COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 6, 885–898
http://dx.doi.org/10.1080/16864360.2016.1168238

Leveraging feature information for defeaturing sheet metal feature-based CAD part model

Yogesh H. Kulkarni[a], Ravi K. Gupta[b], Anil Sahasrabudhe[a], Mukund Kale[c] and Alain Bernard[b]

[a]College of Engineering Pune, India; [b]Ecole Centrale de Nantes, France; [c]Siemens PLM, India

ABSTRACT

Complex models prepared in CAD applications are often simplified before using them in downstream applications like CAE, shape matching, multi-resolution modeling, etc. In CAE, the thin-walled models are often abstracted to a midsurface for quicker analysis. Computation of the midsurface has been observed to be effective when the original model is defeatured to its gross shape.

Defeaturing in this paper proposes a novel approach for computation such gross shape and it works in two phases. First, a proposed sheet metal feature-based classification scheme (taxonomy) is used to determine the suppressibility of the features. Second, a method based on the size of remnant portions of the feature volume is developed to determine the eligibility for suppression. Case studies are presented to demonstrate the efficacy of the proposed approach. It shows that even after substantial reduction in the number of faces the gross shape retains all the important features needed for computation of a well-connected midsurface.

KEYWORDS

CAD; defeaturing; sheet metal features taxonomy; feature-based design

1. Introduction

Humans, while looking at an object, at the first glance perceive the gross shape and then eventually look into more details as needed [19]. This process of simplification of the shape, by ignoring the small and irrelevant details is widely used in geometric computations and is referred to as "Defeaturing." It is the process of suppressing some features based on certain predefined objectives or criteria.

Many existing simplification methods recognize small, irrelevant features on a mesh or a solid body first, then remove them to get the simplified (called 'defeatured') model. Instead, if a Feature-based CAD model is used as an input, then it has the advantage of the availability of ready features, so that the suppression and healing becomes relatively straightforward and robust. In such a feature-based defeaturing method, the primary challenge is the identification of the suppressible features. In the past, the suppressibility used to be based on some insufficient criteria, like using full feature parameters, selecting all the negative features, etc.

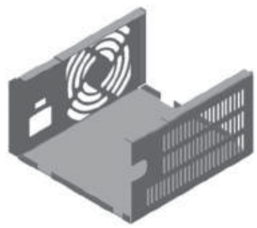
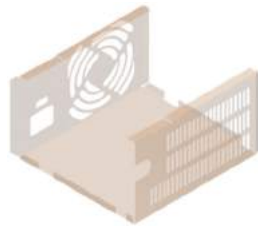
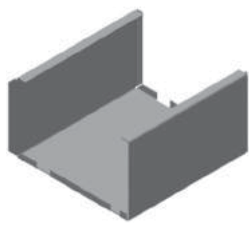
Defeaturing is primarily used in CAE analysis where such simplified models lower the complexity of the finite element mesh and thus reduce the analysis time. It is also used in shape matching & retrieval, fast visualization, hiding proprietary details, transmission across network, finding gross shape, etc. This paper focuses on using defeaturing for finding the gross shape needed in the computation of a medial form, called "Midsurface" [21]. Gross shape is the principal shape that "represents" the given shape but with far lesser features. Gross-ness depends on the size criteria, say, 5% of the total volume/area. Features having sizes below this are the candidates for suppression. With lesser irrelevant details on the input model, the generated midsurface becomes more representative of the original part and the computation becomes robust (small deviations/features in the input do not affect the output in appreciable manner).

Tab. 1 shows the result of an experiment to see whether the defeaturing helps in the quality of the midsurface. The first row shows the original model, its midsurface (with a problem shown in red circle) and the same problem shown in a magnified manner. The second row shows the defeatured model, its midsurface and mentions quality of the midsurface output. The third row shows that even though the defeaturing has positive effect on the midsurface output, it should be done judiciously to retain enough features in the midsurface as well.

Defeaturing has largely been a manual and tedious process. Small and irrelevant features are first recognized in the mesh/solid body based on the heuristic rules and removed [5]. Finally the model is healed to form a closed

CONTACT Yogesh H. Kulkarni kulkarniyh12.mech@coep.ac.in

Table 1. Effect of Defeaturing on the Midsurface Generation.

Part & actions	Model	Midsurface	Problems
Original model			Gaps and missing surfaces 
Manual defeaturing of the small irrelevant features.			No major errors in the midsurface such as gaps, missing surfaces, etc. Such output is desirable.
Defeaturing all the negative features blindly.			Midsurface does not have errors, such as gaps or missing surfaces but it is missing some important features.

watertight shape. Users view this task as too extensive and resort to recreate the necessary geometry than to simplify the existing one [10].

Objective of this work is to develop a computational method to automatically find simplified geometry of a sheet metal part model which can be used for further computations such as the generation of the midsurface. The method works in two phases. In the first, the objective is to identify and classify sheet metal features into suppressible and non-suppressible features, based on type and size criteria. Such classification has been presented in the form of taxonomy. In the second phase, a generic method based on geometric reasoning is proposed to identify the suppressible features.

2. Related work

Research in simplification of geometry has been going on for decades. Andújar [2] surveyed various geometric simplification techniques with input as mesh models, and for applications such as visualization, interface detection, visibility analysis, and transmission of models. He stated that the geometric fidelity and entity reductions are opposite goals, and thus, choice of a validation criterion dictates the quality of the output as well as appropriateness to the end application.

Kim et al. [15] demonstrated the use of operators like wrap-around and smoothing to fill up small holes, concave shapes and protrusions. The approaches presented in that work appear simplistic and would not get scaled for the complex parts.

Defeaturing is one of the most popular techniques of simplification for CAE analysis. Thakur et al. [25] surveyed and classified various techniques used till then, into four categories such as surface-based, volume-based, feature-based, and dimension-reduction. Most methods were based on mesh and solid as the input, with very few based on the feature-based CAD model. The work presented in the current paper is based on defeaturing with feature-based CAD model.

Feature-based design represents a CAD model in terms of a feature tree where each tree-node is termed as a *feature*. Features not only present modeling in the application-specific-vocabulary to the designers, but also provide parametric editing capabilities. The designers can update the models based on critical driving parameters. They can also suppress certain features and update the model to regenerate the shape. Such capabilities are (and can be) immensely leveraged in the defeaturing algorithms. Another advantage of using feature-based design is in the regeneration of the original model. As during defeaturing, features are only suppressed and not deleted. So, if there is a need to regenerate the original model, the suppressed features can simply be unsuppressed.

Nowadays, many CAD modelers are based on the feature-based design paradigm and provide access to the feature tree. Feature-based defeaturing operations have better applicability than mesh or solid based simplification methods for product design and engineering applications [13]. They also provide ready APIs (Application Programming Interfaces) to exclude/suppress certain features while computing the final shape. Feature based defeaturing methods are reviewed in the following section.

2.1. Feature-based defeaturing

Dabke [6], through the concept of 'global idealization', was one of the first researchers to leverage the feature information for defeaturing. This method was based on the expert system with heuristic rules derived from the analyst's experience. The proposed approach appears to be rudimentary in the usage of features.

Lee [16–18] elaborated a method to reorder the design features in the history tree and then to re-execute the history of the reordered features up to the given level of simplification. Since Brep (Boundary representation) re-evaluation is computationally intensive, he used the cellular model for increasing the performance. One of the major limitations of this approach is that once the model is converted to cellular, its feature update capability would cease to exist, making it difficult for bidirectional change propagation.

Smit [23] surveyed various approaches for CAD-CAE integrations and concluded that since features carry domain-specific information, they can bring context-relevant defeaturing. With this ease, there also comes one complication. Many features are built using entities from the existing features, creating dependencies called "parent-child" relationships. Suppressing a parent feature suppresses the child features too. Deciding the eligibility of suppression of the feature should thus include child features too. Otherwise, one has to build/adopt the part in such a way that the dependencies are removed or rerouted [1].

Feature trees may be different for similar final shapes. So, defeaturing based on the feature tree can yield different results. To achieve consistent defeaturing results independent of the modeling history, some attempts use just the final solid (Brep) model, and then recognize the features [26] to be removed directly from that final model. Even if there are a few limitations on the usage of the feature tree as stated above, having ready feature information is far convenient for defeaturing than doing feature recognition first and then suppressing the qualified ones.

Woo [26] proposed a new method which has subtractive features recognized directly from the final solid, thus eliminating the problem of looking at the history tree. It lacked coverage in terms of variety of features being recognized and thus, was limited to simple shapes.

Hamdi [11] et al. also surveyed simplification techniques and classified them based on the input format, features simplified, defeaturing criterion, advantages, limitations and application domain. Most of the methods were Brep-based and suppressed features like holes, chamfers, fillets, protrusions, depressions, passages, concave regions, etc. They used geometric reasoning (size criterion) as well as application-specific rules (boundary conditions and proximity with loading zone) for identification of the suppressible features.

Russ [22] mentioned that the determination of the non-critical (suppressible) features relies on different attributes of not only the features themselves, but also of the entire part model and analysis. Some of these attributes include the feature type, feature dimensions, proximity of features to the boundary conditions, analysis type, and part dimensions. He used full feature parameters for deciding the eligibility of the suppression.

The selection criteria for suppressing the features, apart from being application-specific, are based on the targeted accuracy, cost or preparation time, etc. [7]. Looking at the variety of inputs, types of analysis and application domains, it is difficult to quantify and generalize. So, each domain typically presents its own feature taxonomy with regards to defeaturing to decide the eligibility for suppression.

Danglade [7] used machine-learning techniques to capitalize the knowledge and experience; where the analysts have to provide the decision making for the defeaturing step. After a large number of learning trials, the system itself suggests relative necessity of the features and their impact on the accuracy of the results.

Joshi and Dutta [12] recognized the sheet metal features on free-form surfaces and then suppressed them for simplification. Their work appears to be limited to only holes, fillets and bosses.

Kang et al. [13] customized the defeaturing criteria for shipyard requirements where, apart from geometric reasoning criteria such as volume, they included application-specific rules for ports and outer boundary.

Commercial packages like ACIS®, Autodesk Fusion®, and Altair's Hypermesh® also provide similar defeaturing capabilities, mainly for CAE analysis.

Some important conclusions based on the literature survey of feature-based defeaturing are:

- Identification of the suppressible features, just by their types, cannot be used blindly across all domains. For example, a rib-like feature may not be relevant in the metal flow analysis but may be relevant in the heat transfer analysis. So, there is a need for domain-specific identification rules, such as rules for sheet metal domain, shipbuilding, etc.
- Geometric reasoning can be used for size-based identification of the suppressible features. Based on the engineering judgment, a size threshold may be decided, below which features can be suppressed. Here, the calculation of the feature size becomes critical as it decides the eligibility for suppression. Full feature dimensions are used by some of the past approaches [13][22] giving wrong results, as the feature is not fully present in the final shape.

The work presented in this paper focuses on feature-based defeaturing specific to sheet metal parts. Sheet metal parts are prevalent in industries, such as automotive, aerospace, process, electronics, etc. Clear definition and classification of sheet metal features is a prerequisite in deciding the defeaturing rules. Some of the available classifications for sheet metal features are reviewed in the following section.

2.2. Sheet metal features

Liu et al. [20] stated that a sheet metal CAD part model is made up of base features (Wall/Face feature) upon which several child/secondary features (Cutouts, etc.) are positioned. Then, the tertiary and connector features are added to complete the desired shape. They classified sheet metal features into *Primitives* features which are independent features, *Add-ons* which are on the *Primitives*, *Connectors* which connect the features and *Composites* which are made up of the earlier-defined types.

Sunil [24] classified sheet metal features into *face*-based features which lie on the face (holes, dimple, and beads), *edge*-based features (flange, ridge) which lie on the periphery of the part, while *transitive* features (bend) lie between the faces.

Survey suggests that for a well-connected midsurface, primitive and connecting/transitive features are critical and they need to be retained, whereas secondary features can be suppressed by comparing their size relative to the given threshold.

This paper proposes a novel method to address the stated issues. Section 3 presents details of the method using two phases and algorithms (Subsection 3.1 and 3.2) whereas in Section 4 we present the results with case studies.

3. Computing gross-shape by feature-based defeaturing

In defeaturing of a feature-based CAD model, the relevance of each feature is measured by an evaluation metrics [4]. The evaluation metrics used in this study is divided into two classes, viz. application context-specific criteria (Phase I) and geometric reasoning-based criteria (Phase II). This work focuses on the sheet metal domain as an example of application context-specific criteria for defeaturing, for the end-use of finding the gross shape needed for computing the midsurface. Fig. 1 shows the overall process where Phase I is directly dependent on the feature information available in the sheet metal feature-based CAD models, and Phase II works on the final solid shape, while leveraging feature information for deciding the suppressibility.

- **Phase I - Defeaturing based on the application context**: The model feature tree is traversed and the candidate features for suppression are identified based on certain criteria. In this study, rules based on the sheet metal features taxonomy are used to decide the suppressibility of the features. For other domains or applications, this phase can be customized by employing rules suitable to those applications.
- **Phase II - Defeaturing based on geometric reasoning**: This phase starts with the final Brep for identifying the remnant portions of the features, and those whose sizes are below the threshold are identified for suppression. This phase, being geometric in nature, can be generic to different applications, with an option of customize the threshold based on engineering judgment appropriately.

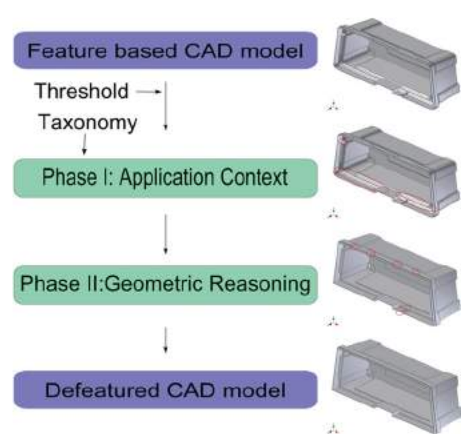


Figure 1. Overall Defeaturing Process.

The combined method (Phase I & II) is called as "**Smarf**" (**S**heet **M**etal and **R**emnant **F**eature). Following sections present the algorithms for both phases in details.

3.1. Phase I: defeaturing based on the application context

The case study shown in Table 1 shows effectiveness of the gross shapes in the generation of a better midsurface. The concept of "gross shape" is subjective and hard to quantify [19]. Thus formulating the rules for identification of suppressible features is the most critical step that affects the output of defeaturing. Apart from similar classifications in the literature, a thorough analysis of inputs from various surveys with engineers and experts on the field was done with respect to the midsurface quality metrics, such as preservation of medial-ness, problems, errors, etc. Proposed taxonomy (Fig. 2) is the result of this analysis. In comparison with the other sheet metal features classifications, such as for features recognition [8,9] and process planning [14], the proposed taxonomy is a novel one since the application context itself is different than the previous works.

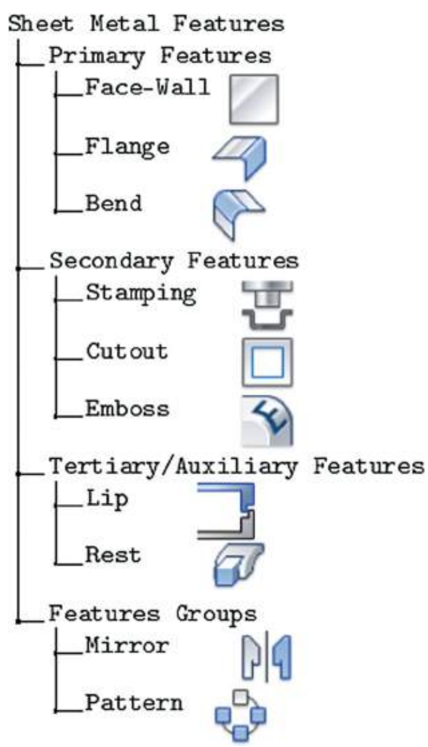


Figure 2. Taxonomy for Gross Shape (Icons source: [3]).

3.1.1. Sheet metal features taxonomy

Taxonomy is represented by "vocabulary" and "structure". In the context of feature based CAD applications it is a scheme to represent and classify features for a particular purpose. In this paper sheet metal features are classified (Fig. 2) for the purpose of defeaturing as follows:

- **Primary Features**: Constituents of the main shape of the body. Features that can exist independently and are created in the initial operations. These are not suppressed, irrespective of their sizes as they form the principal/gross shape and removing these would create the missing midsurface patches. Some examples are:
  - Face-Wall
  - Flange
  - Bend
- **Secondary Features**: Features are placed on the primary features and created after them. These are suppressed, based on their relative size with respect to the size of the whole part. Smaller features unnecessarily create problems in the geometric computation of the midsurface, so they need to be suppressed. Some examples are:
  - Stamping
  - Cutout
  - Emboss
- **Tertiary/Auxiliary Features**: Decorative or helpers and are not part of the main shape but modify the local geometry (point/edge). So they can be suppressed irrespective of their sizes. Examples are:
  - Lip
  - Rest
  - Letterings
- **Feature Groups**: These are an array of features and are modeled together as a single group. Suppression criteria applied is evaluated on the collective group and not on an individual feature. Some examples are:
  - Mirror
  - Patterns

Examples of these features are presented in a sheet metal part model as shown in Fig. 3.

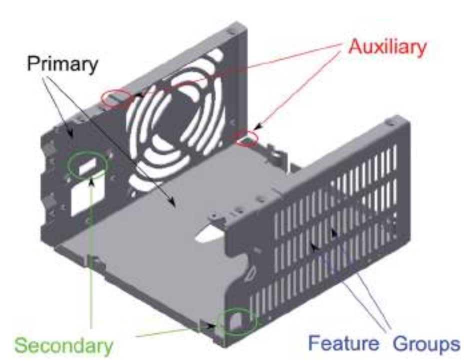


Figure 3. Examples of the classified types.

3.1.2. Defeaturing sheet metal model

The following steps identify the sheet metal features as per the classification presented in section 3.1.1. The identification uses the feature tree available as part of the feature based CAD model.

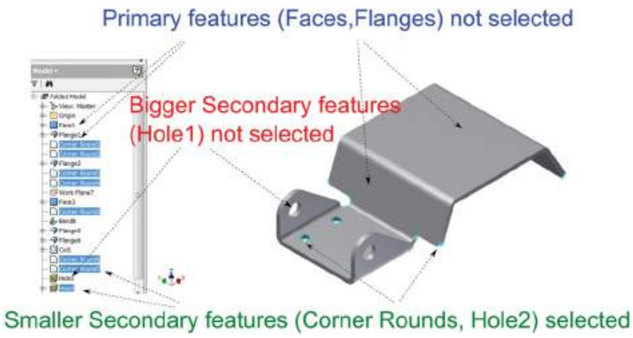


Figure 4. Selection of features based on Taxonomy.

Algorithm to identify candidate features for de-featuring based on Sheet Metal feature taxonomy:

- A List (***sl***) initialized to which the suppressible features are added.
- The model feature tree is traversed and the candidate features for suppression are identified based on a set of heuristic criteria such as "Primary features are not to be suppressed", "Secondary features, if small, are selected" etc. (Fig. 4).
- The identified features are added to ***sl***.
- The ***sl*** is presented to the user for verification and changes, if necessary.
- Features in ***sl*** are suppressed
- The model is regenerated and Defeaturing Effectiveness is computed using Eqn. 1.

3.1.3. Phase I: Algorithm 1

Application context-specific – Sheet metal features

Require: A CAD model with access to the feature tree
While nextFace() ! = null **do**
  F_i = currentFace()
  $Area_{face}$ + = $F_i \rightarrow$ area()
end while
p = getInputFromUser()
$D = \frac{p}{100} \times Area_{face}$ // threshold size
while nextFeature() ! = null **do**
  f_i = currentFeature()
  if $f_i \rightarrow$ isAuxilaryFeature() **then**
    sl → add(f_i)
  else if $f_i \rightarrow$ isGroupFeature() **then**
    $Area_{feature}$ = $f_i \rightarrow$ combinedArea() // Total area of the constituent features
    if $Area_{feature} < D$ **then**
      sl → add(f_i)
    end if
  else
    $Area_{feature}$ + = $f_i \rightarrow$ area()
    if $Area_{feature} < D$ **then**
      sl → add(f_i)
    end if
  end if
end while
sl → suppress()
validate()

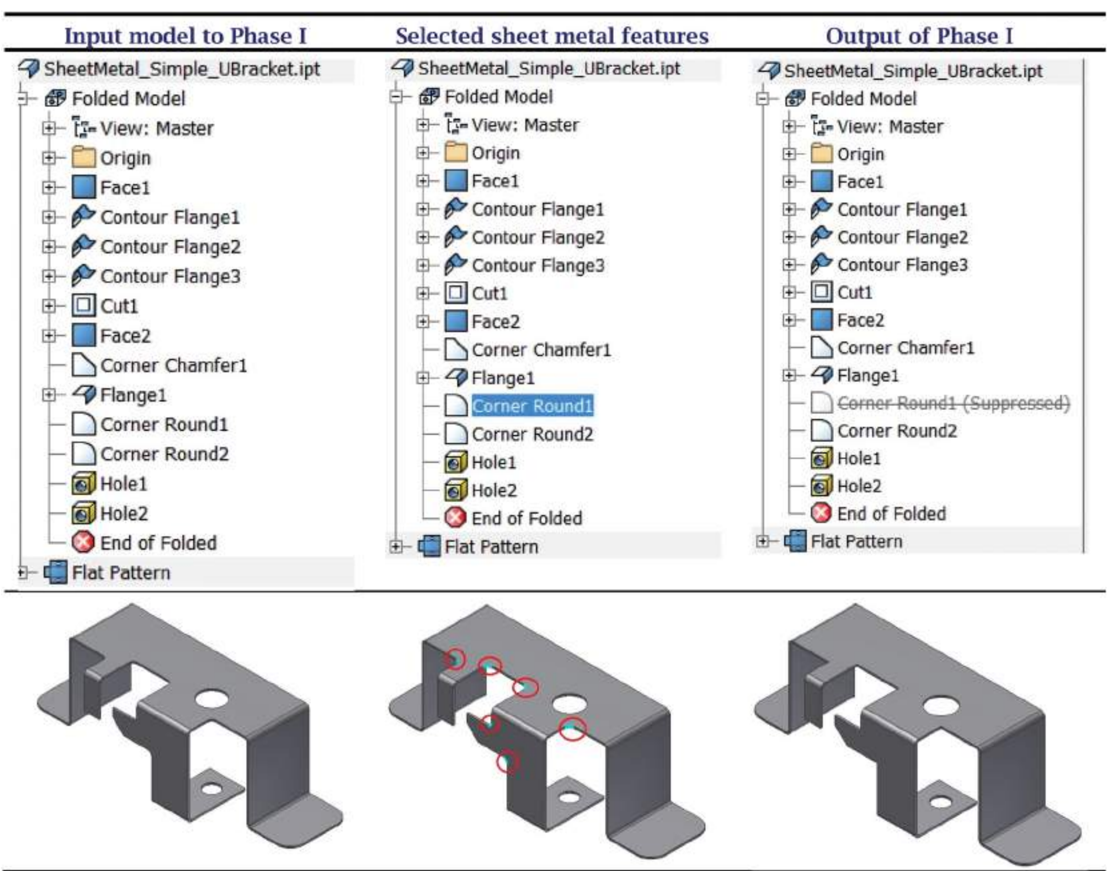


Figure 5. Phase I.

3.1.4. Results of Phase I

The Phase I process is shown pictorially below. The first picture of Fig. 5 shows input to this phase, the model to be simplified. The second picture shows the features selected using the sheet metal feature taxonomy and the last picture shows the defeatured model as the output of Phase I.

3.2. Phase II: defeaturing based on geometric reasoning

In the feature-based design paradigm, the CAD model is built step-by-step using features at each step. Feature parameters are used to compute the 'canonical' (tool-body) volume first, which is then Booleaned to the model built till then. During this operation, some portion of the canonical volume may get consumed, leaving behind the remaining (remnant) volume in the final solid (Fig. 6).

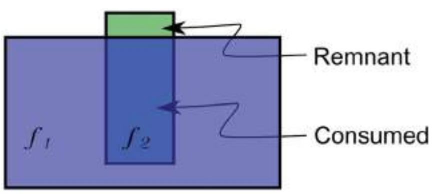


Figure 6. Remnant and Consumed portions of feature volume of f_2

Identification of suppressible features based on the feature volume computed from the full feature parameters yield incorrect results as the final shape may not retain the full feature volume. So, this work has devised a novel method (Algorithm 2) to find the size of remnant feature volume to be used for deciding the suppressibility (Fig. 7) of the features.

This phase starts with the final Brep for identifying the remnant portions of the features. This computation, being geometric in nature, can be generic to many applications, with an option to customize the threshold based on engineering judgment specific to the given application.

3.2.1. Preliminaries

- At j^{th} feature (f_j) the model built till then is referred as M_{j-1}
- $V_j = volume(f_j)$, V_j is the canonical/tool-body volume of f_j
- $M_j = M_{j-1} \oplus V_j$ The model moves to the next state M_j by regularized boolean ($\oplus$) of existing state M_{j-1} with the canonical volume of f_j i.e. V_j
- $V_j = R_j \cup^{C_j}$. Some portion of V_j gets consumed (C_j) is called as the "Consumed Feature Volume", whereas the portion that remains (R_j) is called as the "Remnant Feature Volume" (Fig. 6).
- $M_j = M_{j-1} \cup^{R_j}$ and thus $R_j = M_j - M_{j-1}$ is the remnant feature volume [15].

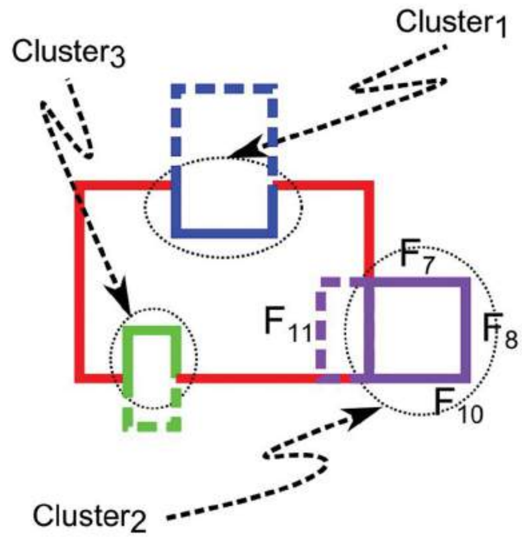


Figure 7. Formation of Clusters.

Algorithm to identify candidate features for de-featuring based on Remnant Feature method:

- Faces of the final body are iterated.
- For each remnant face, its owning feature is extracted via attributes stored on them.
- Clusters/Groups of faces are built based on the owning features as shown in Fig. 7. The dotted portion in a cluster represents the Consumed Feature, whereas the encircled portion is the Remnant feature.
- Size of the cluster can be calculated by various methods like Influence Volume (obtained as a difference of the volume, if the feature is suppressed and then unsuppressed) or the union of bounding-boxes, etc. This work uses summation of the area of the remnant faces (Tab. 2) as the Size criterion.
- Each cluster-owning feature(s) is added to ***sl*** based on the threshold value given by the user.
- The ***sl*** is presented to the user for verification and changes, if necessary.
- Features in ***sl*** are suppressed.
- The model is regenerated and Defeaturing Effectiveness is computed (section 3.3).

Table 2. Evaluating Cluster sizes.

Clusters	Size	Feature
$Cluster_1$	0.25	$Extrude_2$
$Cluster_2$	0.25	$Extrude_3$
$Cluster_3$	0.125	$Hole_1$

3.2.2. Phase 2: Algorithm 2

Remnant Faces method

```
Require: A CAD model with access to the feature tree
While nextFace() != null do
  F_i = currentFace()
  feat = F_i → owingFeature()
  addedFlag = false
  while nextCluster() != null do
    cl_j = currentCluster()
    if cl_j → owingFeature() == feat then
      cl_j → add(F_i)
      addedFlag = true
    end if
  end while
  if addedFlag == false then
    cl_n = newCluster()
    cl_n → owingFeature() = feat
    cl_n → add(f_i)
  end if
end while
while nextCluster() != null do
  cl_k = currentCluster()
  size = cl_k → calculateSize()
  if size < D then                         // Threshold 'D' defined in Alg 1.
    sl → add(cl_k)
  end if
end while
sl → suppress()
validate()
```

3.2.3. Results of Phase II

The Phase II process is shown pictorially in Fig. 8.

- The first picture shows input to this phase (it is the output of the previous Phase I)
- The second picture shows the features selected by the remnant feature method, and
- The last picture shows the defeatured model as output of the Phase II.

Output of Phase II (Fig. 8) is the gross shape of the input given (Fig. 5). With 50% reduction in the number of features and 17% reduction in the number of faces, the gross shape resulted has retained all the important features necessary for the computation of a well-connected midsurface.

3.3. Effectiveness of defeaturing

Effectiveness of the defeaturing process can be computed using a wide variety of methods. They can be classified into input-based and output-based methods. In the input-based method, based on the engineering judgment, the initial defeaturing parameters (such as, size threshold,

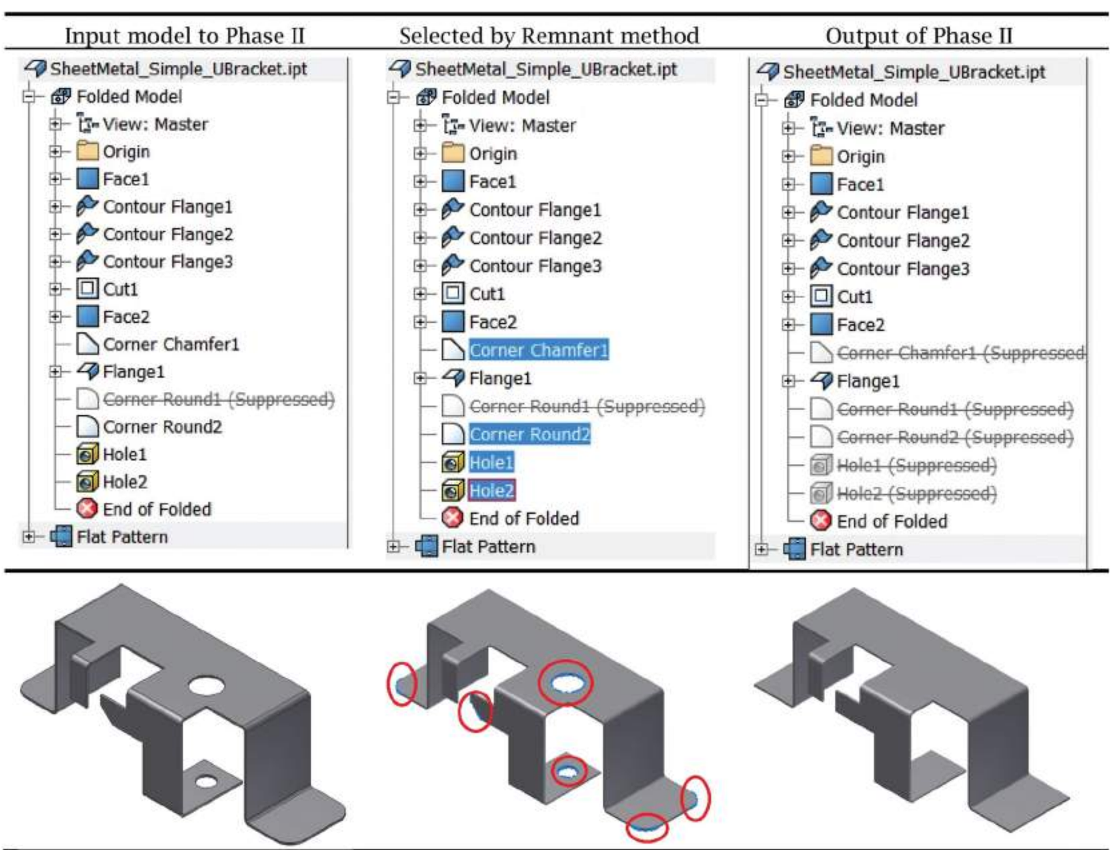


Figure 8. Phase II.

feature taxonomy, etc.) are set, and the output resulted is considered as the valid one. In the output-based method, some initial defeaturing parameters are set and the output is assessed against desired benchmarks, such as, reduction in volume/faces/features, etc. The process is repeated till the desired output is achieved.

In this work, the effectiveness of defeaturing is computed by measuring **Percentage reduction in the number of the faces**. More the percentage, the more effective is the defeaturing process. Features can also be used in place of faces to form another criterion for measuring the effectiveness:

- Total number of faces in the original part (nF)
- Number of sheet metal features suppressed in Phase I (nS)
- Number of faces left after Phase I (mF)
- Number of features suppressed in Phase II (nR)
- Number of faces left after Phase II (rF)
- Defeaturing effectiveness (pR) while keeping the overall shape intact (%)

$$pR = \left(1 - \frac{rF}{nF}\right) X\ 100 \qquad (1)$$

Apart from pR, there could be some other and more involved criteria that can be used as follows:

- **Medial Axis Comparisons**: Small features create branches in their Medial Axis Transform (MAT [21]) representation. So, a complex model will have branches and its corresponding defeatured model won't have them. Comparison of both can give idea about the effectiveness.
- **Mesh**: Comparing faceted mesh generated by body defeatured by **Smarf** with the mesh simplified by any benchmark mesh simplification methods can give the effectiveness measure.
- **Size**: Comparison of volume of the original and the defeatured model.
- **Shape deviation**: Using Part-Compare functionality, maximum deviation between the original and the defeatured, can be calculated. This deviation should be within predefined limits.

3.4. Implementation

Prototype implementation has been done using Application Programming Interface (APIs) of Autodesk Inventor®, in Microsoft Visual Basic .Net 2010 environment on 2.3 GHz Intel i3, 64 bit processor PC with 4 GB RAM. Many of the example parts have been borrowed from GrabCAD® site (http://www.grabcad.com).

The implemented (Fig. 9) user work-flow is as follows:

- Input part is loaded.
- **Init-Unfold**: Unfold features are suppressed. Part size and thresholds are calculated.
- **Preview S**: Phase I-selected (Sheet metal) features are highlighted
- **Suppress S**: The selected (Sheet metal) features are suppressed.
- **Preview R**: Phase II-selected (Remnant) features are highlighted
- **Suppress R**: The selected (Remnant) features are suppressed.

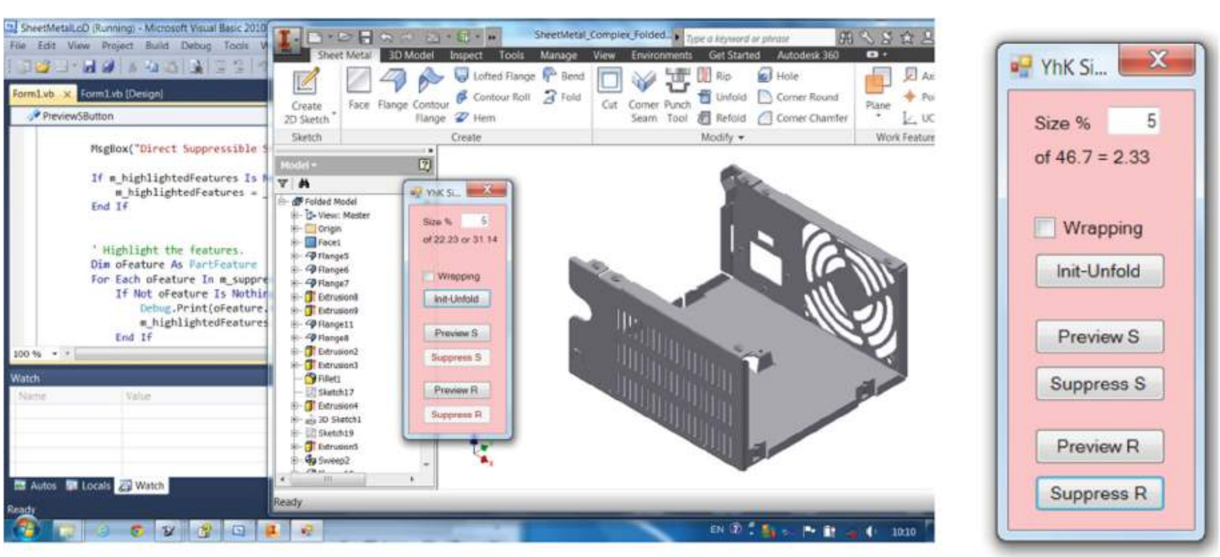
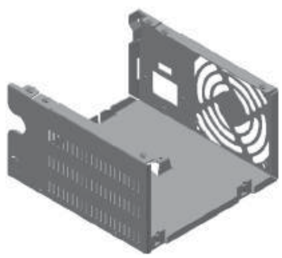
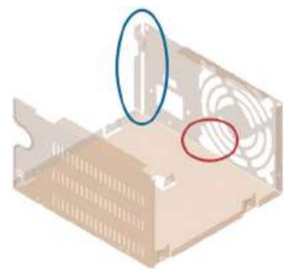
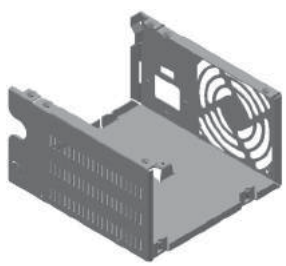
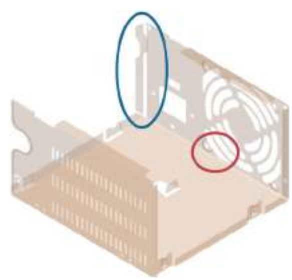
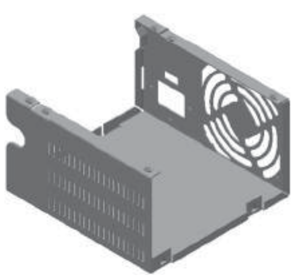
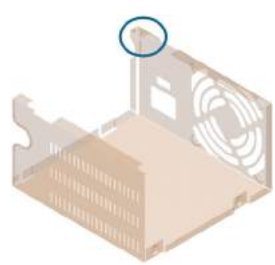


Figure 9. Screen-shot of implemented program.

4. Results

Following test case shows effect of defeaturing on the quality of the midsurface. Size threshold used here is certain percentage of the summation of face-areas of all the faces in the original body.

1. Threshold (D) **3**% of the total part size

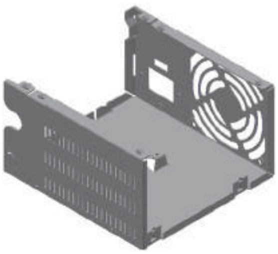
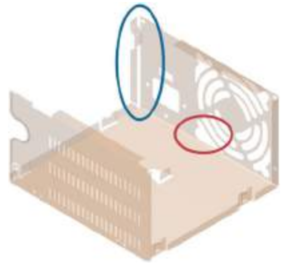
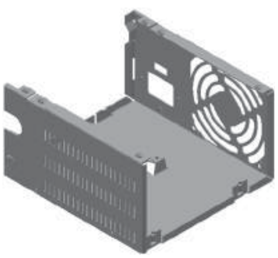
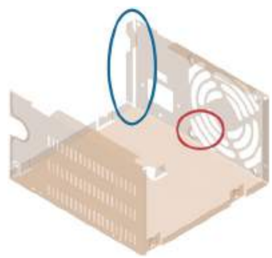
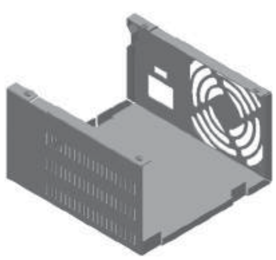
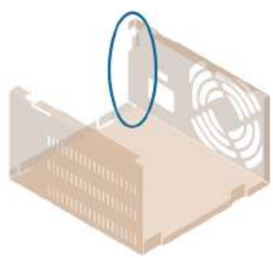
	Model	Midsurface	Explanation
Original/Input			Gaps in the midsurface. Two of the gaps are marked (blue and red).
Output of Phase I and input to Phase II			Although the number of missing gaps in the midsurface has reduced (red gap is filled), but the gaps between the surface patches (blue gap) is still seen. These gaps are marked.
Output of Phase II			Some of the gaps (blue gap top portion) are filled and the output is a bit better-connected midsurface. It retains many of the necessary features adequately 'representing' the gross shape.

Effectiveness of **Smarf** with **3%** threshold, based on the criterion defined by Eqn. 1 is:

Entities	Original	Phase-I	Phase-II
Faces	833	774	697
Suppressed features		7	32

$$pR = \left(1 - \frac{697}{833}\right) X\ 100$$
$$= 16$$

2. Threshold (D) **5**% of the total part size

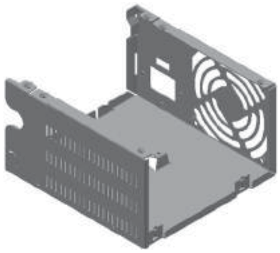
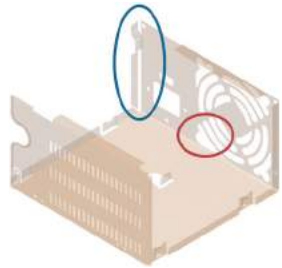
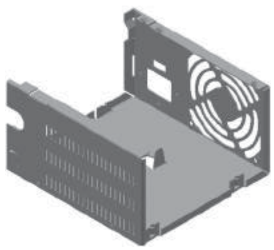
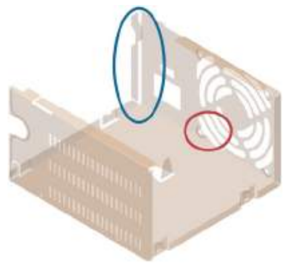
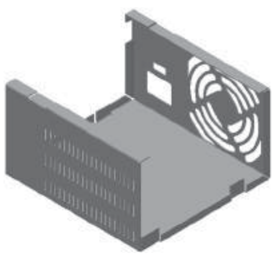
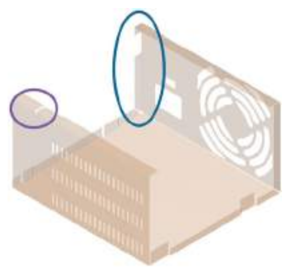
	Model	Midsurface	Explanation
Original/Input			Gaps in the midsurface. Two of the gaps are marked (blue and red).
Output of Phase I and input to Phase II			Although the number of missing gaps in the midsurface has reduced (red gap is filled), but the gaps between the surface patches (blue gap) is still seen. These gaps are marked.
Output of Phase II			Most of the gaps (blue gap full) are filled and the output is a better-connected midsurface. It retains all the necessary features adequately 'representing' the gross shape.

Effectiveness of **Smarf** with **5%** threshold, based on the criterion defined by Eqn. 1 is:

Entities	Original	Phase-I	Phase-II
Faces	833	772	617
Suppressed features		7	40

$$pR = \left(1 - \frac{697}{833}\right) X\ 100$$
$$= 26$$

3 Threshold (D) **10**% of the total part size

	Model	Midsurface	Explanation
Original/Input			Gaps in the midsurface. Two of the gaps are marked (blue and red).
Output of Phase I and input to Phase II			Although the number of missing gaps in the midsurface has reduced (red gap is filled), but the gaps between the surface patches (blue gap) is still seen. These gaps are marked.
Output of Phase II			Most of the gaps (blue gap full) are filled. Removal of purple feature could be the domain decision. It retains all the necessary features adequately 'representing' the gross shape.

Effectiveness of **Smarf** with **10%** threshold, based on the criterion defined by Eqn. 1 is:

Entities	Original	Phase-I	Phase-II
Faces	833	715	522
Suppressed features		17	48

$$pR = \left(1 - \frac{522}{833}\right) X\ 100$$
$$= 37$$

Table 3. Comparison of defeaturing method.

	Size [22]	Wrap [15]	Reorder [16]	Smarf
Input	Features	Brep	Features	Features and Brep
Method	Find Feature size, compare with threshold, suppress.	Wrap-around. Negative volumes are removed.	Features converted to volumetric + ve, −ve and reordered.	Sheet metal features and Remnant face logic.
Disadvantages	Limited to FEM as BC and Load path features are not suppressed.	Concave edge filling creates odd shapes. Principal shape is lost.	Due to merging in Cellular model, update capability is lost forever.	Context dependent.

5. Conclusion

Most of the defeaturing algorithms are based on the mesh/solid (Brep) as input. With the availability of feature information in the feature based CAD models, it has become possible to leverage it for defeaturing purpose effectively. This work proposes two novel algorithms for defeaturing sheet metal CAD models that can be conveniently used for downstream application of generating a well-connected midsurface. With the first algorithm, each candidate sheet metal feature is suppressed based its sheet metal characteristics. The second algorithm leverages the size of the remnant volumes for deciding the suppressibility. Advantages of feature based parametric modeling can be combined with the proposed approach for automatic defeaturing and simplification of the model. Comparison with other defeaturing methods is presented in Tab. 3.

Uniqueness of the **Smarf** approach in comparison with few other relevant approaches [14,16,22]:

- Suppressibility rules specific to the domain such as sheet metal feature-based design.
- Suppressibility rules based on the remnant and not full feature/part volume.
- No blind suppression of all the negative features or filling-up of the concave volumes.

Practical example shown in section 4 demonstrates the efficacy of the proposed approach. It is evident that, even after substantial defeaturing the gross shape computed retains the features essential for computation of a well-connected midsurface.

ORCID

Yogesh H. Kulkarni http://orcid.org/[0000-0001-5431-5727]
Ravi K. Gupta http://orcid.org/[0000-0003-4419-8111]
Anil Sahasrabudhe http://orcid.org/[0000-0003-0486-0596]
Mukund Kale http://orcid.org/[0000-0003-1518-9443]
Alain Bernard http://orcid.org/[0000-0002-7037-2980]

References

[1] Ames, A.L.; Rivera, J.J.; Webb, A.J.; Hensinger, D.M.: Solid model design simplification. Technical Report #SAND97-3141, Sandia National Laboratories, http://dx.doi.org/10.2172/567504

[2] Andújar, C.: *Geometry Simplification*. Research report Nr LSI-99-2-R, Universitat Politècnica de Catalunya, Barcelona.

[3] Autodesk: Autodesk Inventor 2014 Help http://help.autodesk.com/view/INVNTOR/2014/ENU/

[4] Autodesk: Techniques for Electronic Analyses. http://knowledge.autodesk.com/support/simulation-cfd/learn-explore/

[5] Belaziz, M.; Bouras, A.; Brun, J.-M.: Morphological analysis for product design., *Computer-Aided Design*, 32(5–6), 2000, 377–388. http://dx.doi.org/10.1016/S0010-4485(00)00019-1

[6] Dabke, P.; Prabhakar, V.; Sheppard, S.: Using Features to Support Finite Element Idealizations, Proceedings of the ASME Computers in Engineering, Minneapolis, MN, 1994, 183–193. http://dx.doi.org/10.1.1.31.9589

[7] Danglade, F.; Pernot, J.-P.; Véron, P.: On the use of Machine Learning to Defeature CAD Models for Simulation, *Computer-Aided Design and Applications*, 11(3), 2013, 358–368. http://dx.doi.org/10.1080/16864360.2013.863510

[8] Gupta, R.K.; Gurumoorthy, B.: Classification, representation, and automatic extraction of deformation features in sheet metal parts, Computer-Aided Design, 2013, 1469–1484. http://dx.doi.org/10.1016/j.cad.2013.06.010

[9] Gupta, R.K.; Gurumoorthy, B.: Unified Taxonomy for Reference Ontology of Shape Features in Product Model., PLM, 2013, 295–307. http://dx.doi.org/10.1007/978-3-642-41501-2_30

[10] Halpern, M.: Industrial Requirements and Practices in Finite Element Meshing: A Survey of Trends. CAE: Ready for the next leap forward, a survey of trends, 1997, 399–411

[11] Hamdi, M.; Aifaoui, N.; Louhichi, B.; BenAmara, A.: Idealization of CAD model for a simulation by a finite element method, *European Journal of Control*, 19(4), 2010, 419–439. http://dx.doi.org/10.3166/ejcm.19.419-439

[12] Joshi, N.; Dutta, D.: Feature Simplification Techniques for Freeform Surface Models, *Journal of Computing and Information Science in Engineering*, 3(3), 2003, 177–186. http://dx.doi.org/10.1115/1.1603307

[13] Kang, Y.; Kim, B.; Mun, D.; Han, S.: Method to simplify ship outfitting and offshore plant equipment three-dimensional (3-D) computer-aided design (CAD) data for construction of an equipment catalog, *Journal of Marine Science and Technology*, 19(2), 2014, 185–196. http://dx.doi.org/10.1007/s00773-013-0239-9

[14] Kannan, T.R.; Shunmugam, M.S.: Processing of 3D sheet metal components in STEP AP203 format. Part I: feature recognition system, *International Journal of Production Research*, 47(4), 2009, 941–964. http://dx.doi.org/10.1080/00207540701510055

[15] Kim, S.; Lee, K.; Hong, T.; Kim, M.; Jung, M.; Song, Y.: An integrated approach to realize multi-resolution of B-rep model, Symposium on Solid Modeling and Applications, 2005, 153–162. http://dx.doi.org/10.1145/1060244.1060262

[16] Lee, S.H.: A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, *Computer-aided Design*, 37(9), 2005, 941–955. http://dx.doi.org/10.1016/j.cad.2004.09.021

[17] Lee, S.H.: Feature-based multiresolution modeling of solids, *ACM Transactions on Graphics*, 24(4), 2005, 1417–1441. http://dx.doi.org/10.1145/1095878.1095887

[18] Lee, S.H.: Feature-based non-manifold modeling system to integrate design and analysis of injection molding products, *Journal of Mechanical Science and Technology*, 23(5), 2009, 1331–1341. http://dx.doi.org/10.1007/s12206-009-0407-3

[19] Lee, Y.; Lee, K.: Geometric detail suppression by the Fourier transform, *Computer-aided Design*, 30(9), 1998, 677–693. http://dx.doi.org/10.1016/S0010-4485(98)00022-0

[20] Liu, Z.; Li, J.; Wang, Y.; Li, C.; Xiao, X.: Automatically extracting sheet-metal features from solid model, *Journal of Zhejiang University Science*, 5(11), 2004, 1456–1465. http://dx.doi.org/10.1631/jzus.2004.1456

[21] Ramanathan, M.; Gurumoorthy, B.: Generating the Mid-Surface of a Solid using 2D MAT of its Faces, *Computer Aided Design and Applications*, 1(1–4), 2004, 665–674. http://dx.doi.org/10.1080/16864360.2004.10738312

[22] Russ, B.: Development of a CAD Model Simplification Framework for Finite Element Analysis, Master's Thesis, University of Maryland (College Park, Md.)., 2012.

[23] Smit, S.: Integration of Design and Analysis Models, *Computer-Aided Design and Applications*, 6(6), 2009. http://dx.doi.org/10.3722/cadaps.2009.795-808

[24] Sunil, V.B.; Pande, S.S.: Automatic recognition of features from freeform surface CAD models, *Computer-Aided Design*, 40(4), 2008, 502–517. http://dx.doi.org/10.1016/j.cad.2008.01.006

[25] Thakur, A.; Banerjee, A.G.; Gupta, S.K.: A survey of CAD model simplification techniques for physics-based simulation applications, *Computer-Aided Design*, 41(2), 2009, 65–80. http://dx.doi.org/10.1016/j.cad.2008.11.009

[26] Woo, Y.: Automatic simplification of solid models for engineering analysis independent of modeling sequences, *Journal of Mechanical Science and Technology*, 23(7), 2009, 1939–1948. http://dx.doi.org/10.1007/s12206-009-0509-y