Proceedings of DETC'04
2004 ASME Design Engineering Technical Conferences
September 28 - October 2, 2004, Salt Lake City, Utah USA

DETC2004-57702

LOCAL FEATURE EXTRACTION USING SCALE-SPACE DECOMPOSITION

Dmitriy Bespalov[†] **Ali Shokoufandeh**[†] **William C. Regli**[†‡] **Wei Sun**[‡]

Department of Computer Science[†]
Department of Mechanical Engineering & Mechanics[‡]
College of Engineering
Drexel University
Philadelphia, PA 19104

ABSTRACT

In our recent work we have introduced a framework for extracting features from solid of mechanical artifacts in polyhedral representation based on scale-space feature decomposition [1]. Our approach used recent developments in efficient hierarchical decomposition of metric data using its spectral properties. In that work, through spectral decomposition, we were able to reduce the problem of matching to that of computing a mapping and distance measure between vertex-labeled rooted trees.

This work discusses how Scale-Space decomposition framework could be extended to extract features from CAD models in polyhedral representation in terms of surface triangulation. First, we give an overview of the Scale-Space decomposition approach that is used to extract these features. Second, we discuss the performance of the technique used to extract features from CAD data in polyhedral representation. Third, we show the feature extraction process on noisy data – CAD models that were constructed using a 3D scanner. Finally, we conclude with discussion of future work.

1 Introduction

The problem of 3D object recognition is often formulated as that of matching configurations of features. Such configurations are often represented as vertex-labeled graphs, whose nodes represent 3D features (or their abstractions) and whose edges represent spatial relations (or constraints) between the features. The relations are typically geometric or hierarchical, but can include other types of information. To match two 3D models means to establish correspondences between their constituent features. In this context, *features* are intrinsic properties of the 3D shape which may encompass local geometry and topology related to design or manufacturing operations.

The problem of feature extraction is very important; hence, it could be used for similarity assessment [2, 3], provided that an exact representation for the models is provided (i.e. Brep). Unfortunately, these approaches can not be used if only approximate representations (i.e. polyhedral) are available. In [1] we showed how Scale-Space decomposition could be used to extract features from 3D models in polyhedral representation.

If a feature extraction produces consistent results even if only partial data is available (local feature extraction), then it could be used for partial matching. Our current work shows how Scale-Space decomposition can be extended to address the issue of extracting local features from 3D models in polyhedral representation. In other words, if we are given only a part of an object (for instance, a single 3D laser scan) we want to be able to extract features from this object and then find complete objects in our database that have similar features. In this paper, we discuss such feature extraction process.

2 Related Work

Our research aims to bring information retrieval to CAD databases, enabling them to have indexing and query mechanisms like those beginning to found in multimedia databases and knowledge management systems.

2.1 Feature Extraction on Solid Models

Scale-Space decomposition promises to decompose a 3D model into structurally relevant components *automatically*. This especially important when the underlying 3D data is noisy. An important fact is that such decomposition will allow for efficient comparison of 3D models in terms of similarity of underlying features [1]. Scale-Space decomposition is very popular in Computer Vision for extraction of spatially coherent features. Most of the work in this community has focused on the Scale-Space features of 2D images using wavelets or Gaussian filters [4, 5].

Some work has been done to automate feature extraction process on CAD models. Elinson et al. [6] and Cicirello and Regli [7–9] examined how to develop graph-based data structures to capture feature relationships and create heuristic similarity measures among artifacts. More recent work in [10] examined manufacturing feature-based similarity measurement.

Historically Group Technology (GT) coding was the way of indexing of parts and part families [11]. This facilitated process planning and cell-based manufacturing by imposing a classification scheme (a human-assigned alphanumeric string) on individual machined parts. While there have been a number of efforts to automate the generation of GT codes [12–14], none have been fully transitioned to commercial practice.

2.2 Feature Extraction on Shape Models

Thompson et al. [15, 16] reverse engineered designs by generating surface and machining feature information of range data collected from machined parts. Hilaga et al. [17] use Multiresolutional Reeb graphs to capture the feature-like information of the models.

3 Feature Decomposition

During the last decade, hierarchical segmentation has become recognized as a powerful tool for designing efficient algorithms. The most common form of such hierarchical segmentations is the scale-space decomposition in computer vision. Intuitively, an inherent property of real-world objects is that they only exist as meaningful entities over certain ranges of scale. The fact that objects in the world appear in different ways depending on the scale of observation has important implications if one aims at describing them. Specifically, the need for multi-scale representation arises when designing methods for automatically analyzing and deriving information from real-world measurements.

In the context of solid models, the notion of scale can be simplified in terms of the levels for the 3D features. The notion of a *feature* in this sense draws from the computer vision literature [4] rather than the CAD literature. Namely, given an object $\mathscr{M}$, we are interested in partitioning $\mathscr{M}$, into k features $\mathscr{M}_1,...,\mathscr{M}_k$ with $\mathscr{M}_i \cap \mathscr{M}_j = \emptyset$, for $1 \le i < j \le k$, and $\mathscr{M} = \bigcup_i \mathscr{M}_i$ subject to maximization of some coherence measure, $f(\mathscr{M}_i)$, defined on the 3D elements forming each $\mathscr{M}_i$. At a finer scale, each feature $\mathscr{M}_i$ will be decomposed into $j = 1,...,k$ sub-features, subject to the maximization of some coherence measures.

There are three central components in the aforementioned process: the number of components at each scale of decomposition, k; the feature coherence function $f(.)$; and the number of scales of decomposition process, ℓ. In most pattern recognition applications, k is a control parameter. If models $\mathscr{M}$ and $\mathscr{M}'$ are topologically similar, the k major components at every scale should also be similar. The coherence function $f(\mathscr{M}^i)$ will assign an overall metric to the quality of 3D elements participating in the construction of feature $\mathscr{M}^i$. Finally, the depth of decomposition will be controlled depending on the quality of a feature in comparison to all its sub-features. Specifically, assume $\mathscr{M}^i$ represents a feature at scale i, and $\mathscr{M}_1^{i+1},...,\mathscr{M}_j^{i+1}$, for $j \le k$ represent its sub-features at scale $i+1$. The decomposition process should proceed to scale $i+1$ with respect to feature $\mathscr{M}^i$ if and only if $f(\mathscr{M}^{i+1}) \le f(\mathscr{M}_1^{i+1}) + f(\mathscr{M}_2^{i+1}) + ... + f(\mathscr{M}_j^{i+1})$. This simple criteria for expansion of scale-space at every feature has its roots in information theory. It is in fact motivated by linear form similar to entropy of feature $\mathscr{M}^i$ as opposed to its sub-features $\mathscr{M}_1^{i+1},...,\mathscr{M}_j^{i+1}$. In the end, a set of the leaf nodes in a decomposition tree would correspond to the final features of a given model.

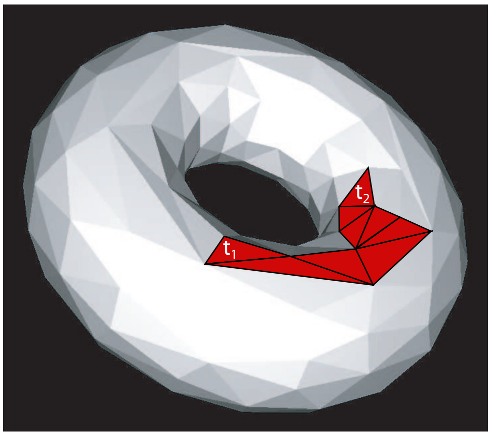


Figure 1. Illustration of the angular shortest path between triangular faces t_1 and t_2.

3.1 Decomposition Algorithm

We are given a 3D model $\mathscr{M}$ in polyhedral representation (in our experiments we used models in VRML format). Before we can proceed with the scale-space decomposition of model $\mathscr{M}$, we must choose a suitable distance function to capture the affinity structure of $\mathscr{M}$. A well-known metric functions is the shortest-path metric $\delta(.,.)$ (geodesic distance [18]) on the triangulation of $\mathscr{M}$ with respect to points $\{v_1,...,v_n\}$; i.e., $\mathscr{D}(u,v) = \delta(u,v)$, the shortest path distance on the triangulated surface between u and v for all $u,v \in \mathscr{M}$. We have used such shortest path distance function in original decomposition.

In this work we used a different distance function which is computed with respect to triangular faces of the model $\mathscr{M}$

$\{t_1,...,t_n\}$. Here and in the rest of the paper n denotes the number of triangles in the model. We define the angular shortest path between two triangular faces t_i and t_j to be the shortest path on the surface of the model which is computed in terms of angular difference between faces. Figure 1 shows an angular shortest path between two faces t_1 and t_2. Specifically, let $t_i \rightsquigarrow t_j$ denote the angular shortest path $(t_i,t_m,t_l,...,t_j)$ between faces t_i and t_j. And let $t_m \rightarrow t_l \in t_i \rightsquigarrow t_j$ denote two adjacent triangular faces t_m and t_l on the angular shortest path $t_i \rightsquigarrow t_j$. Then, the distance function used in this work is defined as

$$\mathscr{D}(t_i,t_j) = \max_{t_m \rightarrow t_l \in t_i \rightsquigarrow t_j} \angle(t_m,t_l). \tag{1}$$

Intuitively, distance $\mathscr{D}(t_i,t_j)$ is the maximum angle between adjacent faces on the angular shortest path between t_i and t_j. The rationale behind such measure is to quantify the smoothness of the surface – small angle between adjacent faces correspond to smooth surface.

Observe that by construction the matrix $\mathscr{D}_{\mathscr{M}} = [\mathscr{D}(t_i,t_j)]_{n\times n}$ is symmetric. Also note that distance measure $\mathscr{D}$ is not a metric function, but it captures the geometric structure of the model $\mathscr{M}$.

Let v_i be the i^{th} row (or column) in $\mathscr{D}$, then v_i is an n-dimensional vector, characterizing the distance structure of face t_i in model $\mathscr{M}$. The problem of decomposing model $\mathscr{M}$ into its k most significant features $\mathscr{M}_1,...,\mathscr{M}_k$ is closely related to k-dimensional subspace clustering (k-DSC). In k-DSC, we are given a set of distance vectors $v_1,...,v_n$, and the objective is to find a k-dimensional subspace $\mathscr{S}$ that minimizes the quantity:

$$\sqrt{\sum_{1\le i\le n} d(v_i,\mathscr{S})^2}, \tag{2}$$

where $d(v_i,\mathscr{S})$ corresponds to the smallest distance between v_i and any member of $\mathscr{S}$. In practice, if $\mathscr{S}$ is given, then $\mathscr{M}_1,...,\mathscr{M}_k$ can be computed using the principle components $\{c_1,...,c_k\}$ of the k-dimensional subspace $\mathscr{S}$ [19]. Observe that, these k vectors will also form a basis for $\mathscr{S}$. Specifically, t_i will belong to the feature $\mathscr{M}_j$ if the angle between v_i and c_j is the smallest among all basis vectors in $\{c_1,...,c_k\}$, i.e., the triangular face t_i that corresponds to the vector v_i will belong to the feature vector $\mathscr{M}_j$ iff the angle between v_i and c_j vectors is the smallest compared to all other basis vectors.

To construct the subspace $\mathscr{S}$, the optimal solution of k-DSC, we will use the technique commonly known as singular value decomposition (SVD) clustering [19]. First, observe that the symmetric matrix $\mathscr{D} \in \mathbb{R}^{n\times n}$ has a SVD-decomposition of the form

$$\mathscr{D} = U\Sigma V^T, \tag{3}$$

where $U,V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and

$$\Sigma = \mathrm{Diag}(\sigma_1,\sigma_2,...,\sigma_n), \tag{4}$$

with $\sigma_1 \ge \sigma_2 \ge ... \ge \sigma_{n'} > 0$, $\sigma_{n'+1} = ... = \sigma_n = 0, n' \le n$. Let us define the order k compression matrix $\mathscr{D}^{(k)}$ of $\mathscr{D}$, for $k \le n'$ as:

$$\mathscr{D}^{(k)} = U\mathrm{Diag}(\sigma_1,...,\sigma_k,0,...,0)V^T. \tag{5}$$

Then,

Theorem 1. [follows from Eckart-Young Theorem [20]].

$$||\mathscr{D}-\mathscr{D}^{(k)}||_2 = \min_{rank(H)=k} ||\mathscr{D}-H||_2. \tag{6}$$

That is, matrix $D^{(k)}$ is the best approximation to $\mathscr{D}$ among all matrices of rank k. In fact, this result can be generalized to many other norms, including *Forbenius* norm:

Corollary 2. *For $A \in \mathbb{R}^{n\times n}$, let*

$$||A||_F = \left(\sum_{i,j} A_{i,j}^2\right)^{1/2}, \tag{7}$$

then,

$$||\mathscr{D}-\mathscr{D}^{(k)}||_F = \min_{rank(H)=k} ||\mathscr{D}-H||_F. \tag{8}$$

Next, assume $\mathscr{S}$ is the range of matrix $\mathscr{D}^{(k)}$ (the subspace spanned by the columns of matrix $\mathscr{D}^{(k)}$), and let c_j, for $1 \le j \le k$, denote the j^{th} column of $\mathscr{D}^{(k)}$. Let $\mathscr{S}' \ne \mathscr{S}$ be any k-dimensional subspace of $\mathbb{R}^n$. For every $t_i \in \mathscr{M}$ let $q_i \in \mathscr{S}'$ be the closest face in $\mathscr{S}'$ to t_i. Define $\mathscr{Q} \in \mathbb{R}^{n\times n}$ with the i^{th} column equal to q_i. Clearly, $rank\mathscr{Q} \le k$. Using Corollary 2 we have:

$$\sum_{i=1}^{n} d(t_i,\mathscr{S}')^2 = \sum_{i=1}^{n} d(t_i,q_i)^2 \tag{9}$$
$$= ||\mathscr{D}-\mathscr{Q}||_F^2 \tag{10}$$
$$\ge ||\mathscr{D}-\mathscr{D}^{(k)}||_F^2 \tag{11}$$
$$= \sum_{i=1}^{n} d(t_i,c_i)^2 \tag{12}$$
$$\ge \sum_{i=1}^{n} d(t_i,\mathscr{S})^2. \tag{13}$$

Consequently;

Proposition 3. . *The set $\mathscr{S} = range(\mathscr{D}^{(k)})$ is the optimal solution to k-DSC problem.*

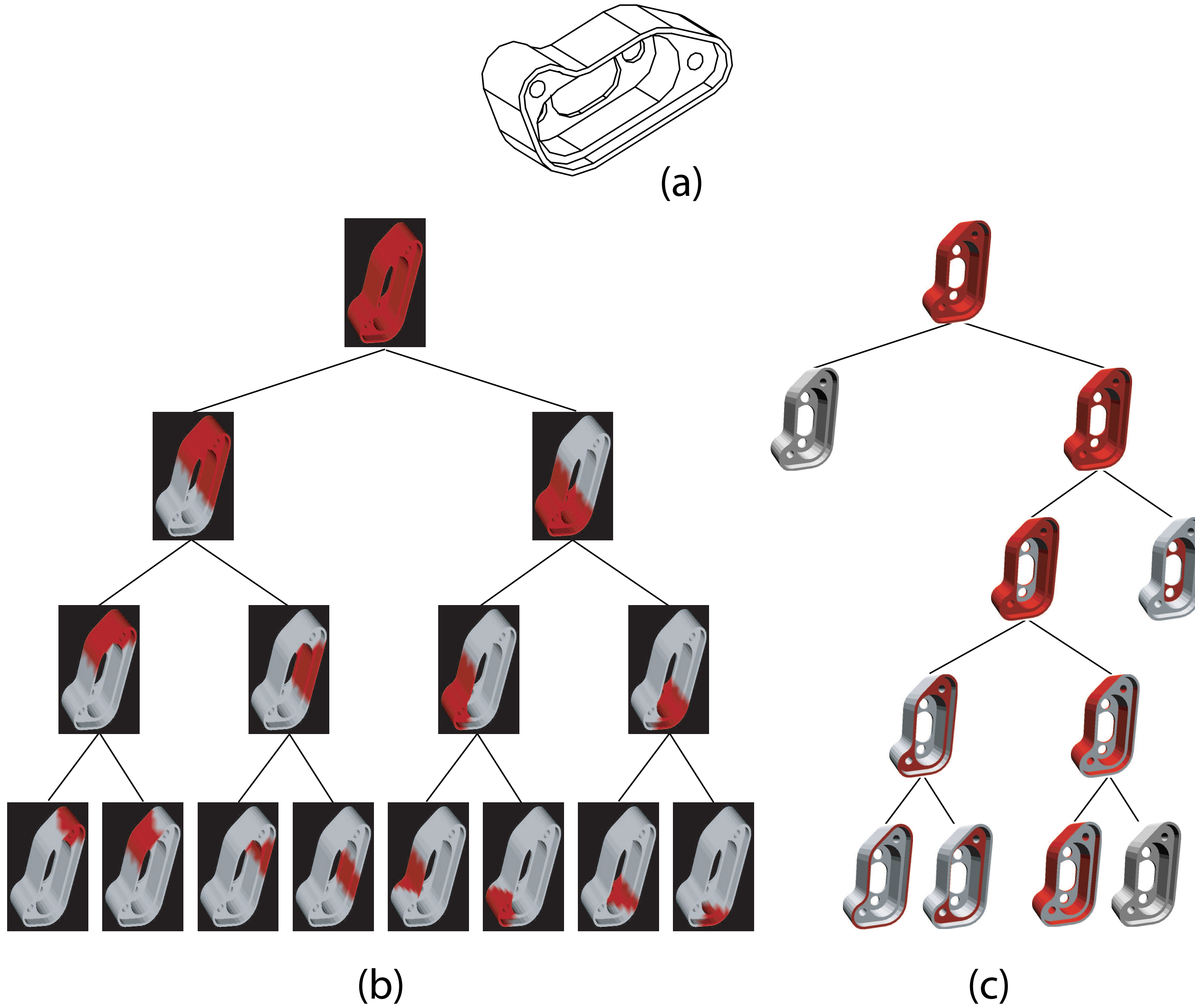


Figure 2. Results of applying FEATURE-DECOMPOSITION($\mathscr{M}, k$) to a model using different distance functions $\mathscr{D}$ for $k = 2$. (a) Sample view of the model. (b) Decomposition using geodesic distance function [1]. (c) Decomposition tree using new angular distance function. Note that this is not a full decomposition tree. The leaf nodes of the tree shown do not correspond to the actual final features extracted from this model.

Algorithm 1 summarizes one phase of scale-space decomposition of $\mathscr{M}$ into its k most significant features, $\mathscr{M}_1, \ldots, \mathscr{M}_k$. Algorithm 1 returns the partitioning of $\mathscr{M}$ by placing each face t_i in $\mathscr{M}$ into one of the partitions $\mathscr{M}_j$, such that the angle between vector t_i and basis vector c_j corresponding to the partition $\mathscr{M}_j$ is minimized. Figure 2 shows two decomposition trees of the model – using geodesic distance and using the distance based on angular measure.

The bottleneck of Algorithm 1 is the $O(n^3)$ SVD decomposition, for an $n \times n$ matrix. The polyhedral representation of a model provides us with a planar graph of a 2D manifold. If we consider only neighboring vertices in the construction of the distance matrix $\mathscr{D}$, the number of non-zero entries in $\mathscr{D}$ would be at most $3n$ (due to planarity of the graph). Computing SVD decomposition for sparse matrices is much faster and takes $O(mn) + O(m\mathrm{M}(n))$ [21]. Where m is the maximum number of matrix-vector computations required and $\mathrm{M}(n)$ is the cost of matrix-vector computations of the form $\mathscr{D}x$. Since M is a planer map and $\mathscr{D}$ is a sparse matrix, $\mathrm{M}(n) = O(n)$ and $m = O(n)$.

3.2 Controlling Decomposition Process

The decomposition process as presented in Section 3.1 does not allow for an explicit mechanism to stop the indefinite break up of a feature. Clearly, we could use a prescribed value to control the decomposition depth of the feature trees, i.e., decomposition process will be stopped when a root branch in feature

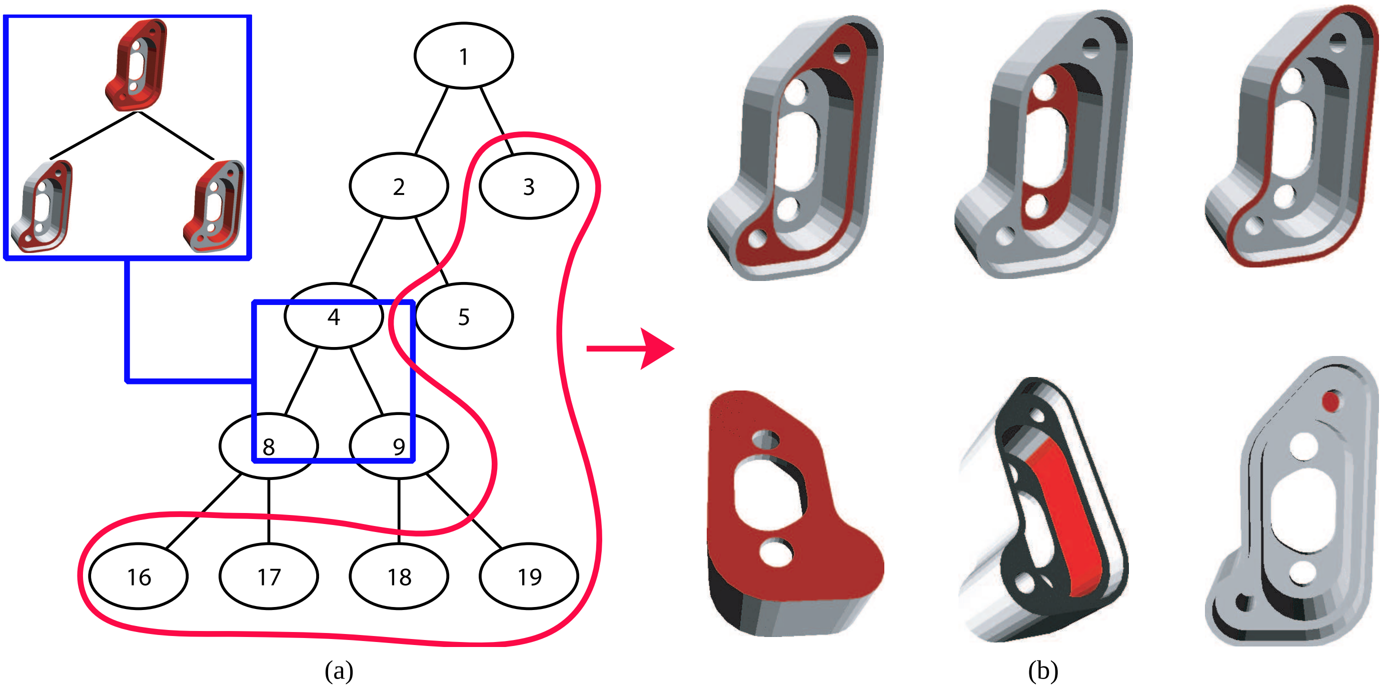


Figure 3. Feature extraction process. (a) Decomposition tree is obtained using FEATURE-DECOMPOSITION($\mathscr{M},k$) algorithm. (b) Leaf nodes of the tree correspond to the features. Note that, for illustration purposes only a subset of extracted features is shown, and features shown in (b) do not correspond to the leaf nodes in (a). The actual decomposition tree is quite large for this model.

Algorithm 1 FEATURE-DECOMPOSITION($\mathscr{M},k$)

1: Construct the distance matrix $\mathscr{D} \in \mathbb{R}^{n\times n}$.
2: Compute the SVD decomposition $\mathscr{D} = U\Sigma V^T$, with $\Sigma = \text{Diag}(\sigma_1, \sigma_2, ..., \sigma_n)$.
3: Compute the order k compression matrix $\mathscr{D}^{(k)} = U\text{Diag}(\sigma_1, ..., \sigma_k, 0, ..., 0)V^T$.
4: Let c_j denote the j^{th} column of $\mathscr{D}^{(k)}$, for $j = 1, ..., k$, and form sub-feature $\mathscr{M}_j$ as the union of faces $t_i \in \mathscr{M}$ with $d(t_i, \mathscr{S}) = d(t_i, c_j)$.
5: Return the set $\{\mathscr{M}_1, ..., \mathscr{M}_k\}$.

decomposition tree reaches a given depth. In this section we will give overview of a mechanism that will control the feature decomposition. Intuitively, the use of this control mechanism will terminate the decomposition process only when all coherent features are extracted.

Let $\mathscr{M}$ be the original model's face set. Assume in the decomposition process a feature $\mathscr{M}_1$ in $\mathscr{M}$ can be decomposed into sub-features $\mathscr{M}_2$ and $\mathscr{M}_3$ (e.g., without loss of generality assume we are bisecting feature $\mathscr{M}_1$). We say that decomposition of the feature $\mathscr{M}_1$ into sub-features $\mathscr{M}_2$ and $\mathscr{M}_3$ is significant if the angular distance between components of $\mathscr{M}_2$ and $\mathscr{M}_3$ is large. Formally, this condition could be expressed as follows:

$$\forall t_i \in \mathscr{M}_2,\ t_j \in \mathscr{M}_3\ \exists t_m \to t_l \in t_i \rightsquigarrow t_j\ s.t.$$

$$t_m \in \mathscr{M}_2\ \wedge\ t_l \in \mathscr{M}_3\ \wedge\ \angle(t_m, t_l) = \mathscr{D}(t_i, t_j),$$

i.e. if the angular shortest path between $t_i \in \mathscr{M}_2$ and $t_j \in \mathscr{M}_3$ contains two faces t_m and t_l (from $\mathscr{M}_2$ and $\mathscr{M}_3$ respectively) with large angular distance, then we should decompose $\mathscr{M}_1$ into $\mathscr{M}_2$ and $\mathscr{M}_3$. Intuitively, if $\mathscr{M}_1$ is smooth we do not want to bisect it any further. On the other hand, if discrepancy between the neighboring triangle in $\mathscr{M}_1$ is significant, we should bisect $\mathscr{M}_1$ it.

4 Experimental Results

We have performed feature extraction on a number of CAD models in polyhedral representation. These models were converted from ACIS format, which is exact representation format. As a result, all of the models have nice structure (i.e. no missing= faces).

In our experiments we would like to examine the qualities features extracted using FEATURE-DECOMPOSITION($\mathscr{M},k$) algorithm. To these ends, we recursively apply FEATURE-DECOMPOSITION($\mathscr{M},k$) to each model for $k = 2$. Once a decomposition tree is obtained, we consider the last layer of the decomposition tree (leaf nodes) as a set of features. Note, that the union of the features (leaf nodes) is equivalent to the surface of the entire model (see Figure 3 for the illustration of feature extraction process).

Figure 4 shows extracted features for several models. We present these images in order to illustrate the type of features the technique can extract. Observe that each feature corresponds to a relatively smooth surface on the model. If there is a significant angular difference on the surface, then it gets decomposed into separate features. Any closed smooth surfaces (i.e. hole) are decomposed into two (i.e. hole) or more (i.e. surface is concave) features. We plan to address the problem of how to use the decomposition trees for matching in the future work.

4.1 Noisy Data – Scanned Models

We have established that the feature extraction procedure allows us to obtain relevant subsets of a model that reflect complexity of its 3D structure. Our next experiment was aimed to assess whether the technique is capable of handling models that were obtained using a 3D digitizer – full 3D view and partial 3D view of 3D objects. Such data is known to be very noisy, often with broken connectivity and missing faces. Ideally, we would like to be able to take a single scan of a 3D CAD model, decompose it into features, and select models from the database that contain the same feature arrangements. We used three CAD parts to create six 3D models – full and partial (one scan) for each CAD part. Once the point clouds were obtained, we faceted them, and then extracted features using FEATURE-DECOMPOSITION($\mathscr{M}, k$) algorithm.

Figures 5 (a), (b) and 6 (a) show correspondence of extracted features for fully and partially scanned models. Figure 5 (c) shows some additional features extracted from fully scanned models that are not present in the partially scanned objects. Extracted features from models that were originally in exact representation are presented in Figure 6 (b).

The performance of the technique is certainly not as remarkable as on previous data set. Although, we believe that the extracted features are meaningful and reflect the structure of the models. In addition, it is clear that there are similarities between feature decompositions of fully and partially scanned models, and 3D CAD models from our database.

5 Discussion and Conclusions

We have introduced a computationally efficient approach to automatic decomposition of 3D models in polyhedral representation into features that could be used to assess similarity between 3D models. The decomposition is based on local surface structure of a model, as a result similar features could be extracted in the presence of partial model information (i.e. partial 3D view of the model). Such locality representation can potentially be used for 3D matching purposes. Further, Scale-Space decomposition technique is robust with respect to noise, therefore it could be used on models that are constructed using devices such as 3D laser scanners.

The notion of feature presented here is highly tuned to the efficient identification of shape and topological categories. Even though, features obtained using our approach could be different from traditional CAD features, they could be used to establish partial similarities between CAD models in polyhedral representation.

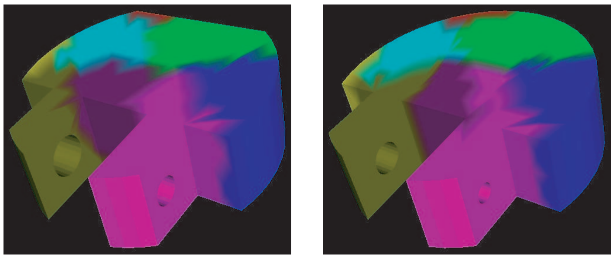


Figure 7. Results of matching between two SWIVELS, with matched regions having similar colors. Features trees are obtained for each model using FEATURE-DECOMPOSITION($\mathscr{M}, k$) with geodesic distance function.

Our work is in its preliminary stages, and we plan to extend its scope by introducing an efficient matching algorithm to assess partial similarity measures. From the above experiments we conclude that in order to perform successful matching, the technique must have the following properties: 1) be tolerant to noise that scanned data introduce; 2) be able to perform many-to-many matching, since it is possible that a feature could get divided in to several features; 3) be efficient, so it could be used in the National Design Repository database [1]. One of the main aspects of such matching technique would be the distance function that assigns a numerical value to a pair of features. Our previous work [1] successfully used such function. That function is based on area and Euclidean distance measurements within features. Please see Figure 7 for a sample view of two models with matched regions.

The other possible directions for our future work: 1) explore techniques to extract features that resemble traditional CAD features; and 2) exploit the possibility of using Scale-Space features as signatures for indexing purposes.

Acknowledgments

This work was supported in part by Office of Naval Research Grant N00014-01-1-0618. We would like to thank Rachana Jethwani for her help in obtaining 3D laser scanned models.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

[1] http://www.designrepository.org

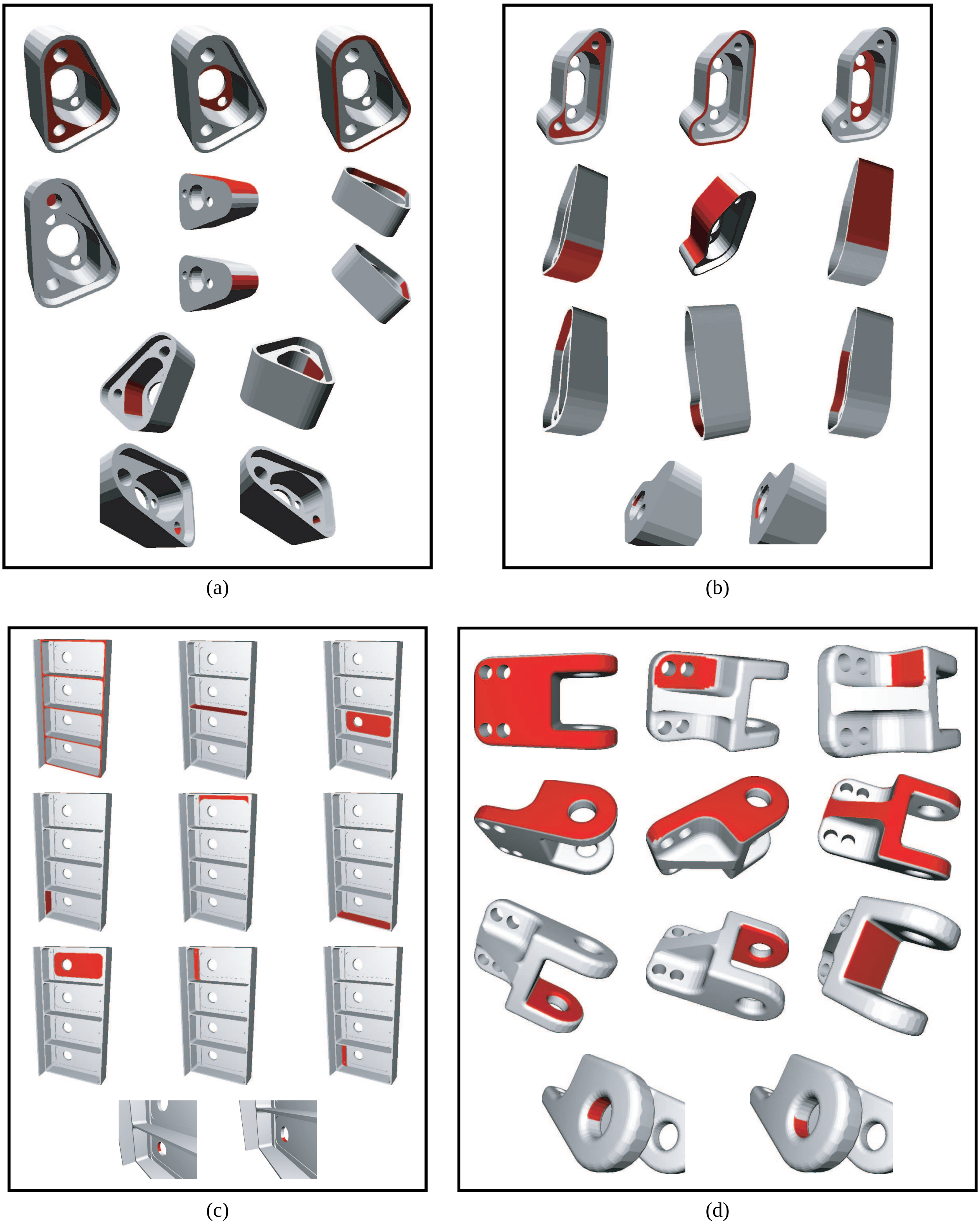


Figure 4. Extracted Features. Only subset of the features are shown in order to illustrate what kind of features are extracted. (a) Part 9, (b) Part 10, (c) Simple Boeing, (d) Cimplex. The names for these CAD models were assigned by the organizations that provided the files to us. We chose to use such names for the purpose of referencing the models within this paper.

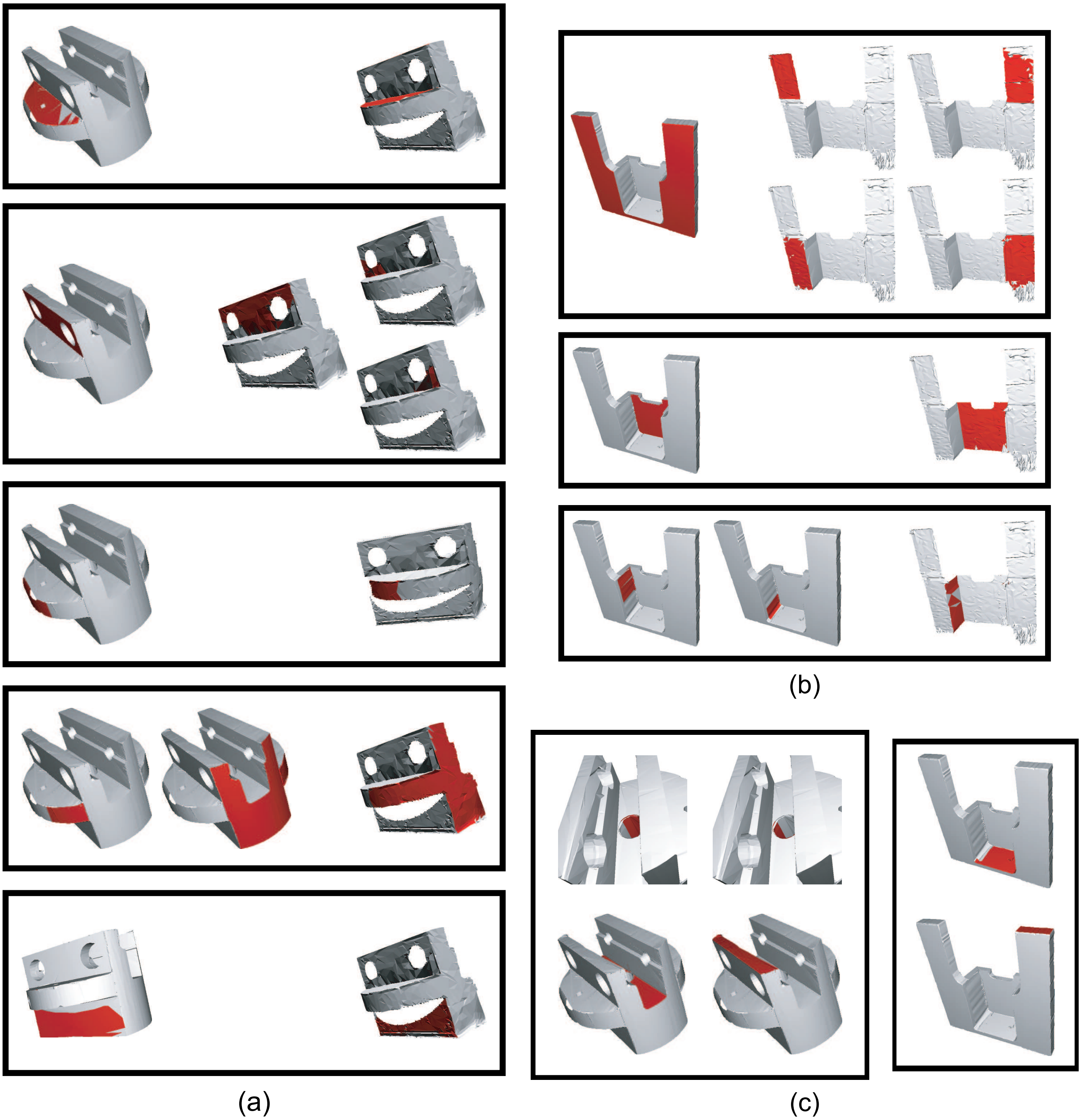
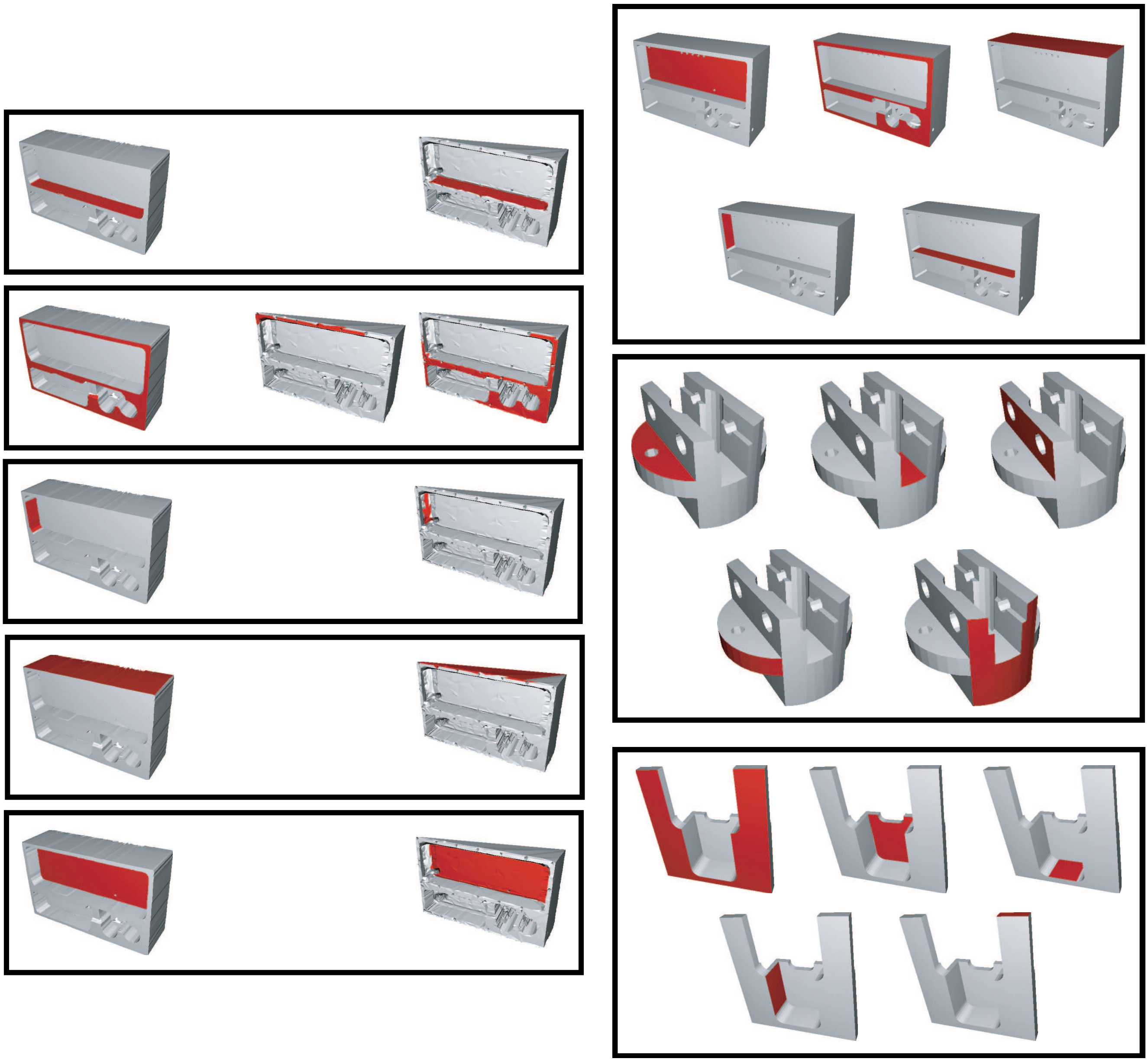


Figure 5. Scanned Data. (a) Socket model. Features for fully-scanned model are on the left, and for partially-scanned on the right. Features are grouped based on correspondence between full and partial models. (b) Bracket model. Features for fully-scanned model are on the left, and for partially-scanned on the right. Features are grouped based on correspondence between full and partial models. (c) Additional features for fully-scanned models that are not present in partial scans.



(a) (b)

Figure 6. Scanned Data. (a) Ex model. Features for fully-scanned model are on the left, and for partially-scanned on the right. Features are grouped based on correspondence between full and partial models. (b) Features for the models converted from exact representation format (ACIS). Note, that features correspond to the features extracted from scanned models.

REFERENCES

[1] Bespalov, D., Shokoufandeh, A., and Regli, W. C., 2003. "Scale-space representation of 3d models and topological matching". In Symposium on Solid Modeling and Applications, pp. 208–215.

[2] Han, J.-H., Regli, W. C., and Pratt, M. J., 2000. "Algorithms for feature recognition from solid models: A status report". IEEE Transactions on Robotics and Automation, **16** (6) [December], pp. 782–796.

[3] Shah, J., Anderson, D., Kim, Y. S., , and Joshi, S., 2001. "A discourse on geometric feature recognition from cad models". ASME Transactions, the Journal of Computer and Information Science in Engineering, **1** (1) [March], pp. 41–51.

[4] Lindeberg, T., 1990. "Scale-space for discrete signals". IEEE Trans. Pattern Anal. Mach. Intell., **12** (3), pp. 234–254.

[5] Shokoufandeh, A., Marsic, I., and Dickinson, S., 1999. "View-based object recognition using saliency maps.". *Image and Vision Computing,* **17**, pp. 445–460.

[6] Elinson, A., Nau, D. S., and Regli, W. C., 1997. "Feature-based similarity assessment of solid models". In Fourth Symposium on Solid Modeling and Applications, C. Hoffman and W. Bronsvoort, Eds., ACM, ACM Press, pp. 297–310. Atlanta, GA.

[7] Regli, W. C., and Cicirello, V., 2000. "Managing digital libraries for computer-aided design". Computer Aided Design, **32** (2) [February], pp. 119–132. Special Issue on *CAD After 2000*. Mohsen Rezayat, Guest Editor.

[8] Cicirello, V., and Regli, W. C., 1999. "Resolving non-uniqueness in design feature histories". In Fifth Symposium on Solid Modeling and Applications, D. Anderson and W. Bronsvoort, Eds., ACM, ACM Press. Ann Arbor, MI.

[9] Cicirello, V. A., 1999. Intelligent retrieval of solid models. Master's thesis, Drexel University, Geometric and Intelligent Computing Laboratory, Department of Mathematics and Computer Science, Philadelphia, PA 19104, June 1999.

[10] Cicirello, V., and Regli, W., 2001. "Machining feature-based comparisons of mechanical parts". In International Conference on Shape Modeling and Applications, ACM SIGGRAPH, the Computer Graphics Society and EUROGRAPHICS, IEEE Computer Society Press, pp. 176–187.

[11] Snead, C. S., 1989. *Group Technology: Foundations for Competitive Manufacturing*. Van Nostrand Reinhold, New York.

[12] Ames, A. L., 1991. "Production ready feature recognition based automatic group technology part coding". In Symposium on Solid Modeling Foundations and CAD/CAM Applications, J. Rossignac and J. Turner, Eds., ACM SIGGRAPH, ACM Press, pp. 161–169.

[13] Shah, J. J., and Bhatnagar, A., 1989. "Group technology classification from feature-based geometric models". Manufacturing Review, **2** (3), pp. 204–213.

[14] Ham, I., Marion, D., and Rubinovich, J., 1986. "Developing a group technology coding and classification scheme". Industrial Engineering,, **18** (7), pp. 90–97.

[15] Thompson, W. B., Riesenfeld, R. F., and Owen, J. C., 1996. "Determining the similarity of geometric models". In Proceedings of the ARPA Image Understanding Workshop.

[16] W.B.Thompson, Owen, J., de St. Germain, H., Stark Jr., S., and Henderson, T., 1999. "Feature-based reverse engineering of mechanical parts". IEEE Transactions on Robotics and Automation, **12** (1) [Feburary], pp. 57–66.

[17] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L., 2001. "Topology matching for fully automatic similarity estimation of 3d shapes". In SIGGRAPH, ACM, ACM Press, pp. 203 – 212.

[18] Bredon, G. E., 1993. *Topology and Geometry (Graduate Texts in Mathematics, No 139)*. Springer Verlag.

[19] Thomasian, A., Castelli, V., and Li, C.-S., 1998. "Clustering and singular value decomposition for approximate indexing in high dimensional spaces". In The seventh international conference on Information and knowledge management table of contents, pp. 201 – 207.

[20] Eckart, C., and Young, G., 1939. "A principle axis transformation for non-hermitian matrices". *Bulletin of American Mathematical Society,* **45**, pp. 118–121.

[21] Golub, G. H., and van Loan, C. F., 1989. *Matrix Computations*. John Hopkins Press.