Shape Semantics from Shape Context

István T. Hernádvölgyi, Giuliana Ucelli, Martin Witzel, Olga Symonova, Loris Delpero, and Raffaele de Amicis

GraphiTech, 38068 Rovereto (TN), Via F. Zeni 8, Italia

Abstract. 3D models play an important role in many industrial applications. Therefore semantic processing for the purposes of comparing, cataloging and archiving shapes is a major concern. Most previous work considers comparisons based on the object's overall geometry or in a reference frame which is computed from the object's geometry alone, disregarding its context. There are also approaches which propose to match feature points to perform context alignment to better analyze a single element. In complex assemblies created in a CAD system, however, the parts (components and layers) are often explicitly marked and named and therefore the geometric context is evident. In this paper we show how this can be exploited with the aid of Knowledge Management tools to establish accurate frames of reference where the individual shapes can be better analyzed.

1 Introduction

In many applications, digital 3D models and shapes play a pivotal role in the creative design process, visualization and analysis. The application domains include Industrial Design, Architecture and Medical Imaging, just to name a few. Therefore extracting, storing and retrieving semantics associated with these models is important to be able to reuse designs.

Semantics stored in a digital encoding of a 3D model is inherently implicit. Hence, it must be extracted from the model's geometry by a dedicated algorithm. Unfortunately, today's state of the art shape matching systems (e.g. [24, 3]) which can retrieve objects similar to a query shape are not accurate enough to distinguish between shapes which have a similar overall shape or have mostly subtle differences. There are dedicated systems, however, which were designed for the geometrical analysis for a very specific task (e.g. [22, 21]) and therefore the algorithms can rely on "hard-coded" domain knowledge to achieve better performance.

In this paper, we investigate the plausibility to gain domain knowledge automatically from the geometrical context of a shape S and use it to derive algorithms which are better suited to analyze S. This is achieved by using context geometry to establish a precise reference frame and therefore the shape descriptor does not have to be invariant under rotation, translation and scaling. These are the usual requirements for descriptors derived to compare 3D shapes of which nothing more than the geometry is known. For this work, our system must be able to establish the shape's context efficiently. In a multi-component 3D model created in a CAD environment, each element has its own geometry and can be given its own name. Moreover, it is relatively straight-forward to produce these identifiers which can be used to isolate the individual geometric pieces. It is expected that the same logical model created by two different designers will have the same components but these will almost certainly be named differently, unless a precise naming convention is enforced. In this paper we propose to use Knowledge Management to establish the correspondence of the design elements and also give examples how the geometric context established by an assembly's topology can be used to derive shape descriptors to analyze individual components.

2 Related Work

Storing and modeling semantics is a major concern in modern design products. The new MPEG-7 standard [17, 16] provides a standard to annotate multi-media content. In the case of a 3D model, it can be used to describe the content of a scene and its semantics, both in human and machine readable format. MPEG-7 also has shape descriptors (2D and 3D), but these are too general for subtle analysis.

The descriptors of Osada et al [20], Kazhdan et al [9] and Novotni et al [19] were designed to mine the Web for 3D models which resemble a query geometry (either uploaded as a full geometry or sketched). Since models posted on the Web are arbitrarily scaled and oriented, these descriptors are invariant under rotation, scaling and translation. It is also common to establish orientation only considering the model's own geometry. These are usually the center of mass and the Principal Component Analysis (PCA) axes [26]. They are used to align the object first and the shape signature is derived with respect to this coordinate system [25]. Using domain knowledge to align the object for better analysis has been considered to compare medical images [13], bones [22] and archaeological artifacts [21]. In these systems the feature points are selected by comparing the sample to other images or by semi automatically identifying regions of interest on the object. Semantics inferred from certain geometric features of the object were considered in [4], but without the context of the objects. Körtgen et al [12] consider matching feature points inferred from geometry to establish context alignment.

Since in CAD models the components are explicitly isolated, to infer context a correspondence has to be established between differently named but logically equivalent elements. This is a typical Knowledge Management (KM) problem which requires an ontology that encodes knowledge about the design element. The use of KM tools has already been recognized as a means to enhance competitiveness of business companies [5, 11, 10]. The currently running WIDE [27] and Wise [1] projects are specifically targeting knowledge management in the Engineering domain. As far as we can tell, we were the first to propose context geometry inferred with the help of knowledge management tools to be considered in the derivation of shape descriptors.

3 Geometric Context

A usual 3D model is made up of a hierarchy of layers. A layer is a geometric grouping of elements which, in practice, often corresponds to a functional cohesion. Figure 1 depicts an engine which itself is a layer of geometries and it

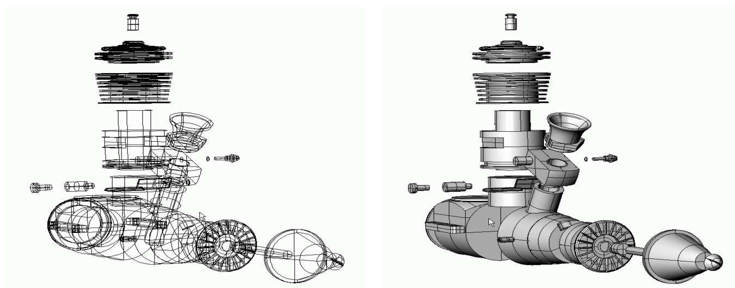


Fig. 1. Complex assembly with layers.

.

also includes sublayers corresponding to smaller components. When the model is exported in some formats, the names of layers and their elements can verbatim be found in the file and can be used to isolate the individual geometries. Elements which belong to different layers could also define topologies which are often representative for a number of different products in the same product line. For example, consider the layout of the car body parts depicted in Figure 2. While there are many different car designs, the elements that make up the body of a car and their topology are very similar for all makes. This layout is obvious for anyone familiar with cars, but it would be a very hard task to reverse engineer from geometry alone. On the other hand, the topology can easily be translated into logical sentences which can be used to identify elements that are connected to each other. For example, the hood of a car is between the front fenders and between the windshield and the front bumper. These axioms can easily be expressed using the `between` predicate (in Prolog style notation).

```
...
between(_,fr_fender,fl_fender,hood).
between(_,windshield,f_bumper,hood).
left(fl_fender).
right(fr_fender).
...

?- between(sedan,X,Y,hood).
```

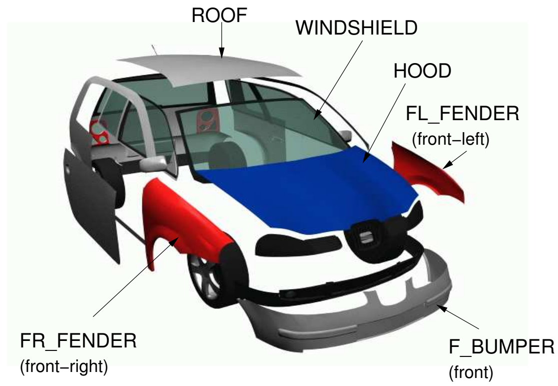


Fig. 2. Typical topology of car body parts.

.

From this, the context geometry of the hood consists of the windshield, the left and right front fenders and the front bumper.

4 Reference Frames

"Knowing" (domain knowledge) that the hood of the car is aligned with the top of the front fenders, we can establish a reference frame in which hoods can be analyzed. Consider the two cars shown in Figure 3. The front fenders and the

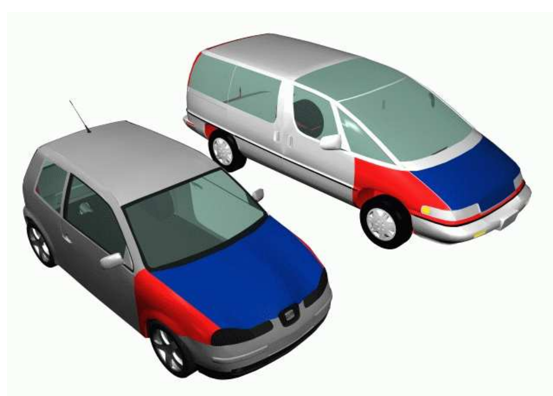
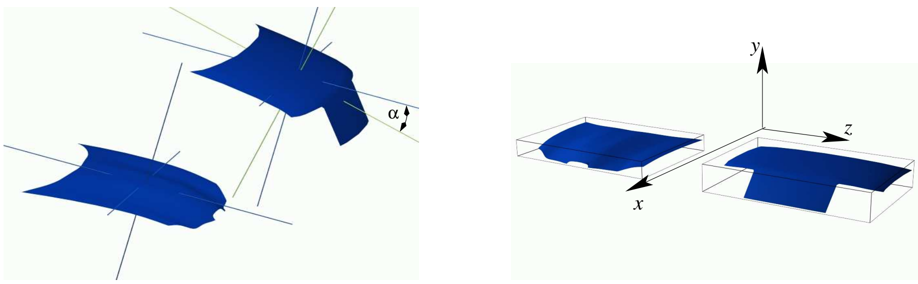


Fig. 3. Two cars with different hoods

.

hoods are highlighted. The fender inclines are quite different for both cars and the van's hood also includes a sticking out grill. Even if the two car models are scaled proportionally and oriented the same way (which is quite unlikely) the actual positioning of the hoods is not a good choice for analyzing their shapes because of their different slopes. A popular approach to establish a coordinate system is to perform Principal Component Analysis. As it can be seen from Figure 4, in our case, the protruding grill tilts the principal axes considerably which makes the resulting PCA coordinate systems inadequate to use as a common frame of reference. α shows the rotation angle between corresponding axes. A common



(a) (b)

Fig. 4. (a) PCA axes from the tessellated hood surfaces, (b) reference frame obtained from the fenders' geometry.

reference frame to compare the hoods can be established as follows. A plane is fitted between the top line of the fenders and this is taken to be the $y = 0$ plane. Since fenders are often slightly curved, the slope can be approximated by finding the linear regression lines with the top line segment of the fenders that also have contact with the hood. The x and z axes are also clear, since the fenders are symmetric and the mid-line between the front fenders toward the front bumper is the "forward direction" (x-axis) which will halve the $y = 0$ plane and the z-axis is perpendicular to these. The origin can either be taken at the centroid's projection on the $y = 0$ plane or at the midpoint between the fenders. The hoods, now in the same reference frame, are also depicted in Figure 4. Also note that regardless how the model is scaled, from the components of the model an accurate scale can often be determined (eg. knowing that the model is to scale and the diameter of the wheels should be 17").

5 The Shape Descriptor

Now we derive two shape descriptors for comparing the hoods. These descriptors are examples, only meant to demonstrate how one can exploit context and domain knowledge. This construction is inspired by [25], but we use 2D and 1D discrete cosine transforms (also used in the JPEG compression algorithm) instead of a 3D Fourier transform (we obtained better surface reconstruction using the same number of harmonic components with the cosine transform). Besides compression, cosine transforms have also been used to obtain feature vectors for face recognition [7]. The dimension reduction comes from the domain knowledge that a car hood is usually a bent piece of sheet metal and therefore it is a surface rather than a volume. The coordinate system is obtained by aligning the hood on the front fenders as described in the previous section. For our first descriptor, D^1, we take the bounding box and subdivide its xz face (Figure 4) into a $N_x \times N_z$ grid. Let f^1_{ij} be the average y value in the grid i, j, $0 \leq i < N_x, 0 \leq j < N_z$. We take the 2D discrete cosine transform of f^1.

$$\omega^1_{uv} = c_u c_v \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} f^1_{ij} \cos\left(\frac{\pi(2i+1)u}{2N_x}\right) \cos\left(\frac{\pi(2j+1)v}{2N_z}\right) \tag{1}$$

where

$$c_u = \begin{cases} \sqrt{\frac{1}{N_x}} \text{ for } u = 0 \\ \\ \sqrt{\frac{2}{N_x}} \text{ for } u = 1, 2, ..., N_x - 1 \end{cases} \tag{2}$$

The shape descriptor is composed of the low harmonics ω^1_{uv} for $0 \leq u < K_x, 0 \leq v < K_z$, where $K_x << N_x$ and $K_z << N_z$.

The other shape descriptor, D^2, is derived from contour points on the surface. We take n cylinders with increasing radii with base on the xz plane and intersect the surface. $f^2(\phi, r)$ is the y value of the intersection point with the cylinder whose radius is r measured at the positive angle ϕ from the x-axis. The shape descriptor is composed of the first few low harmonics of the discrete 1D cosine transform of the n contours.

$$\omega^2_{u,r} = c_u \sum_{i=0}^{N-1} f^2\left(\frac{2i\pi}{N}, r\right) \cos\left(\frac{\pi(2i+1)u}{2N}\right) \tag{3}$$

We verified with our models that, indeed, the low frequency components are the major constituents. In our experiments, for D^1, we sampled the hood surface based on a 50×50 grid and for D^2 we used 5 contours, each sampled at 100 equally spread angles. We took only the first 5 low order harmonics and therefore both feature vectors are composed of 25 real values. Both descriptors were accurate to distinguish our synthetic hood surfaces. As it can be seen from Figure 5, the hoods reconstructed from their respective surfaces, even at this very high compression ratio, preserve characteristics of the original shapes. Again, we would like to emphasize that these shape descriptors would certainly be inadequate to analyze volumes or free form shapes. They, however, illustrate how geometric context (common coordinate system) and domain knowledge (hoods are surfaces) can be incorporated into more informed shape descriptors.

6 Knowledge Management

Building a conceptual model in order to formally describe aspects related to a product domain is necessary to allow the identification of the context from its

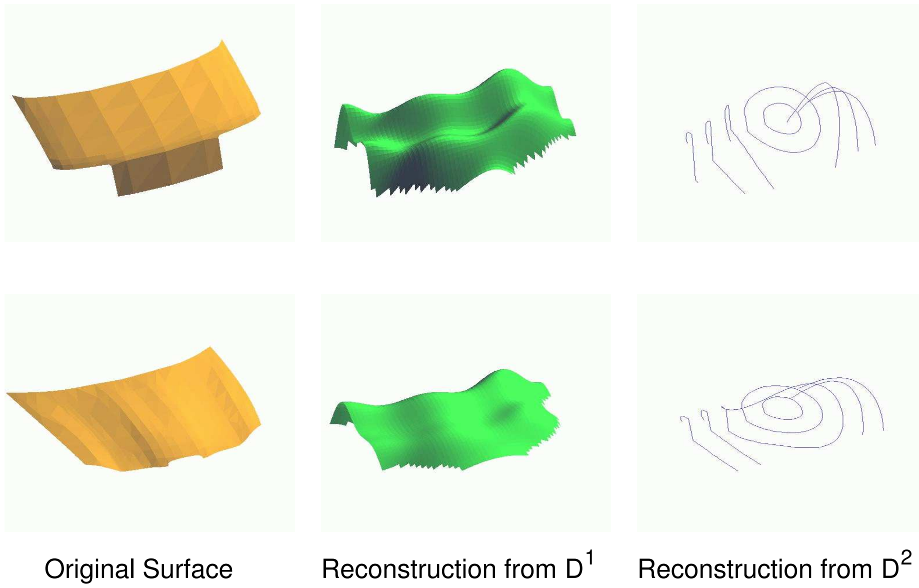


Fig. 5. Surface reconstructions from the descriptors

.

various parts. This is usually accomplished by identifying and formalizing the modeling primitives that describe the domain concepts. From this, one can build a product line's ontology. Ontologies offer the means for sharing knowledge between humans and software agents and provide flexible means for establishing explicit relations between concepts. The purpose of our ontology is to encode domain knowledge which would otherwise be impractical, difficult or even impossible to reverse engineer from geometry alone but can provide useful context information also for the geometrical analysis of the components.

To build this ontology, first, the elements making up the design have to be identified in terms of conceptual entities. We are particularly concerned to reflect the conceptual layout of body panels for different car designs (sedan, cabriolet, hatchback, van, pick-up, etc). The entities, then, are arranged according to specific relations. The standard CAD layering, for example, naturally translates to a "part-of" relationship. This is further complemented by predicates which express the spacial arrangement of the components (eg. the `between` predicate we used earlier). For example, `fl_fender` and `fr_fender` are instances of the class `fender` with different attribute settings to express their orientations (left,right,front or rear). Being `fender`s they also inherit the relationship "adjacent to hood".

Our method needs relatively complete domain ontologies, which are not readily available at the moment. For complex and diverse domains (like engine parts) the encoding of the ontologies would also be a non-trivial task. There are domains, however, where the task of specifying spatial relationships is quite straight forward. In fact, formal logic may not even be needed to obtain a useful encoding.

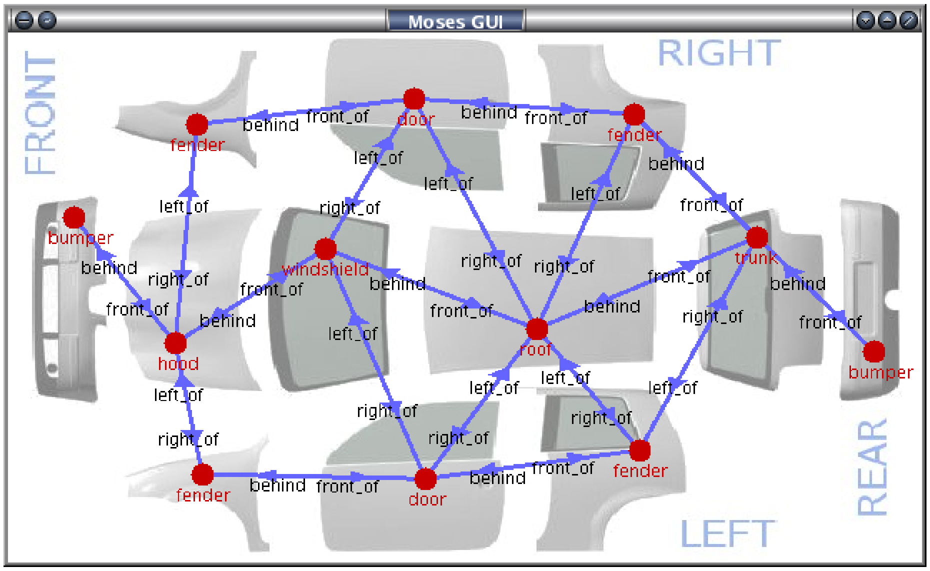


Fig. 6. Spatial relationships of car body panels.

Figure 6 shows a tool in action, which we built to create "context graphs". The spatial relationships (eg. `left_of`, `behind`) and the concepts (eg. `hood`, `roof`) can be loaded dynamically from a domain library. To aid the creation of the graph, the user can also load a blueprint image in the background. The completed graph is then used to navigate the model and to identify the context of individual elements. For example, the front and rear extremities are the two bumpers which can be used to orient the model. Then, a `left_of-right_of` association between two panels indicates that part of the panel's bounding box is a left or right shared edge with the neighbouring panel. While there are more than one fenders in the same model, they can also be identified easily once the model is oriented. For example, the center of mass of the panels (in the car panel domain at least) can directly be aligned to the concept graph.

CAD systems allow the designer to label layers and group elements but this information is not always present in exported models. For us, however, these tags provide valuable information about the hierarchical structure of the model and they can also help to isolate individual components. For this purpose, we have built our own exporter tool for Rhino [23] 3D models. It isolates the components associated with each layer and generates several VRML files, such that the file names verbatim reflect the layer labels chosen by the designer. Our tool also creates a meta file in the MPEG-7 standard format [17], which provides pointers to the files associated with each individual component and it also encodes the original layering hierarchy. This way we have a fully annotated model in which the context geometry is explicitly encoded and is readily accessible.

Layered CAD models of the same multi-component object created by different designers will likely have many corresponding layers, but these will almost inevitably have different names. Thus, to match the groups and layers, we must also provide a term resolution mechanism. For example, we have referred to the cover of the engine as *hood* (usually preferred by American English) while it is also often called *bonnet*. Since the words are actual synonyms (they are in the same synset in WordNet), once hood has been identified as a synonym of bonnet, the terms will be treated as the same component. In the actual model, it is still unlikely that the layer name of the hood would actually be hood or bonnet. In practice, designers often abbreviate the component name which is still suggestive of the original component; eg. `_hd1` for hood or `_bnt` for bonnet. Our algorithm to align the layer names to the most likely concepts is based on the standard Dynamic Programming algorithm used for sequence alignment (eg. in Bioinformatics) [14]. We, however, use a specific cost function which seems to be very effective to tell if the acronym corresponds to the concept. First, as part of the preprocessing stage, we identify the synonyms of the domain concepts. This is achieved using WordNet. To identify the proper sense of a word, we prefer synsets whose description field also contains "hint terms" that indicate that the sense is more relevant to the original domain. In our case these include terms like "auto", "car", "engine" and "engineer". All synsets are considered, but the ones that contain a hint term are given higher priority. The next step is to identify the best match concept term corresponding to the acronym. We calculate a match score for each concept and their synonyms and choose the concept with the highest score. The score corresponds to the ratio of the longest common subsequence (as in [8]) and our modified Levenshtein [15] metric. This latter metric, with actual value $M_{n,m}$ is the dynamic programming solution of the recurrence

$$
\begin{array}{lll}
M_{0,0} & = 0 & \\
M_{i,0} & = M_{i-1,0} + \mathrm{del}_c(A[i]) & \text{boundary condition} \\
M_{0,j} & = M_{0,j-1} + \mathrm{ins}_c(B[j]) & \text{boundary condition} \\
M_{i,j} & = min(& \text{select} \\
 & \quad M_{i-1,j-1} + c(A[i], B[j]) & \quad \text{- substituting } B[j] \text{ for } A[i] \\
 & \quad M_{i-1,j} + \mathrm{del}_c(A[i]) & \quad \text{- delete } A[i] \\
 & \quad M_{i,j-1} + \mathrm{ins}_c(B[j]) & \quad \text{- insert } B[j] \\
 & \ ) &
\end{array}
\tag{4}
$$

Here, A and B are the sequences to be aligned. B, of length m is the acronym and A of length n is the name of the concept. $\mathrm{del}_c(x)$ is the cost of deleting the character x from sequence A and $\mathrm{ins}_c(x)$ is the cost of inserting character x from B. Since acronyms are usually obtained by omitting characters from the original term, we only penalize a deletion from A by 0.5. On the other hand, it is unlikely that one would insert a non-present character into the acronym, hence we give it the full penalty of 1, unless it is a special character (eg. underscore or digit), which we do not penalize at all. $c(x, y)$ is the cost of using character y from sequence B in place of sequence A. Unless $x = y$, or y is a special character, we assign the penalty value 1. With this cost structure, the value of the alignments of _bnt and bent to bonnet are 2.0 and 1.5 respectively. In both cases the longest common subsequence is 3 (b.n.t) and the Levenshtein values are 1.5 and 2.0 (as shown in Figure 7). The ratios give the values $3/1.5 = 2$ and $3/2.0 = 1.5$. _bnt is the more likely alignment, since inserting a special character (_) carries less penalty than having a character in the acronym (e) which does not occur in the concept term.

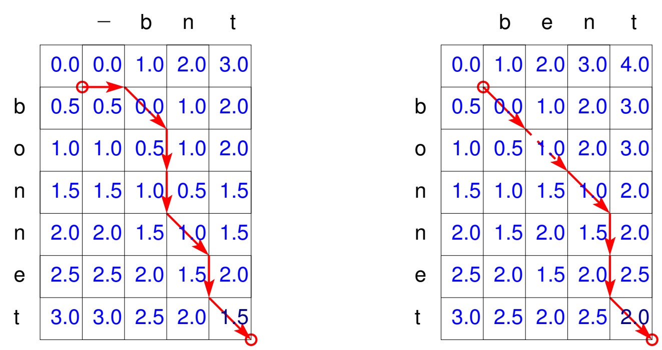


Fig. 7. Aligning _bnt and bent to bonnet.

It is an additional requirement for our project to be multi-lingual. For this reason, we are going to switch to MultiWordNet [18] which allows the alignment of Italian to the famous Princeton WordNet. MultiWordNet brings also the advantage of extending the WordNet lexical database with further syntactic relations that permit to take “phrasets” into account [2]. Phrasets are free combinations of words which are recurrently used to express a concept (called Recurrent Free Phrases or RFPs). Many English and Italian terms in the mechanical domain are considered to be RFPs by lexicographers which cannot be encoded in the original WordNet.

7 Concluding Remarks

In this paper we have described how context geometry can provide the opportunity to better analyze an individual component. For many multi component industrial products, there is often a well known topology of elements. The example we have carried in this paper focuses on the panels comprising a car’s body. In CAD models, the layering hierarchy is also explicitly present when the model is created and this information can easily be available. Therefore, with the use of Knowledge Management tools, one can extract the geometries of any individual component and can also infer its geometric context. This context, in turn, can be used to establish a common frame of reference which provides additional information for analysis of the individual component. We have derived two well performing example shape descriptors to show that it can be relatively simple to incorporate context and domain knowledge.

We also proposed to use an ontology and a WordNet based term resolution engine to establish correspondence between the components of existing models.

We also believe that providing the geometric context to the stylist in the CAD/VR environment can also aid the artist (not only the Mathematical details of shape analysis). For example, when the designer wants to create a hood, generic front fenders would automatically "pop-up" in faint rendering so they are non-intrusive but can lead the "pen" (mouse, 3D mouse or some other input device). A similar approach is used in SpaceDesign [6], where the designer can choose context objects (such as engine volume or passenger) to aid the construction of a car panel surface. The reference frame created by these aid objects can also be exploited for shape analysis.

There is also a wealth of models available on the Internet which do not have proper layering but would still be useful if they could be made available for the designers. These models can be retrieved using existing "context free" search engines and then converted to properly annotated layered models with the help of semi-automatic authoring tools (which use standard techniques, such as PCA analysis for alignment). The conversion of untagged and poorly structured models is a major concern as proprietary designs actually used in industry are rarely made available and the model libraries collected by Web spiders contain many detailed, but unstructured models.

Acknowledgments

Part of this work has been done within the projects MoSeS and SIMI-Pro, financed by the Provincia Autonoma di Trento. The authors would also like to thank the consortium of Aim@Shape for their financial support related to this research activity. The models in this paper were created by Stefano Agostini.

References

1. Y. Barnard and A. Rothe. Knowledge management in engineering: Supporting analysis and design processes in innovative industries. In *Building the Knowledge Economy, Issues, Applications, Case Studies*, pages 931 – 938, 2003.
2. L. Bentivogli and E. Pianta. Extending wordnet with syntagmatic information. In *Second Global WordNet Conference*, pages 47–53, 2004.
3. Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum (EUROGRAPHICS'03)*, 22(3):223–232, September 2003.
4. J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod. Coarse filters for shape matching. *IEEE Computer Graphics and Applications*, pages 65–74, 2002.
5. T. D. Davenport and L. Prusak. Working knowledge: How organizations manage what they know. Harvard Business School.
6. M. Fiorentino, R. De Amicis, A. Stork, and G. Monno. Spacedesign: conceptual styling and design review in augmented reality. In *ISMAR 2002 IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 86–94, 2002.

7. Z. M. Hafed and M. D. Levine. Face recognition using the discrete cosine transform. *International Journal of Computer Vision*, 43(3):167–188, 2001.
8. D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Comm. Association of Computing Machinery*, 18:341–343, 1975.
9. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, 2003.
10. B. Koch and U. Jasnoch. DEPOT: An internet-based knowledge management system for design. In *Proceedings of ISE, Las Vegas*, pages 49 – 51, 2001.
11. B. Koch and M. Koch. An internet-based information service to support the product development process in design offices. In *Proceedings of FAIM, Maryland*, pages 77–82, 2000.
12. M. Körtgen, G.-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *The 7th Central European Seminar on Computer Graphics*, April 2003.
13. G. Krell, H. R. Tizhoosh, T. Lilienblum, and C. J. Moore. Fuzzy image enhancement and associative feature matching in radiotherapy. In *International Conference on Neural Networks (ICNN), Houston, TX*, June 1997.
14. C. S. Wallace L. Allison and C. N. Yee. When is a string like a string? In *International Symposium on Artificial Intelligence and Mathematics*.
15. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707 – 710, 1966.
16. B. S. Manjunath, P. Salembier, and R. Sikora. *Introduction to MPEG-7*. John Wiley & Sons, LTD, 2002.
17. MPEG. http://www.chiariglione.org/mpeg/.
18. MultiWordNet. http://multiwordnet.itc.it/.
19. M. Novotni and R. Klein. 3D Zernike Descriptors for Content Based Shape Retrieval. In *The 8th ACM Symposium on Solid Modeling and Applications*, June 2003.
20. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications*, 2001.
21. A. Razdan, M. D. Liu Bae, M. Zhu, G. Farin, A. Simon, and M. Henderson. Using geometric modeling for archiving and searching 3D archaeological vessels. In *CISST*, 2001.
22. Anshuman Razdan, Jeremy Rowe, Matthew Tocheri, and Wilson Sweitzer. Adding semantics to 3D digital libraries. In *5th International Conference on Digital Libraries (ICADL 2002), Singapore*, 2002.
23. Rhino. http://www.rhino3d.com/.
24. P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark, June 2004. To appear in Shape Modeling International, Genova, Italy.
25. D. V. Vranic and D. Saupe. 3D shape descriptor based on 3D Fourier transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001), Budapest, Hungary*, September 2001.
26. D. V. Vraniĉ, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In *IEEE Workshop Multimedia Signal Processing, Cannes*, 2001.
27. WIDE. http://www.ist-wide.info.