controller design for federate maximization of machining process, Journal of Achievements in Materials and Manufacturing Engineering 17 (2006) 237-240.

[4] T. Grimm, Stereolithography, Selective Laser Sintering and PolyJet™: Evaluating and Applying the Right Technology, Accelerated Technologies, 2002.

[5] Z.D. Zhou, J.D. Zhou, Y.P. Chen, S.K. Ong, A.Y.C. Nee, Geometric Simulation of NC Machining based on STL Models, CIRP Annals 52/1 (2003) 129-134.

[6] B. Vaupotic, M. Kovacic, M. Ficko, J. Balic, Concept of automatic programming of NC machine for metal plate cutting by genetic algorithm method, Journal of Achievements in Materials and Manufacturing Engineering 14 (2006) 131-139.

[7] I. Pahole, M. Puc, B. Vaupotic, J. Balic, Comparicon of technology of forming the sheet metal by numerical simulations, Journal of Achievements in Materials and Manufacturing Engineering 18 (2006) 403-406.

[8] M. Brezocnik, M. Kovacic, M. Psenicnik, Prediction of steel machinability by genetic programming, Journal of Achievements in Materials and Manufacturing Engineering 16 (2006) 107-113.

[9] T. Brajlih, I. Drstvensek, M. Kovacic, J. Balic, Optimizing scale factor of the PolyJet™ rapid prototyping procedure by generic programing, Journal of Achievements in Materials and Manufacturing Engineering 16 (2006) 101-106.

[10] B. Vaupotic, M. Brezocnik, J. Balic, Use of PolyJet technology in manufacture of new product, Journal of Achievements in Materials and Manufacturing Engineering 18 (2006) 319-322.

[11] M. Weck, T. Nottebaum, Optimization of large composite structures, CIRP Annals 40/1 (1991) 411-414.

[12] C.R. Boër, J. El-Chaar, E. Imperio, A. Aval, Criteria for optimum layout design of assembly systems, CIRP Annals 40/1 (1991) 415-418.

[13] W.H. Wah, Introduction to STL format, 1999, http://rpdrc.ic.polyu.edu.hk/content/stl/stl_introduction.htm.

[14] Stereolithography, ‘STL (stereolithography) Files, STL file description, http://www.stereolithography.com/stlformat.php

[15] T. Nakagawa, A. Makinouchi, 3-D Plotting of finite element sheet metal forming simulation results by laser stereolithography, CIRP Annals 41/1 (1992) 331-333.

[16] B. Valentan, T. Brajlih, I. Drstvensek, J. Balic, Evaluation of shape complexity based on STL data, Journal of Achievements in Materials and Manufacturing Engineering 17 (2006) 293-296.

[17] I. Drstvensek, T. Brajlih, B. Valentan, J. Balic, Development of rapid prototyping of large hybrid tools, Maribor: Faculty of Mechanical Engineering Maribor, 2005, (in Slovenian).

[18] D. Rusin, Newsgroups: www.sci.image.processing, sci.math, Volume of a part in an .STL file, 1995.

[19] http://www.math.niu.edu/~rusin/known-math/95/volume.poly.

[20] Y. Chen, H. Sundaram, Estimating complexity of 2d shapes, Arts Media Engineering, Arizona State University, Tempe, AZ 85281, AME-TR-2005-08, (2005).

[21] N. Iyer, S. Jayanti, K. Ramani, An Engineering Shape Benchmark for 3D models, DETC2005/CIE-85612 Proceedings of ASME IDETC/CIE 2005, Proceeding of the 25th Computers and Information in Engineering Conference (CIE),Long Beach, California, 2005.

[22] D.V. Majstorovic, P. Bojanic, V. Milacic, CAD-CAI Integration for complex surfaces, CIRP Annals 41/1 (1992) 535-538.

[23] G. Lorenz, Principal component analysis in technology, CIRP Annals 38/1 (1989) 107-110.

[24] B. Brecko, S. Brezovnik, Protection of a voice guided wheelchair with the use of fuzzy logic, Proceedings of the 14th International Conference of Electronic and Computer Science ERK 2005, Portoroz, Slovenia, 2005, 389-390 (in Slovenian).