



Converting Topology Optimization Results into Parametric CAD Models

Shane Larsen¹ and C. Greg Jensen²

¹Brigham Young University, shanelars@gmail.com

²Brigham Young University, cjensen@byu.edu

ABSTRACT

A semi-automatic method for converting topology optimization results into a smooth, parametric CAD model is presented in which predefined 2D shape templates are fit to the optimization results and a surface is swept through the 2D templates to create a 3D feature. This method is repeated for each feature until a full 3D model is generated. This approach to topology optimization results interpretation reduces the time required to utilize the results, helps to standardize the interpretation process, and enables the designer to have control over the trade-off between closeness of fit to the theoretically optimal model and feature complexity.

Keywords: topology optimization, shape recognition and templates, polar mapping.

DOI: 10.3722/cadaps.2009.407-418

1. INTRODUCTION

Topology Optimization is a process used widely in industry to create low weight mechanical parts with specific performance attributes. Using Topology Optimization in the conceptualization phase of product design avoids many problems that can occur later in the design process [4]. It also allows the designer to find optimal designs that are, many times, unintuitive or difficult for a human to visualize [8]. This ability can create an important strategic advantage in the competitive market of engineering design.

There are many methods for conducting Topology Optimization. One common method utilized in many commercial software packages is known as the “Soft Kill Method” [4]. This method involves an iterative process in which the Young’s Modulus of each element of a finite element model is pushed toward zero while still maintaining specified part performance criteria. This effectively eliminates elements from the model which are bearing little or no stress under the defined loading conditions. After these unstressed elements are eliminated the process iterates again and continues eliminating elements until only stressed elements remain. This ensures an optimal part design based on given boundary conditions and loading conditions. Throughout this paper, Topology Optimization (TO) refers to the Soft Kill Method of topology optimization.

1.1 Integrating Topology Optimization and Parametric CAD

The term “parametric” refers to a CAD model that has predefined parameters that can be easily changed and adjusted. This may include any length dimensions, angle dimensions, materials, load magnitudes, tolerances, etc. These parameters can be linked to different types of analyses so that the effects of changes in the parameters can be seen instantly and automatically. This becomes a great advantage over finite element representations of parts when updates from one area affect model performance in another area. With a parametric part, these updates can be incorporated into the

master model and the effect of the update on all other aspects of the design (i.e. aesthetics, mechanical design, manufacturing, packaging, flow dynamics, finance/economics, etc.) can be assessed easily and automatically. A typical design process which utilizes TO and 3D parametric modeling is illustrated in Fig. 1. For this type of design process to work well, the simplest set of control parameters must be identified in step 3 and used to represent the resulting model. As the number of parameters increases, the complexity of the system increases and its overall utility decreases. For this reason TO results must be reduced to easily manufactured, parametric shapes that are common to engineering design (holes, extrusions, fillets, radii, etc).



Fig. 1: Typical design process utilizing both TO and 3D parametric modeling.

Unfortunately, TO requires a finite element model, the results of which are represented using triangular faceted surfaces. These surfaces are commonly referred to as a tessellation object and merely approximate the boundary of the optimal part. A tessellation cannot be converted directly into a parametric CAD part. The current industry standard for processing TO results is to have a human designer manually create a parametric CAD model that approximates the tessellation. This is a tedious and time consuming process which is highly dependent on the individual designer's interpretation of the TO results. The research presented in the paper focuses its scope entirely on step 3 of the process outlined in Fig. 1. After the conversion is performed, Parametric Optimization (step 4) refines the design further and restores much of the performance loss created by the geometric approximation that will be described below. By creating a link between TO results and a parametric 3D CAD model, significant time savings can be realized while creating a more repeatable, standard process of optimization and part design.

1.2 Recent Advances

Several different approaches to solving this problem have been tried in the recent past. The first method involves creating B-Spline cross sections through standard gray level image interpretation that closely match the optimal part at different locations throughout the part. These cross sections are then linked together to form a 3D CAD model of the optimal topology. Tang & Chang [1] followed this methodology to automatically create a parametric CAD model of a weight optimized Roadarm of a tracked vehicle. They created each B-Spline cross section with the same number of control points and the same order B-Splines so that a NURBS surface could easily be generated from the B-Splines. Hsu and Hsu [2] followed a similar process and added some data refinement so as to avoid small bits of meaningless data in the TO results that would impede the recognition process. They were able to successfully create 3D CAD models of simple cantilevered beam optimization problems with well known structures.

The second approach utilizes various forms of mesh refinement, reduction, and deformation to generate a 3D model based on an ambiguous point cloud [6][7]. This method was developed for creating 3D models from data obtained through a laser scanning device and has not been directly applied to the interpretation of TO results. However, a point cloud can easily be obtained from TO results (see section 2.7.1) upon which the mesh reduction methods can be applied. These methods, in general, take a complex point cloud or mesh and convert it into a simpler mesh that represents the geometry equally well with fewer data points. These mesh reduction methods have been successfully utilized to recreate CAD models of simple parts from point cloud data. Mesh reduction methods do not typically create a 3D feature based CAD model that is useful for parametric optimization or manufacturing process planning.

1.3 Current Method

Although both of the previously mentioned methods were very successful, they each have certain drawbacks that are not ideal for use in product engineering and design. Some of these drawbacks include the lack of the ability to make simpler types of cross section shapes which require fewer defining parameters than B-Splines, the lack of user interactivity so as to guarantee that the proper interpretation of the TO results is occurring, and the lack of a logically structured feature operation tree at the end of the refinement process. If these abilities could be included in the TO results interpretation process, designers would be able to work with the part in a more intuitive way and the part would be more usable for future design steps.

To include more of the above mentioned capabilities in the design process it is proposed that a method developed by Lin and Chao [3] for use in 2D topology optimization interpretation be extended to 3D. Lin and Chao proposed the use of Shape Templates that could be fitted into gaps that are recognizable within the TO results. These Shape Templates are created in advance based on the level of complexity one is willing to utilize in the results interpretation. If a high level of detail is needed, a more complex template must be defined. To create a shape template, a suitable, generic, convex shape is drawn and the distance (L) between a reference point within the shape and the edge of the shape is measured. This is repeated for various points around the periphery of the shape at different angles (θ). A polar map of the shape can then be graphed as shown in Fig. 2.

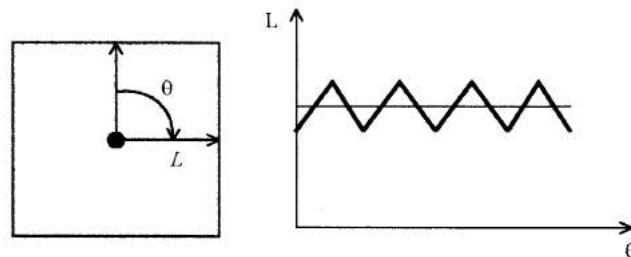


Fig. 2: Square shape template polar map.

Once the desired templates are defined, samples from the TO results can be taken and compared to each of these templates and the best fit approximation to the TO results can be determined with a simple Least Squares comparison.

To extend this method to 3D, several adjustments must be made, some of which come from concepts introduced by Osada and Funkhouser in their work in 3D shape similarity matching using Shape Distributions [9]. First, since the orientation of the hole is no longer fixed due to a 2D alignment, an orientation direction, vector, or geometry must be obtained. Second, with the added complexity of a 3D shape, the feature or hole that is being recognized by the algorithm must be identified or segmented out from the rest of the model. Both of these adjustments are done with interactive software tools that allow the user to specify the feature orientation geometry and select the faceted surfaces provided by the TO results that will define a single feature.

At this point it may be useful to define exactly what is meant by a “feature”. Throughout this paper, the term “feature” is used in reference to a set of cross sections that define a surface in 3D. Each of these cross sections must be a closed loop so that the resultant 3D surface can be closed and subsequently used in a Boolean operation on the original solid model design part to add or subtract material. The cross sections must be planar, but there is no need for the set of cross sections to be parallel. Fig. 3 shows three examples of shapes that can be made using this modeling method. As can be seen, a wide array of 3D shapes can be represented by a set of cross sections linked together to form a single feature. The cross sections used to define the feature are highlighted in orange.

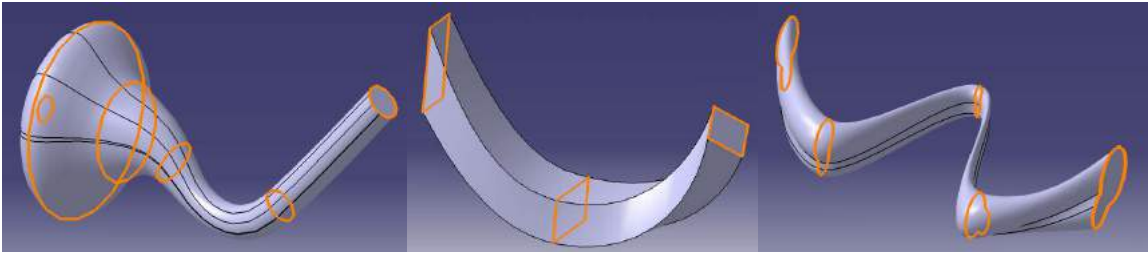


Fig. 3: Feature examples with defining cross sections highlighted.

This method is entirely CAD based and is intended to be utilized within a CAD package. It is a feature by feature process which must be repeated for each feature of a part. It can be used to define a hole to be removed, or a solid feature to be added to the model. There are 9 steps in the process and the entire process flow is as follows:

1. Generate an IGES file containing the faceted surfaces of the TO results
2. Import the design space part and overlay with the faceted surfaces
3. Identify feature surfaces among the faceted surfaces of the IGES file
4. Identify design space surfaces that the feature intersects
5. Signify feature orientation geometry
6. Specify the number of cross sections to use
7. Sample the model at each cross section location
8. Compare samples to polar maps of defined shape templates
9. Utilize CAD API to cut geometry away from design space part

2. MAIN BODY

The following is a description of the general process flow proposed for converting TO results into a parametric CAD model through the use of shape templates as defined by Lin and Chao along with extensions of this method that allow it to be utilized in 3D space.

2.1 Generate IGES File

In order to have a CAD-centric application that designers are comfortable with, the TO results must first be read into the CAD package. The easiest way to do this is by using the IGES file format. Since the data to be represented is merely triangular surfaces, there are rarely any errors throughout the IGES import step as IGES is very reliable at simple surface representation. These surfaces can now be used to create a more robust parametric CAD model of the optimal part design

2.2 Incorporate the Design Part

The design part represents the maximum volume that the resultant part can occupy and still be considered feasible. Based on the surfaces that were imported in the previous step, features (as defined above) will be cut out of the design part until the optimal part is all that is left over. For ease of visualization, it is helpful to overlay the design part of the TO on top of the faceted surfaces of the IGES file. This allows the designer to interact with the part in a very intuitive manner in order to see if the updated part fits what the optimal part looks like. Most CAD packages allow this to be done. The design part will completely envelope the IGES surfaces from the TO results and therefore, the transparency of the design part should be adjusted so that both the design part and the optimal part representation can be viewed simultaneously.

2.3 Feature Segmentation

Once both parts are in the same CAD environment and can be viewed simultaneously, each feature of the optimal part can be cut out of the design space part one by one until the design space part matches the optimal part. To do this, the user must first define which of the faceted surfaces are to be included in the definition of a single feature. This is done using CAD API multi-selection user tools that are

included in most CAD packages. In Fig. 5, the polygon selection trap multi-selection tool is utilized to select a group of faceted surfaces that define a single hole feature to be cut out of the design part.

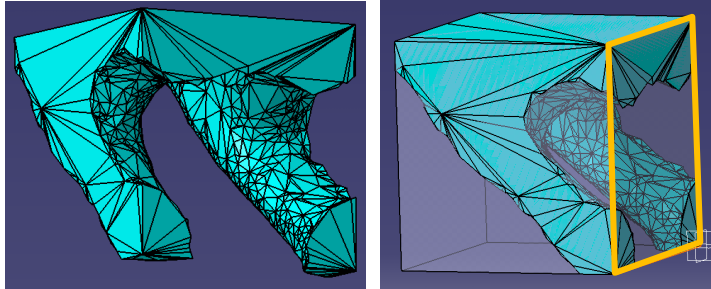


Fig. 4: Imported IGES Geometry and overlay of design space.

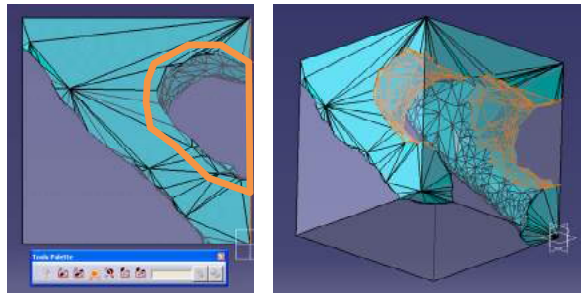


Fig. 5: Feature segmentation by user selection of faceted surfaces.

2.4 Identify Design Space Intersection Surfaces

In order to obtain the optimal feature fit, it is useful to know if the hole feature actually intersects and removes portions of the design space surfaces. In this example, the hole feature intersects one of the surfaces of the design space. This surface is highlighted in orange in Fig. 4. **Error! Reference source not found.** Note that the term “intersection surface” refers only to surfaces that the feature intersects but the feature orientation geometry specified by the user does not intersect. In this case the two design space surfaces at either end of the feature will be intersected by the orientation geometry, and need not be specified as “intersection surfaces”.

2.5 Signify Feature Orientation Geometry

The Feature Orientation Geometry is specified by the user through interactive selection. It is used to obtain a Guide Curve for the feature. Each cross section, in turn, obtains its normal vector based on the tangent vector of the guide curve. For example, if the user selects a 3D curve as the Feature Orientation Geometry, then the curve can be used directly as the guide curve. If a cylinder is selected, the guide curve can be inferred as the axis of the cylinder. If the user specifies three cross sections for the feature, the middle cross section would be located at the midpoint of the guide curve based on the arc length of the guide curve. It would then obtain a normal vector based on the tangent vector of the guide curve at that point (see Fig. 6). The use of guide curves gives this method the flexibility it needs to operate in 3D. The orientation geometry is also used in the cross section sampling step and will be described in greater detail in that section. For simplicity in this proof of concept, straight lines were used as feature guide curves.

2.6 Specify the Number of Cross Sections

The user specifies the number of cross sections to use to define each feature. The more cross sections specified, the closer the match to the actual optimal part. However, adding more cross sections significantly increases the processing time and complexity for subsequent steps in the design process.

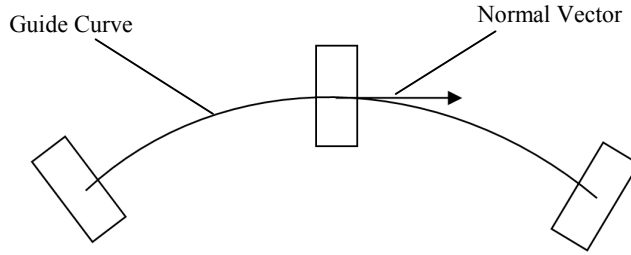


Fig. 6: Feature curve and cross section normal vector.

2.7 Sample the Model at Cross Section Locations

In order to determine which of the defined shape templates matches the TO results best, the model is sampled at various locations and compared to the different shape templates. Sampling is performed in 4 steps:

1. Uniform Point Cloud Generation
2. Point Cloud Segmentation
3. 2D Point Cloud Projection
4. Polar Map Creation

2.7.1 Uniform Point Cloud Generation

Because there are no bounds on the possible sizes of the faceted surfaces that result from TO, a more uniform representation is needed so as not to weight one piece of the geometry more than other pieces of the geometry. For this reason, a random uniform point cloud is generated on the faceted surfaces that define the feature. This is accomplished on a surface by surface basis. The number of points to generate on each triangular surface is calculated based on a point cloud density parameter according to the following equation:

$$NumP = Area \cdot \rho_p \tag{2.1}$$

Where $NumP$ is the number of points to be generated, $Area$ is the area of the particular surface, and ρ_p is the point density parameter set by the user. $NumP$ points are then generated in random locations on the surface according to:

$$P = (1 - \sqrt{r_1})A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C \tag{2.2}$$

where P is the point to generate, r_1 and r_2 are randomly generated numbers between 0 and 1, and A , B , and C represent the vertices of the triangular surface (see [9] for a full definition of this equation). Generating the points according to equation 2.2 guarantees that a uniform, random point cloud results. This same process is performed on each surface identified as a Feature Surface until a random, uniform density point cloud representing the individual feature is created. Fig. 7 shows the point cloud generated on the example feature.

2.7.2 Point Cloud Segmentation

The feature point cloud is now segmented into the appropriate number of cross section point clouds ($NumXSec$). In this example, $NumXSec$ is set to 2 for simplicity. The feature point cloud is split into 2 equal portions based on the orientation geometry provided in an earlier step. In this example, the orientation of the feature is defined by a straight line and, therefore, the guide curve of the feature is equal to the normal vectors of each of the 2 cross sections that will be generated.

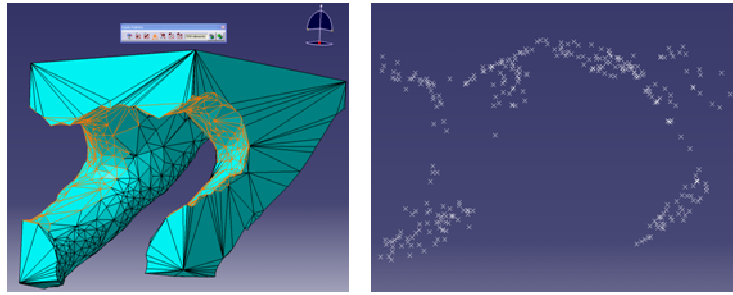


Fig. 7: User selected surfaces and resulting uniform point cloud.

2.7.3 2D Point Cloud Projection

The cross section point clouds are projected to a mid plane along the cross section normal vector to yield a set of points on a plane which can then be sampled. Fig. 8 shows the segmentation and projection steps graphically as well as the resulting 2D projected point cloud representing a single cross section of the feature.

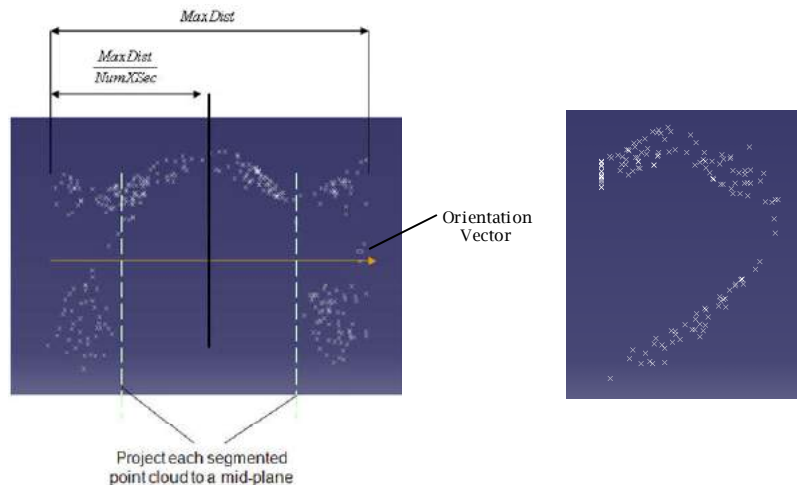


Fig. 8: Point cloud segmentation and 2D cross section projection.

2.7.4 Polar Map Creation

The 2D projected cross section is sampled by calculating the angular location of each of the randomly generated points with respect to some arbitrary reference point on the plane. If the sample point falls in a specified θ "bin" it is averaged with all the other points in that bin and the average point of each bin is used to create a polar map of the cross section as described by Lin and Chao [3]. The reference point used could be an average point, or a centroid or some other arbitrary point. Although it is more numerically stable to have a point close to the center, the algorithm is theoretically indifferent to the position of the reference point.

Many times, as in this example feature, there are large angles with no data provided by the TO results. This creates ambiguity for the algorithm and can cause problems. To remedy this, it is suggested that when a certain number of consecutive θ bins are encountered that contain no point data (there are 4 bins in a row with no data in Fig. 9), each of the user specified intersection surfaces are tested to see if the current direction intersects them. If it does, a point is generated at the intersection and used to represent that angular position in the polar map of the cross section. The effect of this practice is shown graphically in Fig. 10. With this added data, the entire cross section is represented and much of the shape ambiguity is reduced to the point that the shape templates can be fit to the desired feature.

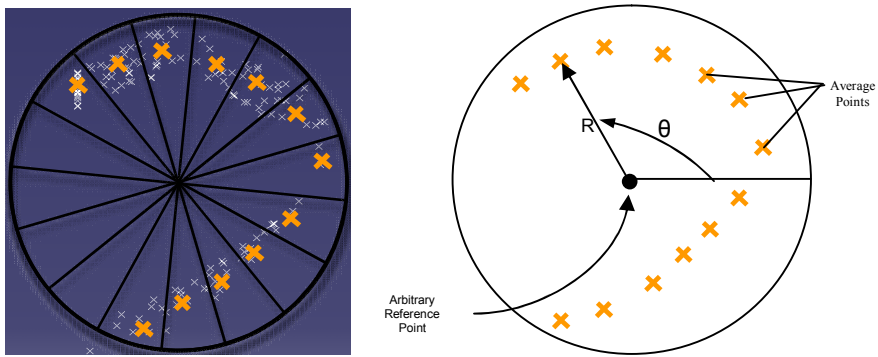


Fig. 9: Polar map generation process.

Once the sampling is complete, the polar map is created by plotting the angular location of each of the average points (θ) against the distance from the arbitrary reference point. The final polar map of the example cross section is shown in Fig. 10.

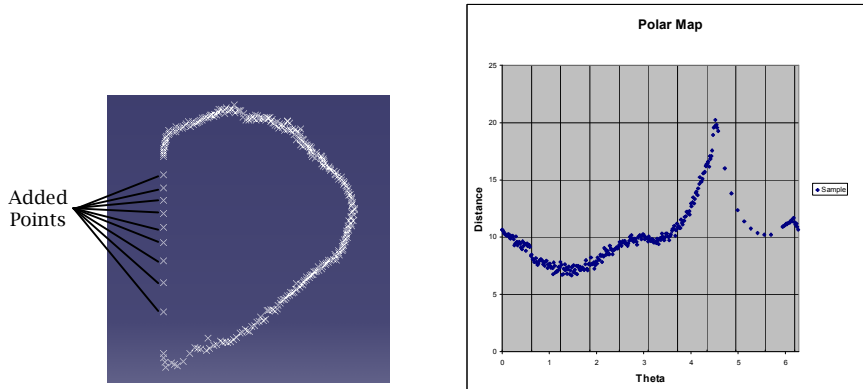


Fig. 10: Final polar map of the example cross-section.

2.8 Shape Template Selection

Comparison between the sample and the defined Shape Templates is performed in 4 steps: 1) Finding the sample peaks, 2) Parametric Template Generation, 3) 2D Fitness function, and 4) 3D Fitness Function.

2.8.1 Finding the Sample Peaks

At this point, the location of the vertices of the best fit shape template can be pulled directly from the sample polar map. Depending on how many vertices each shape template needs to be constructed (i.e. three for a triangle, four for a quadrilateral, etc.) an algorithm runs through the points of the polar map and identifies the peak points of that particular sample. These peak points can then be used to generate best fit shape templates for direct comparison with the sample feature.

2.8.2 Parametric Template Generation

Since the data obtained from the TO results will never fit a shape template perfectly, the shape templates must be parameterized in such a way that they can stretch or shrink to fit the sample data while still holding to their geometric definitions. In this approach, the parameterization is accomplished through 2D polar points. The peaks identified in the polar map act as the parameters to scale the shape templates for direct comparison with this particular feature. Since there is a closed

form mathematical definition for each of the shape templates, a polar map of the templates can be generated using the exact same arbitrary reference point that was used to generate the sample polar map. By using the same reference point, the templates are then able to be compared to each other by overlaying the two templates. Fig. 11 demonstrates graphically how this comparison is done. It is obvious from the graph that the Quadrilateral template defined in this example is the best fit for the sample because the points of the Quadrilateral template are closest to the sample points. However, this comparison must be quantified in order to be useful.

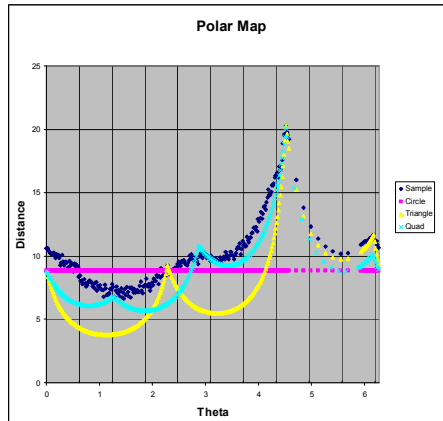


Fig. 11: Shape template comparisons.

2.8.3 2D Fitness Function

To quantify the fitness of each of the generated templates, a simple least squares approach was used. For each point, the difference between the sample distance and the template distance from the arbitrary reference point was squared and summed up. Although the single best fit template for this individual cross section can be identified using this 2D fitness function, it is unknown if this template will be appropriate to represent the entire feature. The scores of all the templates are retained for further analysis of the 3D fitness.

As was mentioned before, there are sometimes large areas where only approximated data is available because the TO results provide no direct data. In Fig. 11 this area is located between $\theta = 4.5$ and $\theta = 5.7$. It is easy to identify these areas due to the uniformity of the point data. Notice that, although shape template data could be generated for every angle θ , the shape templates only have data at positions where there is sample data to compare to. If this were not the case, the fitness functions would no longer be valid.

2.8.4 3D Fitness Function

Once each cross section has been compared to each defined shape template, the overall 3D fitness score can be calculated. Tab. 1 shows the 3D fitness of each of the defined templates for the example feature using 2 cross sections. It is apparent that the Quadrilateral template fits this shape the best because it has the lowest fitness score (remember that the fitness is the sum of the squared errors).

	Circle	Triangle	Quadrilateral
CS 1	1200	4000	600
CS 2	1450	3500	1250
3D Fit	2650	7500	1850

Tab. 1: 3D fitnesses for each defined shape template.

Determining that the quadrilateral has the best fit is not actually that useful since a quadrilateral will always fit better than a triangle because a triangle is simply a quadrilateral with 2 coincident vertices. In general, the more defining parameters a template has, the better it will fit the sample. However, if a triangle fits the shape “good enough”, any designer would prefer to use a triangle rather than a quadrilateral. Giving the designer the ability to define “good enough” is essential so that the designer can have full control over the level of complexity of the design. More important than finding the template that fits best, therefore, is finding the simplest template that fits “good enough”. A simpler template means fewer parameters to have to keep track of which can translate into huge computational savings during subsequent parametric optimization and other analyses.

One way to decide which template is “good enough” is through a pair wise comparison. Tab. 2 presents a pair wise comparison of the three templates defined in the example. The table should be read in this manner “the circle template (template name from the top row) has a 65% better fitness than the triangle template” (template name from the left column). In this case, only the quadrilateral is better than all other templates. However, using a quadrilateral as compared to a circle means the addition of 12 parameters per cross section. In this case, that means 24 more parameters! Imagine what that could mean in a more complex feature that uses 5-10 cross sections. It is, therefore, important for the designer to know the magnitude of the tradeoff between fitness and number of parameters. Only the designer has the knowledge and intuition to know if a 30% increase in fitness is worth the cost of keeping track of 24 more parameters.

% Change	Circle	Triangle	Quadrilateral
Circle	0	-183%	30%
Triangle	65%	0	75%
Quadrilateral	-43%	-305%	0
# Parameters / Cross Section	4	12	16

Tab. 2: Pair wise comparison of template fitnesses.

2.8.5 Non-convex Shape Templates

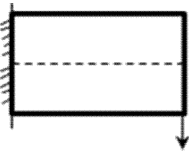
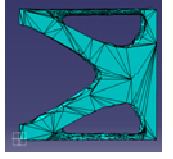

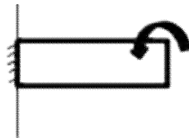
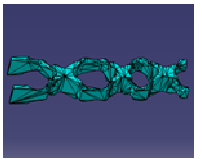
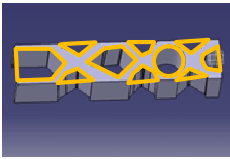
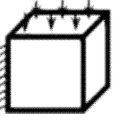
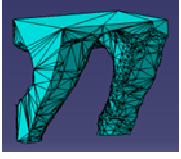
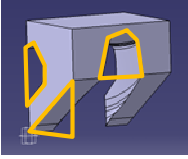
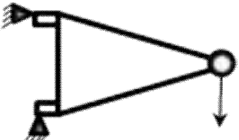
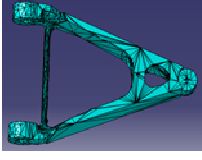
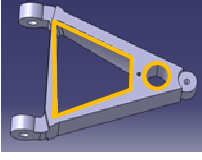
This parameterization scheme has a definite drawback, however, in that it can only be applied to convex shapes. A non-convex shape template can be made, but this issue is not addressed in the current research. (see section 4.1)

2.9 Add/Subtract Geometry

Now that the most appropriate cross sectional shape template has been identified, each cross section is created in the CAD package in the proper location. Any number of methods can then be used to sweep through the cross sections to create the feature including NURBS surfaces or other swept surfaces typical to CAD software.

2.10 Completed Examples

Tab. 3 presents the results of applying the above described methodology to 4 cases. Case I is a cantilevered beam with a symmetry constraint down the center line. Case II is a cantilevered rectangular bar under a single torque load applied at the end of the bar with a draw direction manufacturing constraint. Case III shows a cantilevered cube under a pressure load. Finally, case IV depicts one loading condition of a control arm of a car suspension. Each of these TO cases were done using OptiStruct, after which the results were imported into CATIA V5 R18 where the TO results refinement process was performed. OptiStruct has integrated tools for generating IGES files and dealing with many types of manufacturing constraints such as draw directions (case II) and symmetry lines (case I).

Loading Conditions	TO Results (Optimal Part)	Parametric Model	# Features	% of Optimal Volume
I 			4	133%
II 			10	158%
III 			3	101%
IV 			2	126%

Tab. 3: Examples of refined TO results.

3. CONCLUSION

Although this research is in its infancy, it has proven that the proposed method can be used as a tool in the process of TO results interpretation to provide an intuitive environment for refinement, quantify fitness, control tradeoffs, and simplify model parameters. As part of this method, Shape Templates provide a simpler alternative to the use of B-Splines in order to produce a robust parametric 3D CAD model that can be used for future design or manufacturing steps. The use of Shape Templates, in many cases, will reduce the number of needed defining parameters thus simplifying the model and reducing computational time in future optimization runs and design steps. The proposed method allows the designer to utilize intuition and experience, while guiding this intuition with quantitative data and control structures. The method also significantly reduces the time required to process the TO results.

4. FUTURE WORK

This section introduces promising concepts that were outside the scope of this paper, but are areas where more in-depth research is needed for the advancement of this field in general.

4.1 Non-convex Shape Templates

One of the major drawbacks of the shape templates used in this study is that they were all convex polygon shapes. There are many instances in the real world that a non-convex shape would be ideal. In this study, convex shape templates were used in order to simplify the generation of the polar maps of the defined shape templates. It may be possible, however, to define a more generalized shape template that can be concave or convex thus allowing for the definition of concave shape templates for certain specific applications or industries if need be. This could be done by utilizing more than one interior

reference point and defining the relationship between the two of them. In essence each point would cover a convex portion of the set of points and then the full cross section can be pieced together based on the relationship between the two (or more) interior arbitrary reference points. Fig. 12 depicts a traditional convex shape template as was used in this study, followed by the proposed new definition of a generalized concave shape template.

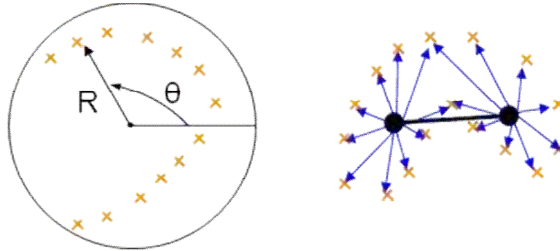


Fig. 12: Convex and concave shape templates.

4.2 B-spline – primitive Relationships

Although B-Splines require more defining parameters than simple polygon shapes, they also offer more adaptability. From a mathematical perspective, in fact, polygon shapes are nothing more than a special case of a periodic B-Spline (more or less). If another layer of abstraction is added to a B-Spline that defines relationships between another set of control points that would represent control points of a more simple shape (such as a polygon), then that would effectively reduce the number of parameters required to define the shape. For example, a circle could be constructed using a NURBS curve and it would require more than 3 parameters, which is all you would need in theory to define a circle (center x location, center y location, and radius). In fact it would most likely require 8 or more parameters since a NURBS circle would likely use 4 control points, each with an x and y parameter. However, a pseudo circle center could be used and a relationship created to define the locations of the 4 control points of the NURBS curve based on the pseudo circle center and some kind of scaling factor that is analogous to a radius. If this is done a NURBS curve can be used to define a “simplified” cross section. The advantage of this method would be to allow for the adaptability of the NURBS curves if needed or wanted, while not sacrificing the simplicity of the model if the fitness is “good enough” as is.

5. REFERENCES

- [1] Tang, P.-S.; Chang, K.-H.: Integration of topology and shape optimization for design of structural components, *Struct. Multidisc Optim.*, 22, 65-82, 2001.
- [2] Hsu, M.-H.; Hsu, Y.-L.: Interpreting three-dimensional structural topology optimization results, *Computers and Structures*, 83, 327-337, 2005.
- [3] Lin, C.-Y.; Chao, L.-S.: Automated image interpretation for integrated topology and shape optimization, *Struct. Multidisc Optim.*, 20, 125-137, 2000.
- [4] Eschenauer, H. A.; Olhoff, N.: Topology optimization of continuum structures: A review, *Appl Mech Rev*, 54(4), July 2001.
- [5] Blattman, W. R.: Generating CAD Parametric Features Based on Topology Optimization Results, MS Thesis, Brigham Young University, Provo, UT, 2008.
- [6] Bernardini, F.; Bajaj, C. L.; Cheny, J.; Schikore, D. R.: Automatic Reconstruction of 3D CAD Models Department of Computer Sciences, Purdue University, West Lafayette, IN, 1999 <http://www.cs.purdue.edu/research/shastra>.
- [7] Jain, A. K.; Zhong, Y.; Lakshmanan, S.: Object Matching Using Deformable Templates, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18-3, 267-278, 1996.
- [8] Campelo, F.; Ota, S.; Watanabe, K.; Igarashi, H.: Generating Parametric Design Models Using Information from Topology Optimization, *IEEE Transactions on Magnetics*, 44-6, 986-989, 2008.
- [9] Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D.: Shape Distributions, *ACM Transactions on Graphics*, 21(4), 807-832, 2002.