Bianca Falcidieno, Nadia Magnenat-Thalmann, Remco C. Veltkamp (Eds.)

Proceedings of the SAMT Workshop on Semantic 3D Media

First International Workshop on Semantic 3D Media, S-3D, Koblenz, Germany, 3 December 2008

Bianca Falcidieno
Nadia Magnenat-Thalmann
Remco C. Veltkamp
(Eds.)

Proceedings of the SAMT Workshop on Semantic 3D Media

First International Workshop on Semantic 3D Media, S-3D,
Koblenz, Germany, 3 December 2008

ISBN 978-90-9023818-0

Cover image created by the http://www.wordle.net/ web application made by Jonathan Feinberg under a Creative Commons Attribution 3.0 United States License.

Table of Contents

Preface

Research in the fields of representation and processing of the semantics of visual media has received a very high push in the last years. Semantic 3D Media is highly interdisciplinary and its evolution is conditioned by how experts in Computer Graphics will be able to communicate and exchange ideas and solutions with the community of the Semantic Web.

The SAMT workshop Semantic 3D Media (S-3D) wishes to establish a scientific forum for exchanging and disseminating novel ideas and techniques in the emerging research field of Semantic 3D Media. S-3D aims to foster the comprehension, adoption and use of knowledge intensive technologies for coding and sharing 3D media content in consolidated and emerging application communities.

S-3D targets at the scientific community working in the field of 3D graphics and knowledge technologies and encourages a dialogue between researchers and 3D content creators/users in a variety of application domains, such as medicine and bioinformatics, gaming and simulation, CAD and virtual product modelling, archaeology and cultural heritage.

The workshop has broadcasted an open call for papers to attract a representative number of papers from leading researchers working on topics related to semantic 3D media and applications, including but not limited to semantics-driven 3D shape segmentation, formalization and representation of shape semantics, content-based 3D retrieval and classification, semantics-driven 3D visualization, 3D media ontologies, and semantics-based 3D modelling. The present proceedings contain two invited contributions, and seven reviewed papers.

The workshop is partially supported by the project FOCUS K3D funded by the EC FP7, and the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

Bianca Falcidieno (CNR-IMATI, Italy)

Nadia Magnenat-Thalmann (MIRALab, Switzerland)

Remco Veltkamp (Utrecht University, The Netherlands)

November 2008

From Semantics to Pragmatics

David Duke

School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.
djd@comp.leeds.ac.uk

Abstract. An understanding of language is usually built on three foundations: syntax, semantics, and *pragmatics*. If we think in terms of the web, XML addressed portable syntax, providing a common framework for structuring data. RDF, OWL, and other semantic-web standards are working towards a level of semantic inter-operability, allowing the sharing of meaning. But what about pragmatics, the question of *how* the components of a language are used by practitioners to express ideas or to achieve goals? I argue that, although pragmatics may be less amenable to formalisation, it has no less important a role in "semantic media" applications. I will discuss pragmatics in the context of both well developed work in data visualization, and my own more speculative work in domain-specific languages within graphics.

From geometric to semantic 3D content: the FOCUS K3D initiative

Bianca Falcidieno

Istituto di Matematica Applicata e Tecnologie Informatiche - CNR
Via De Marini 6, 16149 Genova, Italy
bianca.falcidieno@ge.imati.cnr.it

Abstract. The paper presents the activities and achievements of the recently started project FOCUS K3D on the topic of semantic 3D content. FOCUS K3D aims at bringing together researchers and industries in Europe that are capable of identifying the needs of the users regarding 3D shape knowledge representation and processing. Moreover, through its dissemination activities it will create awareness of the benefits deriving from the re-use, and preservation of valuable scientific knowledge and resources in terms of 3D models, software tools for 3D manipulation and processing, ontologies and metadata.

1 Introduction

3D media are digital representations of either physically existing objects or virtual objects that can be processed by computer applications. 3D content is widely recognized as the upcoming wave of digital media and it is pushing a major technological revolution in the way we see and navigate the Internet. Beside the impact on entertainment and 3D web, the ease of producing and/or collecting data in digital form has caused a gradual shift of paradigm in various applied and scientific fields: from physical prototypes and experience to virtual prototypes and simulation. This shift has an enormous impact on a number of industrial and scientific sectors, where 3D media are essential knowledge carriers and represent a huge economic factor in many content sectors.

Thanks to the technological advances, we have plenty of tools for visualizing, streaming and interacting with 3D objects, even in much unspecialized web contexts (e.g., SecondLife). Conversely, tools for coding, extracting and sharing the *semantic* content of 3D media are still far from being satisfactory. Automatic classification of 3D databases, automatic 3D content annotation, content-based retrieval have raised many new research lines that represent nowadays some of the key topics in Computer Graphics and Vision research. At the same time, knowledge technologies, such as structured metadata, ontologies and reasoners, have proven to be extremely useful to support a stable and standardized approach to content sharing, and the development of these techniques for 3D content and knowledge intensive scenarios is still at its infancy .

FOCUS K3D believes that *semantic 3D media*, as the evolution of traditional graphics media, make it possible to use and share 3D content of multiple forms,

endowed with some kind of *intelligence*, accessible and processable in digital form and in distributed or networked environments 1. The success of semantic 3D media largely depends on the ability for advanced systems of providing efficient and effective search capabilities, analysis mechanisms, and intuitive reuse and creation facilities, concerning the content, semantics, and context.

After the seminal efforts of the AIM@SHAPE project 0, FOCUS K3D aims at reinforcing the exchange of ideas with the application areas and at disseminating to those application areas emerging techniques in the research field of *semantic 3D media.*

The project aims also at the identification of current issues on knowledge intensive 3D media, which could trace future research and technological directions, and at the establishment of new partnerships to promote innovative projects addressing a highly multi-disciplinary community, both from academia and industry. To this end, it targets scientists not only in CG but in all the disciplines that make strong use of 3D modelling and simulation; professional developers of tools for 3D content creation and management; publishers/dealers of 3D repositories on line; creators of digital 3D content.

2 FOCUS K3D Scenarios

In FOCUS-K3D, specific application scenarios have been chosen as targets of specific dissemination and take-up actions that will demonstrate how semantic 3D content can answer a number of open problems in the content production and processing chain in those domains. In particular we will focus on Medicine and Bioinformatics, Gaming and Simulation, CAD/CAE and Virtual Product Modelling; Archaeology and Cultural Heritage.

The FOCUS-K3D list of application scenarios is obviously not exhaustive, and these example domains have been chosen because they are good representatives of fields characterised by a massive use of 3D digital resources and huge amount of 3D data. Moreover, in these fields, the use of 3D data is not only related to visual aspects and rendering processes but it involves also an adequate representation of domain knowledge to exploit and preserve both the expertise used for data creation and the information content carried.

In these application domains FOCUS K3D will address the needs of the different categories of 3D content providers and users, ranging from the professional creators to the talented amateurs.

2.1 Medicine and Bioinformatics

Medicine on the one hand and structural bioinformatics on the other hand are multi-disciplinary research fields featuring a subtle mix of geometric and knowledge based pieces of information.

In medicine, geometric representations are directly provided by acquisition systems (MRI, CT, etc), and are instrumental in modelling processes. On the other hand,

diagnosis, therapy planning and legal medicine resort to knowledge based technologies putting geometric models in biological and pathological contexts. In structural bioinformatics, a domain concerned with the relationship between the structure of bio-molecules (nucleic acids and proteins) and their function, geometric information is paramount to understand the way molecules adopt their 3D structure (the folding problem) or assemble (the docking problem). Endowing the geometry with complementary attributes (precise type or family of molecules, known binders, connection to metabolic pathways, etc) in a biological environment also calls for knowledge technologies.

2.2 Gaming and Simulation

Game research involves the creation of virtual worlds, in which physical and human behaviours are properly simulated. Obviously, modelling and processing 3D content plays an important role in this application area. Recently it was realised that gaming is a mature field with a high societal and economical impact, which requires multidisciplinary research. Involved disciplines include computer graphics, modelling and animation, (physical) simulation, artificial intelligence and agent technology, human-computer interaction, and semantics. Gaming and simulation is also a very heterogeneous application domain. For example, digital games are no longer just played on PCs or game consoles in living rooms. Instead, there is clear evidence that developments like mobile and ubiquitous gaming are more than just temporary trends. Similarly, games are no longer just played for fun. Impressive examples exist for the gainful application of serious gaming in disaster planning, product development and education.

Members that expressed their interest in FOCUS K3D related activities so far include traditional game developers as well as groups working on serious gaming, universities and research labs as well as commercial vendors, museums, publishing companies, etc. Given the encouraging feedback and positive response we got for our initial activities, we are sure that FOCUS K3D will make some significant contributions in this context.

2.3 CAD/CAE and Virtual Product Modelling

Product Modelling largely contributed to the development of techniques for modelling and processing digital 3D models. It can be informally defined as the whole workflow that stretches from an idea about a new product (e.g. an appliance or a car), to the concept development and shape design, and then to a series of engineering-related steps such as testing, manufacturing or machining the physical object. More recently, many automotive and aerospace companies have heavily invested in Virtual Product Modelling technologies.

Although the digital mock-up (DMU) offers numerous tools for the digital product development process, the workflow of design and redesign is significantly influenced by the usage of 3D data. Products are not just mechanical anymore; in fact, the share

of electronic components and software controlling the mechanical behaviour of products is rising increasingly fast.

In the scientific literature, several proposals exist to employ ontologies for the knowledge-based formalisation of conceptual design know-how and intentions in order to improve retrieval and design reuse but still a lot of work has to be done to couple the semantic information with the geometric aspects of digital shapes. At present the CAD/CAE AWG already comprehends members from automotive, the oil and the electronic engineering industry who expressed their interest in the FOCUS K3D project.

2.4 Archeology and Cultural Heritage

A large part of the European archaeological and cultural heritage exists in digital collections (e.g. virtual museums, digital libraries, scientific repositories) which are becoming more and more demanding in terms of management, preservation, and delivery mechanisms. Images are probably the most common form of non-textual digital content stored, but three-dimensional (3D) content is expected to become predominant. This trend is, however, still in its infancy. In spite of the remarkable scientific and technological advances in areas like digitising 3D artefacts, archiving or presenting the 3D digital content, stakeholders are making little use of it. 3D models and virtual spaces have huge potential for enhancing the way people interact with museum collections..

3D semantic modelling can be beneficial to provide documentation in case of loss or damage, and interactions with precious artefacts without risk of damage. These kind of activities require the acquisition and reconstruction of artefacts, providing high geometric accuracy in the digital models, photo-realism, full automation, low cost, portability andflexibility in applications, while minimising human interaction during the modelling process.

The association with semantics is also crucial to visualise the models properly and retrieve them efficiently from large databases. Efficient retrieval implies equipping 3D content with metadata related to both the whole object and its subparts, developing automatic metadata extraction tools and shape similarity mechanisms to compare objects, providing best practices assisting the processing phase. Semantic 3D media can also be efficiently employed for educational purposes, such as virtual tourism, virtual museums and 3D visualisation of city buildings and monuments, as well as 3D representation of past landscapes and past habitation environments, restoration, reconstruction and visualisation of artefacts.

Acknowledgments. This work is done in collaboration with all partners of the FOCUS K3D Project 1.

References

1. The AIM@SHAPE project http://www.aimatshape.net
2. The FOCUS K3D project http://www.focusk3d.eu.

Semantic-driven Best View of 3D Shapes

Michela Mortara[1], Michela Spagnuolo[1]

[1] CNR IMATI Ge, via de Marini 6, 16149 Genova, Italy
{michela.mortara, michela.spagnuolo}@ge.imati.cnr.it

Abstract. The problem of automatically selecting the pose of a 3D object that corresponds to the most informative and intuitive view of the shape is known as the "best view" problem. In this paper we address the selection of the best view driven by the meaningful features of the shape, in order to maximize the visibility of salient components from the context or from the application point of view. Meaningful features can be automatically detected by means of semantic-oriented segmentations: we tested several approaches with very promising results in the automatic generation of thumbnails for large 3D model databases.

Keywords: viewpoint selection, semantics, segmentation, 3D shapes.

1 Introduction

The problem of automatically selecting the pose of a 3D object that corresponds to the most informative and intuitive view of the shape is known as the "best view" problem. In many applications, like the creation of thumbnails for huge repositories of 3D models or digital catalogues it is necessary to capture a pleasant and informative image of an object; moreover, choosing a specific view represents a mean to apply various Computer Vision techniques in the 3D setting, for instance for shape recognition and classification [1]. Up to now, such snapshots are still manually captured, with extremely high time consumption.

Lately, some approaches to the best view problem have been proposed; in the majority of them, a set of admissible viewpoints are assigned a score with respect to certain characteristics that vary from the bare percentage of visible points from that viewpoint to more sophisticated functions. In [3] the saliency of the visible portion of a shape is used to select the best view, where saliency is strictly related to the mean curvature. In [4] saliency is applied after clustering similar views on the base of view stability.

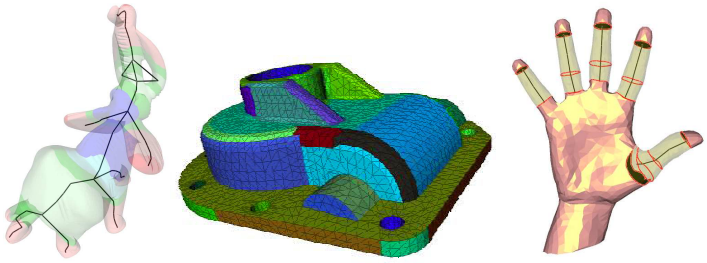
Conversely to other works, [13] takes into account structural information to compute the best view: in fact for a given volume topological feature-based segmentation is performed to yield a set of feature subvolumes and an entropy-based scoring function is evaluated to select the best viewpoint for the volume.

Finally, [2] describes view scoring functions not only related to the geometric complexity of a shape (surface area entropy, visibility ratio or curvature entropy) or

to structural properties (silhouette entropy or topological complexity) but also envisages the exploitation of semantics or meaning of shape components.

In this paper we address the selection of the best view based on the meaningful components (features) of a 3D shape, that is, the quality of a view is bond to the semantics of the displayed features. Such features may be given by a former annotation phase or may be obtained from advanced segmentation algorithms.

Although plenty of segmentation methods have been proposed in the literature [6], the vast majority of them take into account only local geometric attributes of the shape to build up segments, disregarding the structural aspect and eventually the semantics, or meaning, of parts. In this work we just focus on those decompositions identifying parts with a well defined morphological connotation; we have experimented three approaches: the decomposition derived from the Reeb Graph of the shape [9]; the decomposition into a set of Fitting Primitives [8]; and the segmentation into tubular and non tubular features given by Plumber [7] (see Fig. 1). The first method is a topological approach that decomposes a shape into regions of influence of the critical points (maxima, minima and saddles) of a Morse function defined over the surface; the decomposition is strongly dependent of the chosen function. The second method hierarchically fits a set of pre-defined primitives (planes, cylinders, spheres and cones) and annotates the segments accordingly. The last approach identifies tubular features and annotates them with additional geometric and structural attributes. More details on these techniques can be found in [10]. Being these methods developed for triangle mesh representations, we focus on such models; nonetheless, the definition of our scoring function is straightforward for other representation schemes.



(a) (b) (c)

Fig. 1. Segmentations obtained by the Reeb Graph (a), the Fitting Primitives (b) and the Plumber (c) methods.

2 Viewpoint Scoring Function

Let *M* be a triangle mesh embedded in the 3D space decomposed into a certain number of segments, where a segment is a connected cluster of triangles having homogeneous properties.

Like in [2],[4] we determine in advance a finite set of viewpoints uniformly sampled over a sphere surrounding the object, the *viewing sphere*. The sphere is

obtained from an initial icosahedron by applying twice the Loop subdivision scheme, which gives a uniform distribution of 162 viewpoints surrounding the shape (see Fig. 2).

Given a viewpoint *w*, the visibility problem involves determining the portion of *M* that is visible to an observer positioned at *w*: a vertex *v* is visible from a viewpoint *w* if it is not occluded by any other mesh element, that is, the ray from *w* to *v* has no intersection with the mesh. Starting from that, we define the visibility of a segment *s* from the viewpoint *w* as the percentage of its vertices that are visible from *w*:

$$Visibility(s,w) = \#\ visible\ vertices\ from\ w\ in\ s\ /\ \#\ vertices\ in\ s\ . \quad (\mathbf{1})$$

We used the algorithm presented in [5] to compute vertex visibility, which runs in $O(n \log n)$ time being *n* the number of mesh vertices.

Being expressed as the percentage of its visible points, the visibility of a segment does not reflect its relevance with respect to the whole shape in terms of size; in other words, visibility alone may lead to views where small detail features are shown while main structural components are hidden. Thus, we introduce the relevance of a segment *s* as its magnitude as a part of the overall object in terms of surface area:

$$Relevance(s) = Area\ (s)\ /\ Area(M)\ . \quad (\mathbf{2})$$

Since we are going to use semantic-oriented segmentations, we suppose that each segment has a particular annotation or meaning, and some of them may be considered more interesting than others with respect to the object kind or the particular application. Therefore we give more or less emphasis to segments depending on their type (*Type(s)*); finally, we privilege views that display as many segments as possible, taking into account the number of visible segments from that viewpoint (*nvs(w)*). Therefore our scoring function is the following:

$$Score(w) = nvs(w) * \Sigma_{si} (Type(s_i) * Visibility(s_i,w) * Relevance(s_i)) , \quad (\mathbf{3})$$

that is, the score of a viewpoint *w* depends on the number of features visible from *w* multiplied by the sum of each feature's contribution, made up by its visibility, its relevance, and its type. Obviously the feature type is segmentation specific: for Plumber it can be *tube* or *body*; for the Fitting Primitives it can be *sphere*, *cylinder*, *cone* or *plane*; for Reeb it can be *maximum*, *minimum* or *saddle*. We can hard-code specific weight values for each feature type.

Fig.3 shows a few examples of the effects of the different factors on the best view computation.

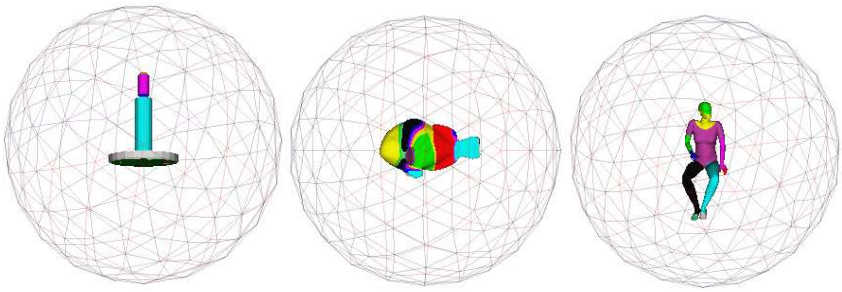
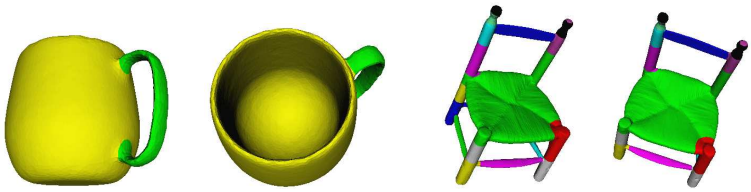


Fig. 2. The viewing sphere surrounding the shapes. Each image is centred on the selected best viewpoint (from left to right the Fitting Primitives, Reeb graph and Plumber segmentations, respectively).



(a) (b) (c) (d)

Fig. 3. Left: best view of a cup computed using specific weights on tubular features (a) and without taking into account the feature type (b). Right: best view of a chair with (c) and without (d) the number of visible feature factor.

3 Results and Future Directions

We tested our algorithm over the database of 400 models generated for the "watertight models" track of the Shape Retrieval Contest (SHREC) 2007 [11], subdivided into 20 classes of 20 models each. We associated to each class two of the three segmentation algorithms taken into account, choosing the more appropriate according to the morphology of shapes in each class (some examples are depicted in Fig. 4 and Fig. 5). In this way, we were able to assess the performance of the best view selection on different segmentations and using different combinations of the factors involved in the score determination.

After the testing we applied the best view computation to automatically capture informative thumbnails for the 3D model database used in the SHREC 2008 - "classification of watertight models" track [12]. Classes were defined by semantic criteria, such as functionality (e.g., "objects for drinking") or presence of characteristic shape features (e.g., "parts with sharp features"). Basing on the a priori knowledge about each class and on the experience of the prior testing phase, we were able to run a shell script which automatically segmented each model according to the segmentation algorithm assigned to its class, computed the best view for that segmentation according to predefined class-specific factors, and finally grabbed and saved a 500X500 pixel image for that model.

Fig. 4. Best view of natural shapes based on the Plumber segmentation (SHREC classes: human shapes, armadillo and four legged animals, respectively).

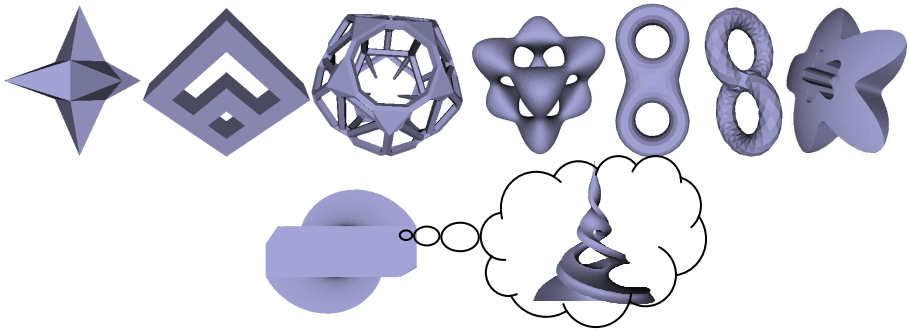


Fig. 5. Top row: Examples of best views for the SHREC class of abstract shapes using the Fitting Primitives segmentation. Bottom: one of the two shapes in this class that required the snapshot to be re-captured manually.

Thanks to semantic-driven segmentations, the selected views are indeed intuitive and informative also compared with previous approaches (see Fig. 6); for instance in the "abstract shapes" class, which seemed the most challenging, only 2 of 41 thumbnails needed to be re-captured manually (the fitting primitives segmentation has been applied, see Fig. 5).

(a) (b) (c)

Fig. 6. Our best view of a hand model using the Reeb Graph segmentation (a) versus the best view computed in [4] (b) and [3] (c). Images (b) and (c) from [4].

When no a priori knowledge about the shape class is available, we found that the fitting primitives method performs best: in fact, being tuned on both planar and rounded surfaces, it is suitable for crafted objects as well as for natural forms. On the other hand, the fitting primitives gives a hierarchical decomposition, from which a

single segmentation must be extracted first (each level corresponds to a specific number of segments). On the SHREC databases we tested segmentations composed by 8 and 13 segments. The best view stability with respect to the number of segments is worth being investigated (Fig. 7 shows an example which suggests a certain stability of best view with respect to such scale changes), as well as the implementation of a hierarchical best view selection.

To conclude, results are promising and we are currently developing a heuristic to automatically set the up-vector of the image, which is still an open problem in the best view context.

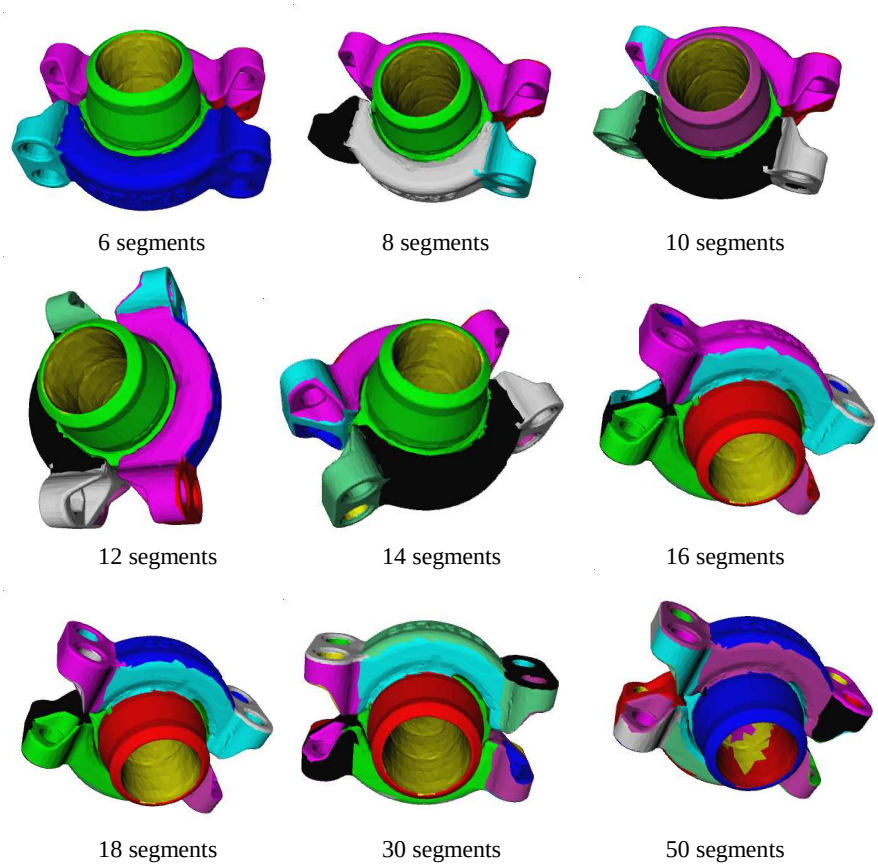


Fig. 7. Best view of a mechanical object segmented by the Fitting Primitives method, selecting different levels in the segmentation hierarchy (different number of segments).

Acknowledgments. This work has been partially supported by the FOCUS K3D Coordination Action (FP7) EU Project. Particular thanks to Dr. Marco Attene and Dr. Silvia Biasotti for extracting the FP and Reeb segmentations, and to Dr. Daniela Giorgi for providing the SHREC models.

References

1. Moktarian, F., Abbasi, S.: Automatic selection of optimal views in multi-view object recognition. In The British Machine Vision Conf. (BMVC ’00), 272--281 (2000)
2. Polonsky, O., Patané, G., Biasotti, S., Gotsman, C., Spagnuolo, M.: What’s in an image? In Visual Computer, 840--847 (2005)
3. Lee, C. H., Varshney, A., Jacobs, D. W.: Mesh saliency. ACM Transactions on Graphics, 24(3): 659--666 (2005)
4. Yamauchi, H., Saleem, W., Yoshizawa, S., Karni, Z., Belyaev, A., Seidel, H-P.: Towards Stable and Salient Multi-View Representation of 3D shapes. In IEEE International Conference on Shape Modeling and Applications 2006 (SMI2006), 265--270 (2006)
5. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. ACM Trans. Graph. 26(3): 24 (2007)
6. Shamir, A.: Segmentation and shape extraction of 3d boundary meshes. In Eurographics 2006 State of the Art Reports , 137--149 (2006)
7. Mortara, M., Patané, G., Spagnuolo, M., Falcidieno, B., Rossignac, J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In: Proceedings of Solid Modeling and Applications (Poster Session), 339--344 (2004)
8. Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical Mesh Segmentation based on Fitting Primitives. The Visual Computer 22(3): 181--193 (2006)
9. Biasotti. S., Giorgi, D., Spagnuolo, M., Falcidieno, B.: Reeb graphs for shape analysis and applications. In: Theoretical Computer Science, vol. 392 (1-3) 5--22. Special Issue on Algebraic and Geometric Computation. Elsevier, (2008)
10. Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., Tal, A.: Mesh Segmentation - A Comparative Study. SMI 2006: 7 14—25 (2006)
11. Shape Retrieval Contest 2007, http://www.aimatshape.net/event/SHREC/shrec07
12. Shape Retrieval Contest 2008, Classification of Watertight Models Track, http://shrec.ge.imati.cnr.it/shrec08_classification.html
13. Takahashi, S., Fujishiro, I., Takeshima, Y., Nishita, T.: A Feature-Driven Approach to Locating Optimal Viewpoints for Volume Visualization. IEEE Visualization 2005: 63 (2005)

Volumetric Modeling of 3D Human Pose from Multiple Video

Berend Berendsen[1]*, Xinghan Luo[2], Wolfgang Hürst[2], Remco C. Veltkamp[2]

[1] Department of Mediamatics, Delft University of Technology
Mekelweg 4, 2628CD Delft, The Netherlands
[2] Department of Information and Computing Sciences, Utrecht University
Padualaan 14, De Uithof, 3584CH Utrecht, The Netherlands
B.J.Berendsen@tudelft.nl, {xinghan, huerst, remco.veltkamp}@cs.uu.nl

Abstract. This paper describes a framework for modeling of 3D human pose from multiple calibrated cameras, which serves as the core part of a player pose-driven spatial game system. Firstly, by multi-view volumetric reconstruction, voxel-based human model is constructed. Secondly, by applying a hierarchical approach with a set of heuristics, fast indirect body model fitting algorithms are used to fit a predefined human model to the reconstructed data, and based on which human poses are modeled and semantically interpreted as certain control inputs to the game.

Keywords: Volume reconstruction, model fitting, tracking, pose semantics.

1 Introduction

We address the modeling of 3D human pose, by fitting a predefined articulated and parameterized body model to the volumetric human body that is reconstructed from multiple calibrated video cameras. Our prospective application is a player pose-driven spatial game, a new type of interactive computer entertainment. Without attaching any extra sensors, the players can control the game by attaining body poses in front of a set of multiple cameras connected to a gaming system. The core parts of such a system are the player pose modeling and semantics interpretation. For our prospective spatial game application, real-time algorithms rather than algorithms that find the perfect fit are preferred, to avoid unpleasant system response delay. Therefore we use an indirect body model fitting method with a hierarchical approach and common sense heuristics to increase speed.

Related work. Voxel-based 3D human model reconstruction [1, 2, 3] is recently recognized as a promising and robust method to recover human shape and motion features, which requires the multiple cameras to be calibrated. For fitting a predefined model to 3D reconstructed data, there are two common approaches: direct and indirect methods. The former is to fit sample points of the data to sample points of the template. To find a correspondence between two sets of points, several closest point and optimization algorithms can be used [2, 12]. The latter is to fit the template body

* Work done while at the Department of Information and Computing Sciences, Utrecht University.

parts around the volume or surface by finding feature points like the center of mass or the use of skin color voxels [3, 11]. Direct matching is more accurate than indirect matching, yet for a spatial game as application where system response speed depends significantly on the pose modeling efficiency, indirect matching is the better and faster solution.

Contribution. We have introduced a number of heuristics such as (i) using a mass seeking box in the torso to keep track of the torso and to be used as estimate for the spine, (ii) the use of the spine vector for finding the neck and check for consistent head direction, (iii) the use of line fitting for the initialization of the shoulders, and the use of clustering for tracking the shoulders, (iv) the placing of spheres on the shoulders and elbows for locating the upper arms and lower arms. In addition, we have systematically evaluated the efficacy of our method.

2 Volume Reconstruction

Camera setup. Our multiple video data are acquired by 4 AVT Marlin Firewire cameras (640×480, 25fps) mounted on 1.9-meter tripods. Camera synchronization is done by software. Because cameras opposite to each other provide the same (mirrored) silhouette, the 4 cameras are setup as shown in Fig. 1 (left), so that no camera is facing directly to any of the others.

Calibration. To specify the correlation between 3D lines in the world and 2D points in the four camera views, we use the respective Matlab camera calibration toolbox [4] and a 6×5 squares checkerboard as calibration object. The square corner coordinates are manually marked from the checkerboard images at different orientations. With the prior knowledge of the number and size of the squares on the checkerboard, the tool box calculates the intrinsic (Fig. 1 middle) and extrinsic parameters (Fig. 1 right) of a camera for each view, and align a global (world) coordinate system for the 4 cameras in this certain setup.

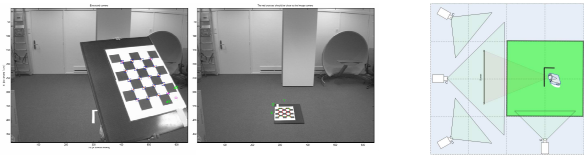


Fig. 1. Camera setup (*left*) and checkerboard calibration (*right*).

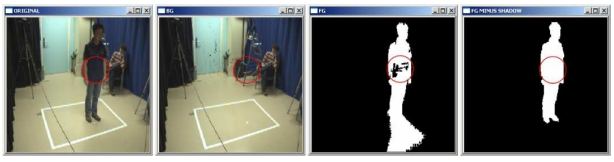


Fig. 2. (a) Video, (b) MoG background model, (c) foreground model, (d) shadow removal and region merging (red circle).

Background subtraction. For reconstructing the model of a person captured by multiple video cameras, first, the person in each video frame needs to be distinguished

from the background and extracted as a foreground mask [2]. We use the Mixture of Gaussians (MoG) method [5, 6] of OpenCV [9] for background subtraction and a simple shadow removal algorithm [7]. Pixel regions are merged according to a set of criteria: horizontal overlapping, X-distance of regions, summed area of merged regions [8]. See Fig. 2.

3D voxel reconstruction. The 2D silhouette points obtained by background subtraction are the projection of a person from 3D to 2D. This projection can be seen as a set of rays or a silhouette cone that contains the 3D points of the subject. The logical conjunction of the silhouette cones of all cameras results into a visual hull [2] that estimates of the subject's 3D shape (Fig. 3). We construct this visual hull using the voxel-based reconstruction method of Kehl et al. [2], resulting in a 3D voxel cloud model of a human subject. First a cube of voxels with a given depth (resolution) in millimeters is created. For each pixel of a camera view a lookup table (LUT) is initialized. To determine the correspondence between the pixels of each view and the voxels in the acquisition space, the center of each voxel is projected onto each view, and the voxel is added to the LUT corresponding to the pixel it was projected on. For each foreground image, an XOR operation detects changes in pixels. If a pixel has become part of the foreground, the corresponding voxels in the LUT are bitwise marked as visible for that view. Voxels that are part of the visual hull are visible from all views. They are selected by bitwise comparison of the LUT's entries.

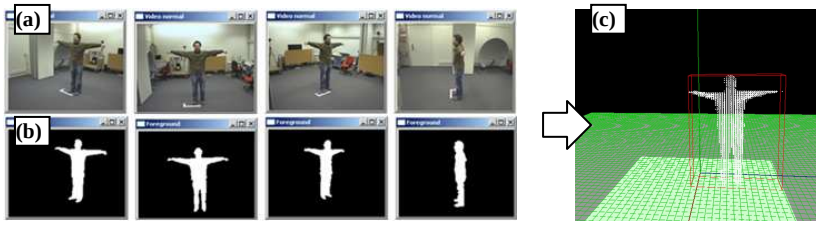


Fig. 3. (a) Four camera views, (b) corresponding silhouettes, (c) voxel reconstruction of visual hull.

3 Pose Modeling

Human body model. Our goal is to extract pose parameters and not model appearance. Therefore, we adopted a simplified generic articulated model for basic shape estimation of a human body and used a combined state vector [10] to parameterize the model. It consists of a 10-joint skeleton (stick figure) representing the basic human kinematic structure, based on which the torso, arms, and legs are modeled as cylinders and the head as a sphere (simple volumetric primitives). The human body parts of the model are defined in relation to body length. See in Fig. 4: Skeleton (left), combined skeleton and body parts (middle), and body parts (right).

Body model fitting approach. We use a fast indirect body model fitting methods similar to [3] and [11]. Template body parts are fitted around the voxel data based on feature points such as the center of mass, and voxels are then labeled to their

corresponding body parts. The fitting order follows a hierarchical approach with common sense heuristics: Due to its distinguished shape, first the head is located. Then the neck and pelvis points are found that determine the torso. From the torso the shoulders can be located, followed by the elbows and hands. In relation to the shoulders and pelvis, the hips are located, followed by knees and feet. This hierarchical approach requires multiple or reoccurring generic algorithms for each body model part. Therefore each fitting module has initialization, estimation, iterative refinement and validation steps as described in the following.

Initialization and global tracking. We use anthropometric measurements [3] to initialize the pose of the person in the scene while the subject is standing straight in a T-shape pose (Fig. 5). Using L and R to denote the length and radius of subsequent body part, and L_{stat} for the length of the subject's stature, which is initialized by the height of the bounding box during the initialization pose. Hence all the other body parts can be initialized using L_{stat} with the definition from [3]: $R_{head} \approx L_{stat}/16$, $L_{torso} \approx 3L_{stat}/8$, $L_{calf} \approx L_{stat}/4$, $L_{farm} \approx L_{stat}/6$, $L_{arm} \approx L_{stat}/6$, $L_{thight} \approx L_{stat}/4$. For the global tracking of the body model, a cube with dimensions $(2*R_{torso})^3$ is placed at the center of the torso C_s that is determined during initialization. For each frame, the mass center of C_s is optimized. The cube is large enough to remain within the torso somewhere along the spine, giving a global estimate of where the body is moving to.

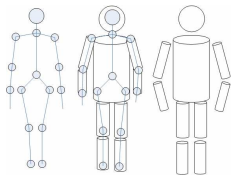


Fig. 4. Human body model

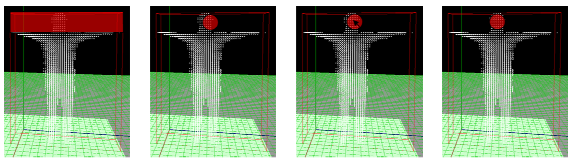


Fig. 5. Head initialization and refinement.

Head fitting and tracking. The head center is initialized by computing the centroid C_h of the bounding box between $z = L_{stat}$ and $z = L_{stat} - 2*R_{head}$ and optimized in a subsequent refinement step (Fig. 5). To estimate C_h, the centroid C_{ul} of the unlabeled head voxels within a sphere centered at the head centroid of the previous frame C_{hp} is computed. From C_{ul} and C_h, a displacement vector $V_d = C_{hp} - C_{ul}$ is calculated with magnitude: $d_m = (R_{head}/m_{hp}) \cdot m_{ul}$ (m_{hp} stands for the previous number of marked head voxels, m_{ul} for the current number of unlabeled voxels within the previous head sphere). The new position of the head is set as: $C_h = C_{hp} + \hat{V}_d \cdot d_m$. For validation, we look at the vector V_s between the torso center C_s and C_h: $V_s = C_h - C_s$. Since the head cannot move significantly into the torso, and the spine can not bend more than 90 degrees at the neck, the dot product between the spine vector and the head displacement vector between the previous and current head center cannot be negative, otherwise a re-estimate has to be made for C_h by relocating the head along V_s with a certain distance d_{rl}. The magnitude d_s of the previous spine vector is $d_s = V_{sp} \cdot \hat{V}_s$. The displacement correction d_c between C_s and C_{sp} is: $d_c = (V_s - V_{sp}) \cdot \hat{V}_s$. Using the relocation distance $d_{rl} = d_s - d_c$, head can be relocated along the spine: $C_h = C_s + \hat{V}_s \cdot d_{rl}$. In Fig. 6 (left to right): If C_h moves into the torso, compute previous spine magnitude, compute displacement magnitude, and relocate head.

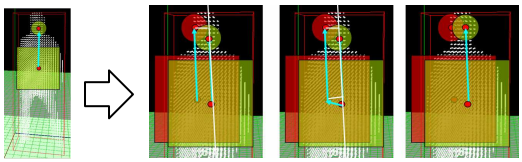


Fig. 6. Head validation and relocation.

Torso fitting and tracking. To initialize the torso cylinder, first the neck centroid C_n in the cube centered at C_h with z-value $L_{stat} - 2 * R_{head}$ is computed. Then the pelvis centroid C_p in the bounding box between $L_{calf} + L_{thight}$ and $L_{calf} + L_{thight} + L_{torso} / 2$, the spine vector $V_s = C_n - C_s$, and pelvis position $C_p = C_n - \hat{V}_s \cdot L_{torso}$ are calculated (Fig. 7). The neck center is estimated by adding a vector with the reversed direction of the spine vector and the head radius R_{head} as magnitude to the head centroid: $C_n = C_h - \hat{V}_s \cdot R_{head}$. The pelvis center is estimated by adding a vector in the same direction with the length of the torso to the neck position: $C_p = C_n - \hat{V}_s \cdot L_{torso}$. C_n is refined by placing a sphere with radius R_{head} on the estimated position. The center of mass of the non-head voxels is computed. C_p is refined by placing the cap of the torso cylinder on the neck and the estimated pelvis position. By fixating the cap corresponding to the neck, and computing the center of mass C_t, the axis of the cylinder can be placed onto C_t. The process is repeated until C_t has been stabilized. Finally, the voxels within the torso cylinder are marked as torso [3, 11].

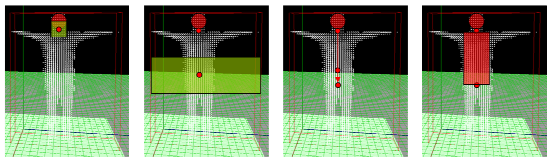


Fig. 7. Initialization of neck, pelvis, and torso.

To estimate the torso radius, firstly a 2D binary image is created by projecting the voxel slices along the spine vector between head and middle torso onto the z-plane while ignoring the head voxels. Least-squares line fitting is applied to the acquired 2D point set to determine the orientation of the torso. Secondly the voxels between the middle torso and pelvis are projected onto the z-plane. After the projection, the points are rotated in line with the orientation of the torso. The maximum x-width and y-height of the point blob now corresponds to the width and depth of the torso (Fig. 8), the average between these two values is then taken as torso diameter.

Fig. 8. Estimation of the torso radius.

Arm/leg fitting and tracking. To locate the arms, first the shoulder centroid C_{sh} must be found. The unit vector $\hat{V}_o$ (collinear to the fitted line) is determined by means of least-squares fitting. The shoulder centroid can be computed as: $C_{sh} = C_n + \hat{V}_o \cdot R_{torso}$, or $C_{sh} = C_n - \hat{V}_o \cdot R_{torso}$ (mirrored to the other shoulder). The shoulder position is then found by placing a sphere with radius R_{arm} on the initial estimate of the shoulder position. The centroid of the torso-marked voxels and the centroid of the unmarked voxels within the sphere are computed. The initial C_{sh} is computed as the point in between both centroids. For estimation of the shoulder location, an outer cylinder is placed around the torso cylinder from the neck cap to the waist (Fig. 9 left). Then the unmarked voxels within the outer cylinder are selected. By applying k-means clustering on the selected voxels, two clusters of arm voxels are found (Fig. 9 right). For refinement, the two neck-to-cluster-center vectors are projected onto the upper cylinder cap (neck). The identification of the shoulders is done based on the distance between the current and the previous shoulder locations. As validation, the distance between these locations should not be larger than a threshold. Otherwise the displacement of the neck is applied on the previous shoulder position to determine the current shoulder position. Finally, the distance between the two shoulders should not be within a certain range. If so the shoulders are set apart.

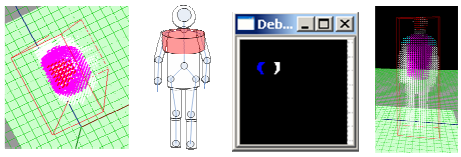


Fig. 9. Estimation of shoulder position.

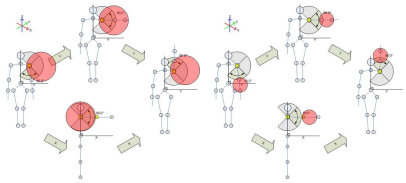


Fig. 10. Estimation of upper (*left*) and lower (*right*) arm direction.

To locate the elbows, an estimate of the direction of the upper arm is made by placing a sphere next to the shoulder C_{sh} with radius L_{arm} centered at $C_{sr} = \hat{V}_{sr} \cdot (R_{torso} - R_{arm})$, where $V_{sr} = C_{sh} - C_n$. Then, the centroid C_{ed} of unmarked voxels is computed (Fig. 10 left). On C_{ed}, another sphere of radius $0.5 * L_{arm}$ is place to refine C_{ed}. By the direction from C_{sh} to C_{ed} and the magnitude of L_{arm}, the elbow position C_e can be estimated as: $C_e = C_{sh} + \hat{V}_e \cdot L_{arm}$, where $V_e = C_{ed} - C_{sh}$. As last refinement step, a sphere with the radius of R_{arm} is placed on C_e to re-compute the centroid. If no voxels are found within radius R_{arm}, then the radius will be enlarged until a centroid is found. As validation, C_e is compared to the previous elbow position. If their distance is larger than a threshold, the previous displacement of the elbow is applied. In the last step, the voxels within the upper arm cylinder are marked.

Similarly, the hand position is found by placing a sphere next to the elbow (in the opposite direction of the upper arm) with a radius of $0.5 * L_{farm}$ centered at $C_{srh} = C_e + \hat{V}_{srh} \cdot (L_{arm} - R_{farm})$, where $V_{srh} = C_e - C_{sh}$ (Fig 10 right). The centroid of the unmarked voxels within the sphere is then computed to find the middle of forearm. By the direction from C_e to the found centroid and the magnitude of L_{farm}, the hand position is estimated as $C_{hd} = C_e + \hat{V}_{hd} \cdot L_{farm}$, where $V_{hd} = C_{hdd} - C_e$. Similarly to the elbow centroid refinement, a sphere with radius R_{farm} is placed on C_{hd} and the centroid of unmarked

voxels is computed. Finally the voxels within the forearm cylinder are marked as a hand.

To locate the legs, we use a similar approach as locating the arms.

Pose semantics. Depending on the specific game, the body pose can now be represented as a 5-tuple BodyPose = {left leg, right leg, left arm, right arm, torso}, where the legs and arms pose can have values {up, side, down}, and the torso pose can have values {up, forward}. For example, a Y-pose can be modeled as {down, down, up, up, up}, denoting a 'yes' input to the game, and a bowing pose can be modeled as {down, down, down, down, forward}, denoting exiting the game.

4 Evaluation and Conclusion

In order to evaluate the performance of our approach, we applied the body model fitting and pose modeling to three video sequences of different persons moving and performing several poses: two Asians (female and male) and one Caucasian (male). Assessment is done by subjective observation. From the initialization of the body model until the end of the sequence, one out of every twelve frames is evaluated with respect to the positions of head, torso, upper and lower arms. The position of the skeletal bone is verified in relation to the voxel and video data. If a bone fits onto the body part, it is marked as a good fit. If a bone is not exactly in but only up to half off, the supposed position is marked as a fair fit. In all other cases, it is marked as a poor fit. The evaluation of the first video sequence containing the Caucasian male is illustrated in Fig. 11 (left). For these frames, body model fitting works very well with almost completely correct matching for everything besides the lower arms. Errors there can be explained by: ‘sticky’ arms while the arms are very close to the torso and stick to it instead of fitting onto the voxel data corresponding to the lower arms; ‘float’ upper arm caused by voxels between the head and the arm; holes in voxel cloud; subject’s long hair etc. The overall fitting evaluation for all three video sequences is illustrated in Fig. 11 (right).

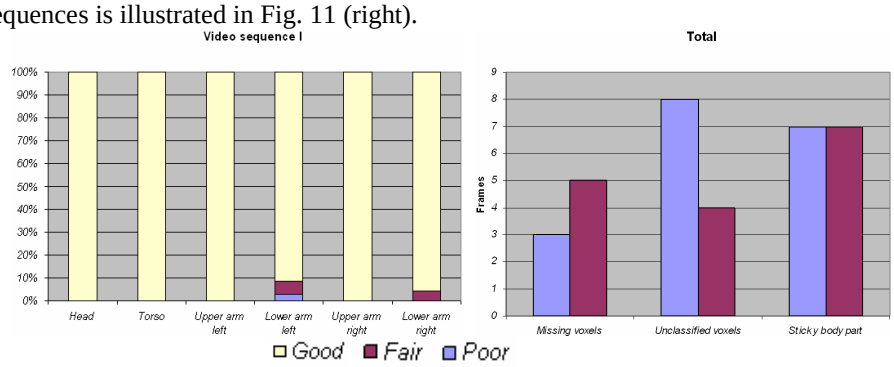


Fig. 11. Example video (*left*) and overall evaluation (*right*).

In this paper we described a framework for modeling of 3D human pose from multiple calibrated cameras (and pose semantics interpretation) as the core part of a spatial gaming system. By using more robust multi-view based 3D pose modeling,

our work opens the way to a more accurate semantics-level understanding of human poses.

For future work, body part constraints can be applied within the validation part of the algorithms to avoid impossible poses or movements of body parts. The evaluation can be extended by testing on more subjects. The framework will be extended to the multi-person case where mutual body occlusion and proximity must be handled.

Acknowledgments. This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

1. Moeslund, T. B., Hilton, A., and Krüger, V. A survey of advances in vision-based human motion capture and analysis. International Journal of Computer Vision and Image Understanding. No. 104, pp. 90–126 (2006)
2. Kehl, R., Bray, M., and Gool, L. V. Markerless full body tracking by integrating multiple cues. In PHI'05 Workshop in Conjunction with ICCV05 (2005)
3. Michoud, B., Guillou, E., and Bouakaz, S. Real-time and markerless 3D human motion capture using multiple views. In 2nd Human Motion Workshop (2007)
4. Bouguet, J.-Y. Camera calibration toolbox for matlab (2007), http://www.vision.caltech.edu/bouguetj/calib_doc/
5. KaewTraKulPong, P., and Bowden, R. An improved adaptive background mixture model for real-time tracking with shadow detection. In 2nd European Workshop on Advanced Video-based Surveillance Systems (2001)
6. Stauffer, C., and Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In CVPR 1999, pp. 2246–2252 (1999)
7. Porikli, F. Human body tracking by adaptive background models and meanshift analysis. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2003)
8. McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. Tracking groups of people. Computer Vision and Image Proceedings of AMDO02, Springer-Verlag, pp. 104–118 (2002)
9. Intel Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv/
10. Thalmann, D., and Magnenat-Thalmann, N. Handbook of Virtual Humans. John Wiley & Sons (2004)
11. Miki, I., Trivedi, M. M., Hunter, E., and Cosman, P. C. Human body model acquisition and motion capture using voxel data. In Proceedings of AMDO02, Springer-Verlag, pp. 104–118 (2002)
12. Tollmar, K., Demirdjian, D., and Darrell, T. Gesture + play exploring full-body navigation for virtual environments. In Proceedings of Computer Vision and Pattern Recognition for Human Computer Interaction (2003)

Accelerating Bag-of-Features SIFT Algorithm for 3D Model Retrieval

Ryutarou Ohbuchi, Takahiko Furuya

4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan
ohbuchi@yamanashi.ac.jp, snc49925@gmail.com

We have previously proposed a shape-based 3D model retrieval algorithm that compares 3D shape based on local visual features. The method first computes a set of multi-scale local visual features from a set of depth images rendered from multiple view orientations about the 3D model. Thousands of visual features per model are integrated into a feature vector for the model by using so-called bag-of-features approach. The algorithm performed very well, especially for models having articulation and/or global deformation. However, the method was computationally expensive; retrieving a model from a 1,000 model database took more than 10s. This because the costs of rendering depth images, extracting local visual features, quantizing these features, and computing distance among the pairs of integrated features can be quite expensive. In this paper, we significantly accelerated the algorithm by using a Graphics Processing Unit (GPU).

Keywords: 3D geometric modeling, content-based information retrieval, GP-GPU algorithms.

1. Introduction

We have previously proposed a shape-based 3D model retrieval algorithm that handles both articulated and rigid models [7]. The algorithm, called Bag-of-Features SIFT, is appearance based, so it accepts a diverse set of 3D shape representations so far as it can be rendered as range images. To achieve invariance to articulation and/or global deformation, the algorithm employs a set of multi-scale, local, visual features to describe a 3D model. We used a saliency-based multi-scale image feature called *Scale Invariant Feature Transform* (*SIFT*) by David Lowe [5] to extract the visual features. To reduce the cost of model-to-model similarity comparison, the set of thousands of visual features for the model is combined into single feature vector by using so-called *bag-of-features* approach (e.g., [2, 3, 9, 11]).

Our empirical evaluation showed that the algorithm performs very well, especially for models having articulation and/or global deformation. For rigid models, such as those found in the Princeton Shape Benchmark [6], the method performed as well as some of the best known 3D model retrieval methods, such as the Light Field Descriptor (LFD) [2] and Spherical Harmonics Descriptor (SHD) [4]. For articulated models, the BF-SIFT significantly outperformed the LFD and SHD.

However, the BF-SIFT algorithm has a high computational cost in computing the feature vector. Rendering depth images, extracting thousands of local visual features per model, and quantizing these features, can be quite expensive.

In this paper, we propose a Graphics Processing Unit (GPU) based approach to accelerate BF-SIFT algorithm. The proposed algorithm employs GPU-based rendering and GPU-based SIFT feature extraction. Along with the use of table lookup in the distance computation stage, the method achieved a query processing time of a few seconds for a hypothetical database having 100,000 models.

2. The BF-SIFT Retrieval Algorithm

We will briefly review the original *Bag-of-Features SIFT* (BF-SIFT) algorithm [7], followed by the method we employed to accelerate the algorithms.

2.1 Original BF-SIFT algorithm

The BF-SIFT algorithm compares 3D models by following the steps below;

1. **Pose normalization (position and scale):** The BF-SIFT performs pose normalization only for position and scale so that the model is rendered with an appropriate size in each of the multiple-view images. Pose normalization is not performed for rotation.
2. **Multi-view rendering:** Render range images of the model from N_i viewpoints placed uniformly on the view sphere surrounding the model.
3. **SIFT feature extraction:** From the range images, extract local, multi-scale, multi-orientation, visual features by using the SIFT [5] algorithm.
4. **Vector quantization:** Vector quantize a local feature into a visual word in a vocabulary of size N_v by using a visual codebook. Prior to the retrieval, the visual codebook is learned, unsupervised, from thousands of features extracted from a set of models, e.g., the models in the database to be retrieved.
5. **Histogram generation:** Quantized local features or "visual words" are accumulated into a histogram having N_v bins. The histogram becomes the feature vector of the corresponding 3D model.
6. **Distance computation:** Dissimilarity among a pair of feature vectors (the histograms) is computed by using *Kullback-Leibler Divergence* (KLD);

$$D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} (y_i - x_i) \ln \frac{y_i}{x_i} \tag{1}$$

where $\mathbf{x} = (x_i)$ and $\mathbf{y} = (y_i)$ are the feature vectors and n is the dimension of the vectors. The KLD is sometimes referred to as information divergence, or relative entropy, and is not a distance metric for it is not symmetric.

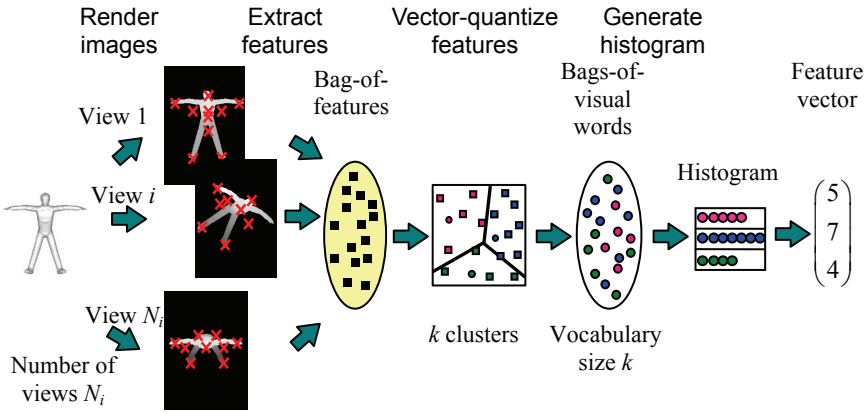


Fig. 1. Range-images of the model are rendered from multiple view angles. Local, multi-scale visual features are extracted from each one of the images using SIFT [4] algorithm. Thousands of SIFT feature per model are vector quantized by using a pre-learned *visual codebook* into *visual words*. Frequencies of visual words are accumulated into a histogram, which becomes an easy to compare and compact feature vector for the model.

2.2. GPU-accelerated BF-SIFT algorithm

The proposed algorithm employs parallelism of a Graphics Processing Unit (GPU) to accelerate two steps, the multi-view rendering step and the SIFT feature extraction step, of the six steps of the algorithm described above,. We keep the vector quantization (VQ) step unchanged. The last step, distance computation step, runs on a CPU but it is accelerated by getting rid of costly calls to logarithmic function by means of table lookups.

1. **Pose normalization (position and scale):** Pose normalization is performed on the CPU.
2. **Multi-view rendering:** Render range images of the model using the GPU, instead of the CPU.
3. **SIFT feature extraction:** As evident in Figure 2, the retrieval algorithm spends the largest amount of time in computing thousands of SIFT features from 42 images of a 3D model. To accelerate the computation, we use the *SiftGPU*, a GPU implementation of the SIFT algorithm by Wu [12]. The SiftGPU does all the work of the SIFT++, that are, construction of a multiresolution image pyramid, detection of interest points, and extraction of features at the interest points, on a GPU.
   While the SiftGPU borrows a lot from SIFT++ by Vedaldi [10], they are not the same. For example, the SIFT++ uses 64bit double precision floating point, while the siftGPU uses 32bit single precision floating point number, for the computation. Thus we compared the retrieval performances of the two SIFT

feature extraction methods. So we experimentally compare the retrieval performance as well as computational cost of the retrieval algorithm using the implementations of the SIFT algorithm.

4. **Vector quantization:** The Vector Quantization (VQ) step is a nearest point query in a very high dimensional (e.g., 1,000) space. It is implemented as a linear search into a set of values whose size N_v is the size of the vocabulary, and runs on the CPU.
5. **Histogram generation:** Quantized local features are accumulated into a histogram having bins, which becomes the feature vector of the corresponding 3D model. This step runs on the CPU.
6. **Distance computation:** Computation of the KLD can be expensive for a high dimensional feature as the computation involves a logarithmic function $\ln(x)$ per element of the feature vector. As the computation is repeated as many times as the number of models in the database, reducing the cost of the log function is quite important for a large database.

   Fortunately, a feature vector produced by the BF-SIFT is a very sparsely populated histogram, in which most of the bins have population zero, and the remaining no-zero elements are small (e.g., <512) positive integers. Thus, the $\ln(x)$ function call can be replaced by a lookup into a small table with no change in performance. The KLD computation using table lookup is performed in the CPU in our current GPU-accelerated implementation.

3. Experiments and Results

We experimentally compared the retrieval performance of our GPU-accelerated algorithm with that of our previous CPU-based implementation by using two benchmark databases: the *McGill Shape Benchmark* (*MSB*) [13] for articulated shapes and *Princeton Shape Benchmark* (*PSB*) [8] for a set of diverse and rigid shapes. We used the same database for learning a visual codebook and for performance evaluation. That is, the codebook generated by using the MSB (PSB) is used to query the MSB (PSB). For each database, the visual codebook is generated by using a set of $N_t = 50,000$ SIFT features, which is chosen pseudo-randomly from all the features extracted from all the views rendered from all the models in the database.

For both SIFT++ (on CPU) and GPU-SIFT, we used parameters of; 42 equally spaced (in solid angle) viewpoints per model, image size of 256×256 pixels per view, scale space having octaves=6, and DOG levels=3. Other parameters for the SIFT++ and the GPU-SIFT are set to their defaults.

We wrote and run the range-image rendering part and SiftGPU feature extraction part using C++, *OpenGL* 2.1.2., and *Cg* 2.0. The vector quantizer and the KLD distance computation parts are implemented by using C++ and run on the CPU. We run the experiment under the Fedora Core Linux on a PC having an Intel Xeon 5440 quad-core CPU (Clock 2.83GHz). The code was single threaded. As for the GPU, we used a mid-range GPU, the Nvidia GeForce 9600GT with 512 MByte of memory, whose core clock is 650 MHz and memory clock is 1.8 GHz.

3.1 SIFT implementations and retrieval performance

We first compare the retrieval performances of the two SIFT feature extractors, the siftGPU [12] that runs on a GPU and the SIFT++ [10] that runs on a CPU. The comparison is done to see the impact of implementation differences, for example, the difference in the precisions of their floating number representations.

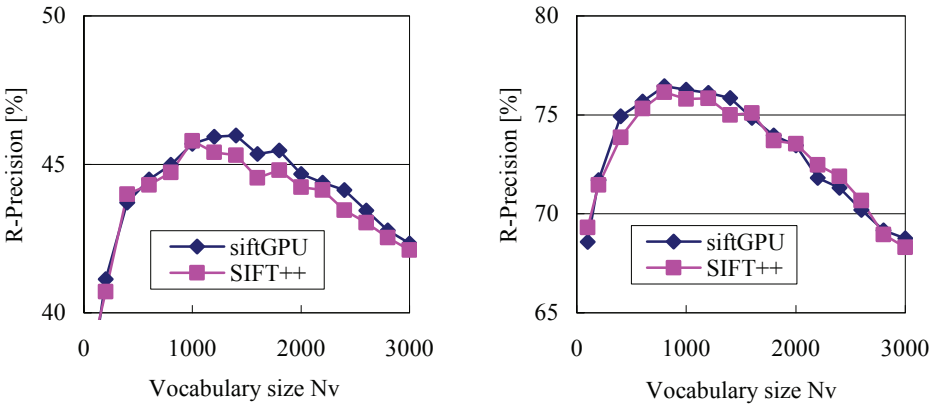
As the performance measure, we used *R-precision* [Baeza-Yates99], which is a ratio, in percentile, of the models retrieved from the desired class C_k (i.e., the same class as the query) in the top R retrievals, in which R is the size of the class $|C_k|$.

Table 1 compares among the two SIFT implementation the average numbers of interest point (i.e., the number of features) per model. For both the PSB and MSB, the number of interest points is essentially the same. The models in the PSB produced more interest points than the MSB, since the PSB models has more detailed shape features than those in the MSB.

Each curve in Figure 2 shows the retrieval performance measured in *R-precision* as a function of vocabulary size N_v. For both of the PSB (Figure 2(a)) and the MSB (Figure 2(b)), retrieval performances are virtually the same for the two implementations; the differences are less than a percentage point.

Table 1. Number of interest points, that are, features, per model for the two SIFT implementations.

	PSB	MSB
SiftGPU	1,989	1,529
SIFT++	1,967	1,570



(a) Result using the PSB. (b) Result using the MSB.

Figure 2. Vocabulary size versus retrieval performances for both CPU-based and GPU-based implementation of SIFT feature extraction. (Note that the vertical scales are different among the two graphs.)

3.2 Computational costs

Figure 3 shows the comparison of computational costs for the three variations of the BF-SIFT algorithm. In the figure, the notation X/X/X/X means that whether the CPU ("C") or the GPU ("G") is used at each of the four stages, that are, (1) depth image rendering, (2) SIFT feature extraction, (3) vector quantization, and (4) distance computation. For the distance computation step, "Cf" denotes the implementation using ln() function, while "Ct" denotes the implementation using table lookup. For example, G/C/C/Cf indicates that the rendering is done on the GPU, while the other stages, that are, the SIFT feature extraction, the vector quantization, and the distance computation using calls to ln() function, are performed on the CPU.

Note that, in this figure, the size of databases is artificially inflated to 100,000 in computing the cost of distance computation. The database size mostly impacts the cost of distance computation. The cost of rendering, feature extraction, and vector quantization are fixed, as they are averages computated from all the models of the database (PSB or MSB). It should be noted that, in reality, the 100,000 model version (if existed) of the PSB (or MSB) will probably have slightly different values of rendering, feature extracton, or vector quantization. But the impact on these values will be small.

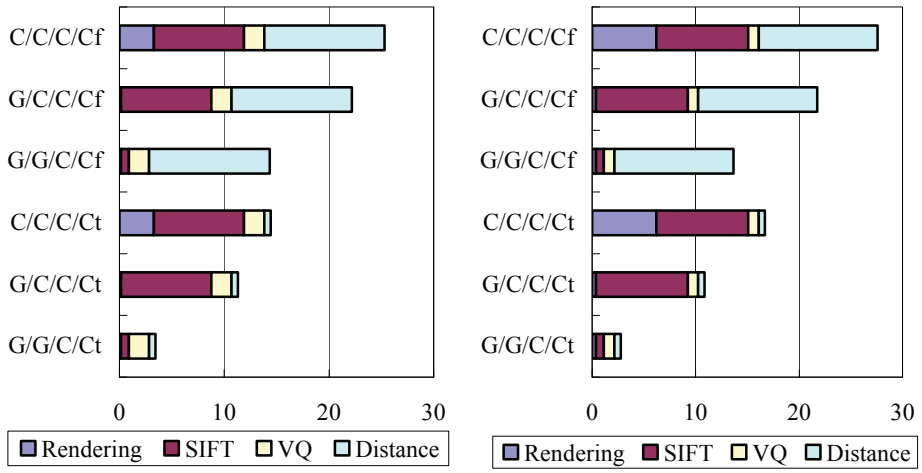
For the all-CPU case using the costly ln() function call, (the case C/C/C/Cf), the distance computation step is dominant, followed closely by the SIFT computation step, and then by the rendering step. Calling ln() function 1,000 times for the 1,000 dimensional feature vector turns out to be qute expensive.

After "modernizing" the implementation so it uses GPU-accelerated rendering, and employing a table lookup to approximate ln() function in the distance computation step running on a CPU (that is, the case G/C/C/Ct), the computational cost of the SIFT feature extraction step becomes dominant. (It is somewhat embarrassing that we did not use GPU-rendering from the start.) Note that the MSB models have higher rendering cost than those in the PSB, since the MSB models have higher polygon counts. The average polygon count of the MSB models is 13,613, compared to 4,373 for the PSB. This is because the models in the MSB are generated as iso-surface meshes from voxel-based models.

If rendering and SIFT feature extraction steps are run on the GPU, and the distance computation on the CPU employs table lookup (that is, the case G/G/C/Ct), the total computation time shrinks to 3.9s for the PSB and 2.9s for the MSB. Compared to the all-CPU implementation that uses table lookup for distance computation (the case C/C/C/Ct), the proposed GPU-based implementation (G/G/C/Ct) is about 3 to 4 times faster.

In the GPU-based implementation (G/G/C/Ct), the cost of VQ has now became the dominant factor. The VQ step for the PSB takes more time than that of the MSB. This is because, on average, a PSB model produces more features than a MSB model (see Table 1.)

Our original algorithm would have taken 25s to query a 100,000 model database statistically identical to the PSB. The proposed accelerated implementation would have performed the query in about 3.9s for the same 100,000 model database.



(a) Cost breakdown for the PSB database queries.

(b) Cost breakdown for the MSB database queries.

Figure 3. Breakdown of computational cost for querying PSB and MSB database. Note that the size of database is artificially inflated to 100,000 in computing the cost of distance computation.

4. Conclusion and Future Work

We have previously published a 3D model retrieval method called Bag-of-Features SIFT (BF-SIFT) that employs a set of thousands of SIFT features [5] to describe a 3D shape. The SIFT is 2D image based, local, multi-scale, and rotation invariant. Our previous experimental evaluation showed that the method is adept at retrieving both articulated and rigid models [7]. However, the method required significant amount of computation, especially for feature extraction.

In this paper, we proposed a GPU-based algorithm that performs multi-view range image rendering and SIFT feature extraction of the BF-SIFT algorithm on the GPU. We also replaced logarithmic functions in the distance computation step with table lookups. Due to these improvements, the method achieved 3 to 4 times speedup with virtually no impact on the retrieval performance. On a hypothetical database similar to the PSB [8] but having 100,000 models, the proposed algorithm would achieve the query processing time of a few seconds.

An analysis of the accelerated implementation indicated a new target for acceleration, the vector quantization and distance computation steps. In the future, we intend to investigate better algorithms for both CPU-based and GPU-based approaches to accelerate these two steps. For example, to vector quantize more efficiently, we we intend to investigate various approximate nearest neighbor algorithms for higher dimensions, e.g., ANN [6].

Reference

1. R. Baeza-Yates, B. Ribiero-Neto, *Modern information retrieval*, Addison-Wesley (1999).
2. D-Y. Chen, X.-P. Tian, Y-T. Shen, M. Ouh-young, On Visual Similarity Based 3D Model Retrieval, *Computer Graphics Forum*, **22**(3), 223-232, (2003).
2. G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual Categorization with Bags of Keypoints, *Proc. ECCV '04 workshop on Statistical Learning in Computer Vision*, 59-74, (2004)
3. R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from Google's image search, Proc. ICCV'05, Vol. II, pp.1816-1823, (2005)
4. M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation Invariant Spherical Harmonics Representation of 3D Shape Descriptors, *Proc. Symposium of Geometry Processing 2003*, 167-175 (2003)
5. D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *Int'l Journal of Computer Vision*, **60**(2), November 2004.
6. D. Mount, S. Arya, ANN: A Library for Approximate Nearest Neighbor Searching, http://www.cs.umd.edu/~mount/ANN/
7. R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for shape-based 3D model retrieval, 93-102, *Proc. IEEE Shape Modeling International* (*SMI*) 2008, (2008).
8. P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton Shape Benchmark, *Proc. SMI* 2004, 167-178, (2004). http://shape.cs.princeton.edu/search.html
9. J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in Videos, *Proc. ICCV 2003*, Vol. 2, pp. 1470-1477, (2003)
10. A. Vedaldi, SIFT++ A lightweight C++ implementation of SIFT. http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html
11. J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, *Proc. ICCV05*, Vol. II, pp.1800-1807, (2005)
12. C. Wu, SiftGPU: A GPU Implementation of David Lowe's Scale Invariant Feature Transform (SIFT) , http://cs.unc.edu/~ccwu/siftgpu/
13. J. Zhang, R. Kaplow, R. Chen, K. Siddiqi, *The McGill Shape Benchmark*, (2005). http://www.cim.mcgill.ca/shape/benchMark/

The SALERO Virtual Character Ontology

Tobias Bürger[1], Philip Hofmair[2], Gert Kienast[2]

[1] Semantic Technology Institute, STI Innsbruck, University of Innsbruck, 6020 Innsbruck, Austria,
tobias.buerger@sti2.at
[2] Institute of Information Systems & Information Management, JOANNEUM RESEARCH, 8010 Graz, Austria,
Philip.Hofmair@joanneum.at and Gert.Kienast@joanneum.at

Abstract. The SALERO project observed a lack of ontologies for the description and annotation of characters in media production. In this field ontologies could be used to support media asset management, information retrieval, automated production or reuse.
This paper presents the SALERO Virtual Character Ontology which can be used to describe and annotate characters in media production and game design to support aforementioned scenarios.

1 Introduction

One goal of the SALERO[3] [1] project is to create ontologies which support the annotation and semantic search of media assets. The main motivation for using ontologies in the project is to overcome the known drawbacks related to the limited reusability capabilities of tools used in the video games and audiovisual entertainment industries which are due to the lack of metadata and formal descriptions of media assets. This paper presents the SALERO Virtual Character

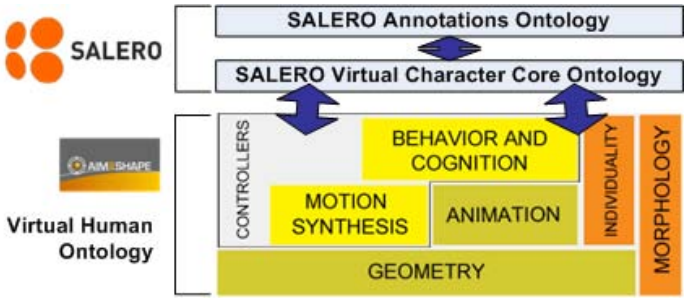


Fig. 1. SALERO Virtual Character Ontology Framework

Core - ontology (SVCC) which has been built following the guidelines in [2]

[3] http://www.salero.eu

and reflecting requirements from the end user partners in the SALERO project which intend to use the ontologies to support annotation, semantic search or communication of information about characters.

The SVCC ontology is based on the AIM@SHAPE[4] ontology for virtual humans (VH) [3] and is itself extended by the SALERO Annotations - ontology (SA) as depicted in Figure 1: (1) The SVCC ontology extends the VH ontology with concepts to describe the personality of characters, genre- and production-related information. (2) Additionally, the SA ontology further abstracts the modeled domain. The SA ontology is supposed to be used for annotation by end users.

2 Representational Requirements

In this section we summarize requirements on the SVCC ontology.

2.1 Motivating Scenarios

The following scenarios of our user partners were identified.

Partner 1 is a games development company whose intention is to support digital asset management and search & retrieval of game assets for reuse in different stages of the games development pipeline.

Partner 2 is creating 3D animations based on a story plot and variable input data automatically. A vital functionality of their tool is to be able to automatically select the right content in terms of fitting to a characters' personality, mood, emotions, context and situation.

Partner 3 is a university of art. One research group of this university is a studio that has the goal to achieve deeper emotional experiences in interactive media and to create production methods to develop content and technology simultaneously in the multitalented teams involving artists, designers, scientists and technology experts. The motivation of using ontologies in 3D content production for this partner is given in the research phase and during character animation.

Partner 4 is a vertically integrated animation production company, focusing on childrens television, with the capabilities of producing cross-media exploitation of properties through website design and development, publishing, or DVD and interactive development. The ontology shall support efficient search and retrieval for fast and easy asset retrieval and use.

2.2 Competency Questions

In order to determine the scope of the ontology we asked the partners to provide a set of competency questions the ontology-based system shall provide answers for. Some examples from the total amount of around 100 competency questions are given below:

[4] http://www.aimatshape.net/

1. Find the head of a caucasian man with a surprised emotion.
2. Find a fictional figure female hero character with supernatural powers.
3. What are the general qualities (status, values etc.) of a specific character?
4. Is this asset derived from another existing asset?
5. Are there any characters with a sad female voice?
6. Which movements are available for a specific character?
7. For which character is a specific property suitable?
8. What assets are visually similar to a specific asset?

3 Conceptual Model

The SVCC ontology extends the AIM@SHAPE VH ontology at several points. Most notably it adds a set of individual descriptors which can be used in the VH ontology to define the behavior of a character. Figure 2 gives an overview of the model and it's central concepts[5]:

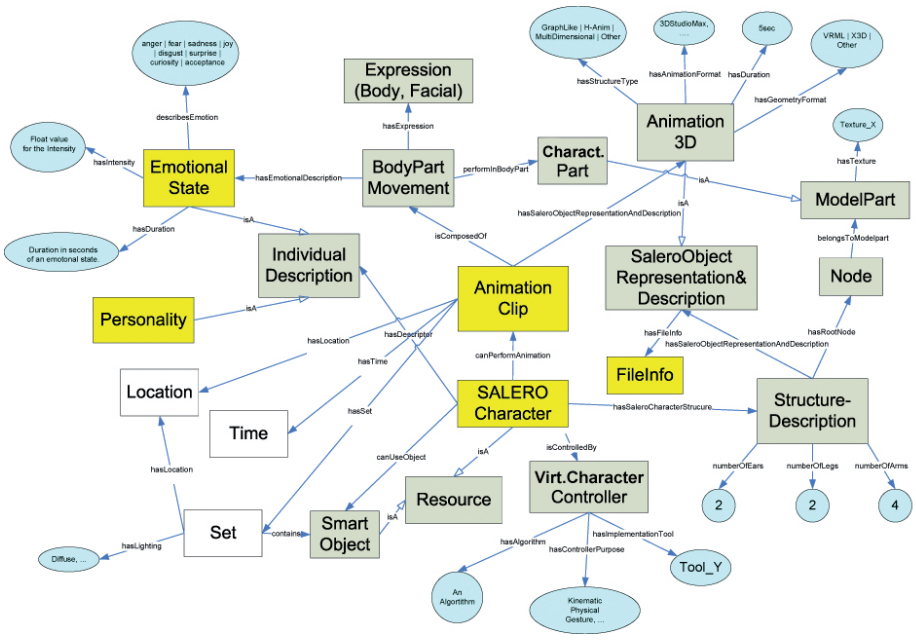


Fig. 2. SALERO Virtual Character Core Ontology

SALERO Character: This is the core concept in our ontology. The *SALERO Character* – concept is modeled as a subclass of the *AIM@SHAPE Virtual*

[5] A bigger picture of the ontology is available in Figure 3 in Appendix A

Human – concept and extends it in several dimensions. A character may most notably have a structural description, a set of animation clips, and a individuality description including personality – and emotional state – descriptions.

Animation Clip: This concepts extends the definition of the *Animation Sequence* concept from the AIM@SHAPE VH ontology. It defines suitable animation sequences that can be applied to the virtual human using body part movements which are either facial animations or body animations accompanied with an emotional descriptor. An *Animation Clip* may have an object representation in the form of a *3D Animation* which might be structurally described as well as physically manifested in a specific animation format (cf. [3]).

Emotional State: As the application partners want to annotate and find a character according to his emotion, *Emotional State* is another central point. The VH ontology characterizes an *Emotional State* using values from Whissel's activation-evaluation space which captures emotions in the dimensions activation and evaluation [4]. Animation gestures can be associated to emotions (cf. [5]).

Personality: *Personality* is an individual descriptor of a SALERO character which extends the *Personality*–concept from the VH ontology. The personality of a *SALERO Character* captures humanoid dimensions of a character like his social role, special abilities, species, or demographics. This extension will be detailed below.

Modeling Personality of a Character: The SVCC ontology adds a set of descriptors to specify the personality of a character such as his demographics, his abilities, his social behaviour or his role in a media production.

The individual descriptors to express the personality of a character are based on the *General User Model Ontology (GUMO)* [6] which we extended to cover properties of fictive, non-human characters. Based on GUMO the following descriptors have been added to the SVCC ontology:

Personal History captures historical information about a character (cities where he lived in, jobs he had, stories about the character, etc.)

Demographics models demographical information like birth place, origin, race, marriage status, etc.

Characteristics models social aspects of the character like if he is artistic, dominant, shy, etc.

Abilities captures abilities of the character like if he is able to walk, to drive a car, to dance, etc.

Special Abilities include characteristic abilities of a character like if he is artistic, shy, dominant, etc.

Social Role models the role of a character in the society like his job position, etc.

Personality models aspect characterizing the personality of a character like if he is introvert, intelligent, romantic, stupid, etc.

Contact Information represent the contact information of a character.

Further extensions have been added to the morphological descriptors of a character in order to express typical gestures which are characteristic for a character and to define the species of a character (e.g. human, robot, animal).

Besides that, the SVCC ontology models the role of a character in a plot and the relation of a character to other characters or the target audience. The core concepts which were added to specify the role of a character in a story are:

Relationship A character can have typed relationships with other characters (e.g. a character is in love with another character) which is based on his role as part of a story.

Role The role of a character is characterized through his plot function, a narrative stance, a dramatic need, and the text available for the role.

Implementation: On the basis of the conceptual model that was done in UML and text, the ontology was built in OWL, reusing the AIM@SHAPE ontology schema and altering or extending it as explained above.

4 Ongoing and Future Work

The ontologies built in SALERO are supported by an ontology workbench which provides ontology management-, semantic search- and annotation-support as briefly described in [7]. We are currently building the SALERO annotations ontology which abstracts from the high complexity of the virtual character core ontology and which will be used for concept-based annotation by the end users in SALERO.

Acknowledgements: The research leading to this paper was partially supported by the European Commission under contract IST-FP6-027122 "SALERO".

References

[1] Haas, W., Thallinger, G., Cano, P., Cullen, C., and Bürger, T.: "SALERO: Semantic Audiovisual Entertainment Reusable Objects" In: Proceedings of the first international conference on Semantics And digital Media Technology (SAMT), December 6-8, 2006, Athens, Greece.

[2] Noy, N. F. and McGuiness, D. L. "Ontology Development 101: A Guide to Creating Your First Ontology", 2001

[3] Gutierrez, M., Garca-Rojas, A., Thalman, D., Vexo, F., Moccozet, L., Magnenat-Thalman, N., Mortara, M., and Spagnuolo, M. "An ontology of virtual humans: incorporating semantics into human shapes" In: The Visual Computer, vol.23 (3) pp.207-218 (2007)

[4] Whissel, C. M.: "The dictionary of affect in language" In: Plutchnik, R. and Kellerman, H. (Eds) Emotion: Theory, research and experience: vol 4, The measurement of emotions. Academic Press, New York, 1989.

[5] Garcia-Rojas, A., Vexo, F., Thalmann, D., Raouzaiou, A., Karpouzis, K., and Kollias, S. “Emotional Body Expression Parameters In Virtual Human Ontology” In: Proceedings of the 1st Int. Workshop on Shapes and Semantics, 2006.

[6] Heckmann, D., Schwartz, T., Brandherrm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. “GUMO the General User Model Ontology” In: Proceedings of the Tenth International Conference on User Modeling (UM 2005), 2005.

[7] Bürger, T.: “The Need for Formalizing Media Semantics in the Games and Entertainment Industry” In: Journal for Universal Computer Science (JUCS), 2008

Appendix A: The SALERO Virtual Character Core Ontology

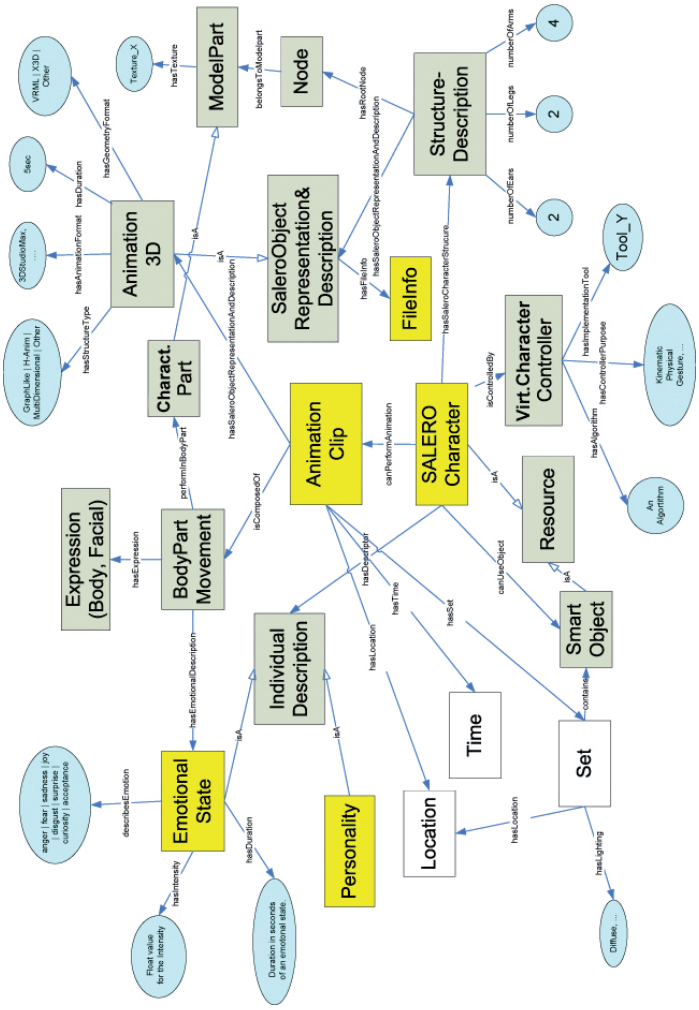


Fig. 3. SALERO Virtual Character Core Ontology

Overview: Semantic Management of 3d Objects

Claudia Ciorascu*, Christophe Künzi, and Kilian Stoffel

Information Management Institute, University of Neuchâtel, Switzerland
{claudia.ciorascu,christophe.kuenzi,kilian.stoffel}@unine.ch

Abstract. In this paper we illustrate an approach for managing 3d objects based on the semantics attached to the objects. Built over a fast indexing mechanism for storing 3d data, a semantic framework for adding and sharing conceptual knowledge about spatial objects is presented.

1 Introduction

The amount of three–dimensional data has drastically increased over the last couple of years. The reasons for this increase are manifold: the availability and the easy use of laser scanners, the improved quality of software for creating 3d data from 2d images, etc. Meanwhile, the development of web technologies led to new possibilities in sharing information. In addition to the traditional type of data these technologies can also be applied to 3d data. Although the need for semantic management of 3d data has been acknowledged for quite a while, there is still a huge deficit of adequate tools. In this paper we will give a short description of an approach and the corresponding tools helping a user to manage, analyse and share information and knowledge about a spatial environment.

From a technical point of view, our system is based on two major components: an efficient storage module for multi–dimensional data and a concept–based representation module for the objects identified in the data. These two components are linked through a query modeling and transformation processing layer. At the basis of the storage module we built a fast indexing structure for storing 3d points. Here we limited ourselves to 3d data, however it can be extended to multidimensional data. The overall system allows 3d queries based on concepts defined in a spatial ontology. Even if some basic concepts are defined in a static reference ontology, the system's knowledge base can be extended by adding the user's own concepts. The points returned by the queries are related to the concepts defined by the user and therefore extended by the new semantics defined in the ontology. Our illustration is build on architectural data. Architectural structures are often based on very complicated models as they have to take into consideration technical as well as aesthetic aspects and therefore ask for semantic integration. To test our approach we propose an ontological model combined with a real set of 3d data of the Pantheon in Rome made available by the Karman Project[1].

* This work was support by the Swiss National Science Foundation, Grant no. 200021–109476

[1] http://www.karmancenter.unibe.ch/

As a first step, we selected an appropriate indexing method to guarantee a very good performance. From the multitude of possible techniques (kd–trees [1], grid files [2], buddy–trees [3], etc.), we finally opted for an adapted version of the X–Tree approach [4], as it satisfies best the needs of the semantic part of our system (see [5]). Based on B–Trees, both the X–Tree and R–Tree [6] structures use the same indexing methods, but the major problem with R–tree–based indexes is that the overlap in the directory is increasing very rapidly with growing dimensionality of the data. In the X–Tree approach, the supernodes are created during insertion only if there is no other possibility to avoid overlap, which will increase the speed of the queries made in the spatial database system. After the storage, we defined the base ontology, the main criteria being clarity and coherence. The basis of the system is a simple geometric ontology. We consider this geometric ontology as a reference system and therefore it should not be modified by the user. In our universe of discourse this is the minimal ontological commitment. It is of course possible to extend the definitions from this ontology. Different user–ontologies could be developed due to the different perceptions of the domain based on cultural background, education, ideology. Finally, we developed the system in such a way that it becomes possible to extend the queries, allowing the user to interact with our system and to introduce the kind of concepts he needs to analyse the spatial data. These different steps are presented in the reminder of the paper.

2 The Reference Ontologies

The system's knowledge base is mainly comprised of two groups of ontologies: the upper (reference) group and the lower (user) group. This difference between the used ontologies has been shown already in [7]. Without losing in generalisation, the user can describe a spatial object in a specific environment by actually constructing the 3d object from elementary shapes. Each elementary shape is described mainly by a transformation (scaling, translation, rotation), one or more positions and one or more dimensions. Since each transformation can be expressed in different ways or is shape–dependent, the upper ontologies comprise the description of different systems and mathematical models that might be used.

2.1 The Coordinate Systems Ontology

The coordinate systems ontology defines a few systems for describing a position in space: cartesian, spherical and cylindrical — the ontology being easily extensible with other systems. Each coordinate system has properties that maps its specific characteristics (the coordinates and/or the angles necessary to uniquely identify a point in the space). Thus, in the example shown in Fig. 1 , the *CartesianSystem* has three length–based properties (corresponding to the x– , y– and z– coordinates), while the *CylindricalSystem* has two metric–like properties and a degree–like property that correspond to the radial, vertical and azimuth values, respectively.

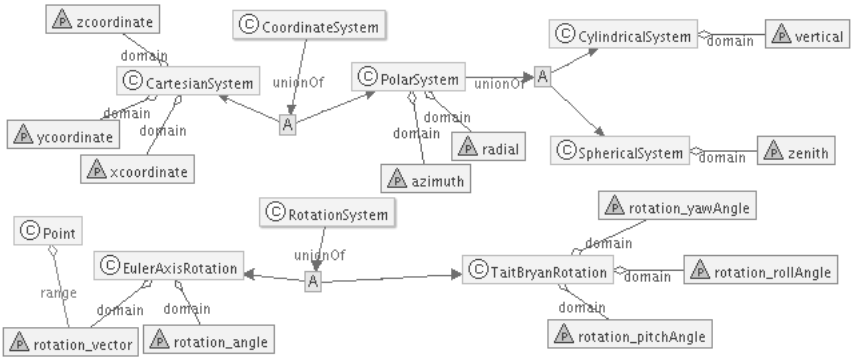


Fig. 1. Excerpt from the Coordinate Systems and Transformation Systems Ontologies

2.2 The Transformation Systems Ontology

The same approach has been used for the transformations ontology. As an illustration, each instantiable rotation system has predefined attributes (e.g. roll angle, vector, etc.) that match their corresponding mathematical elements. For example, the *EulerAxisRotation* defines properties for rotation vector and angle, while the *TaitBryanRotation* has a degree–like property for each dimension, as can be seen in Fig. 1 .

Both coordinate systems and transformation systems ontologies have been designed bottom–up [8], the superclasses being constructed as the union of their subclasses. By this approach, we force the notion of abstract classes (we can not have instances of *CoordinateSystem* , it has to be an instance of *CartesianSystem* or *SphericSystem*) without losing the *typeof* relation and the inheritance mechanism between concepts. Furthermore, the cardinality constraints defined on the properties (such as radial, coordinates, etc.) make those properties mandatory.

2.3 The Geometrical Shapes Ontology

Inspired by [9], the shapes ontology is the most complex one and it formalises the fundamental geometrical shapes such as cuboids, sphere, etc. The central concept of this ontology is the *SpatialObjet* , all basic shapes as well as any user–defined spatial object being subclasses or instances of the *SpatialObjet* concept. In the spatial ontology, each shape is described mainly by a transformation (e.g. *rotatedBy*), a position (*hasPosition*) which actually is seen as the central position of the geometrical form and by its dimensions (*hasDimensions*) or it can be identified by one or more points of reference (*definedBy*). As can be seen in Fig. 2 , the geometrical shapes are described as *Basic3DShape* which are of type *SpatialObject* , meaning we could easily extend the ontology to other multi–dimensional objects. Its dependency with the previously described ontologies gives it more flexibility in the positioning and transformation of the spatial

shapes. The topological and compositional properties defined on *SpatialObjet*s let the user to construct iteratively more complex *SpatialObjet*s, as described in the next section.

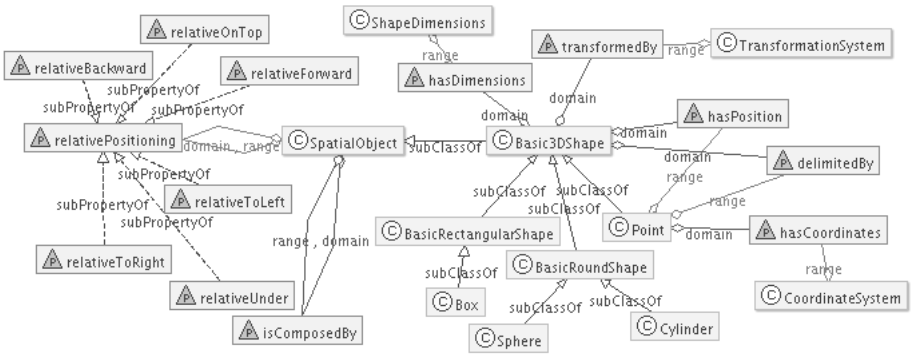


Fig. 2. Excerpt from the Geometrical Shapes Ontology

For all of the upper ontologies, the system considers that two parameters are implicit: the distance unit expressed in meters and the degree unit in radians. For more flexibility, the system could easily be extended by other ontologies that describe the distance and the degree systems.

3 Evolving the User Ontology

When the user starts working with the initial system he will find all concepts described in the reference ontologies. This means he can essentially use *Basic3dShape* and its associated basic operations to define his queries, that correspond to his basic objects. By composing these simple *Basic3dShape* objects, the user can describe new, more complicated shapes.

These new spatial objects have to be defined in the ontology, more precisely in the part reserved for user definitions. To do this he can use the Ontology Web Language[2] (OWL) or the tools provided by the system. A user can define new 3d objects or redefine existing objects (e.g. by changing the coordinate system). We will illustrate now an example of how a user might proceed to define his own objects and make them available as an ontological definition.

Let's imagine for this example that we would like to find Corinthian columns in the Pantheon data. More precisely we want to analyse those present in the entrance of the Pantheon that we can see in Fig. 3 .

By looking at the image of the entrance a user may try to retrieve the points defining a column using the basic definition of a *box*, another user may prefer to

[2] OWL Reference – W3C Recommendation (http://www.w3.org/TR/owl–ref/)

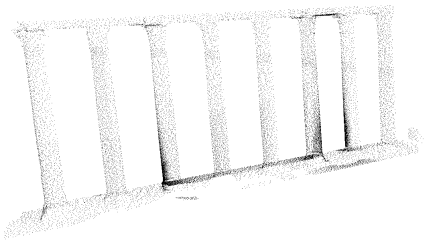


Fig. 3. The entrance of the Pantheon in Rome

retrieve the same points using a *cylinder*, whereas a third user may realise that a combination of the two previous approaches might be more appropriate. Let's say he would use a box for the base element, then a cylinder for the middle part of the column, and another box for the top of the column, all of them being combined to define a Corinthian column. For these types of complex shapes, a new concept can be added to the ontology, named *CorinthianColumn* . Furthermore, another concept *CorinthianEntrance* can be defined as composed of *CorinthianColumn* s. The *Basic3dShape* s used to define the new concepts have precise coordinates and ontological descriptions (see Fig. 4).

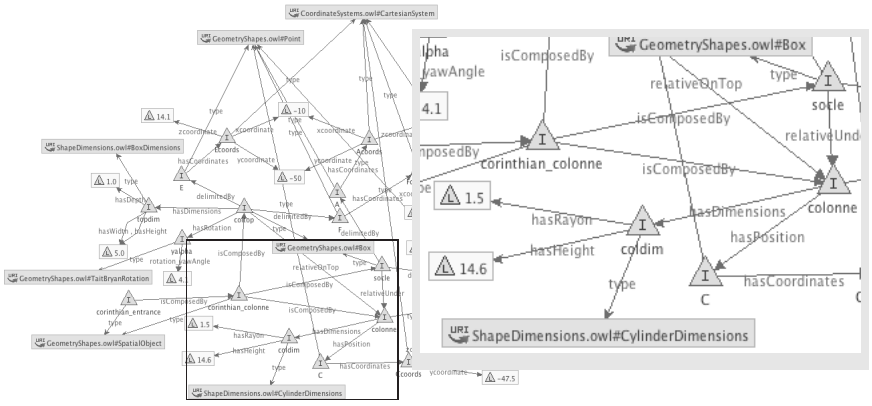


Fig. 4. A user–defined ontological description of a complex spatial object representing a Corinthian column

Based on the extended ontology another user could add his own concepts and make them dependent on the newly introduced concepts of the *CorinthianColumn* . As known from the history of architecture, Corinthian columns might consist of identical base and middle element, but they could differ in their top element. The

ontological definitions should also allow this refinement of the basic definitions of a *CorinthianColumn* .

4 Conclusion

In this paper we have presented an overview of a system combining the latest technologies in 3d data management with the newest developments in semantic data management. The main contribution consists in adding a dynamically created semantic layer to a cloud of points, starting from a reference ontology that describes the basics of the spatial aspects. Each ontology concept is linked with the spatial database system (SDS) through a set of standard queries. The result of the queries are groups of 3d objects delimited in spatial environnement which describe simple or complex ontology concepts. Driven by human interpretation we are able to create a representation of some parts of the pantheon architectural concepts. This idea offers a new way to add dynamic knowledge to a spatial database system and opens the door to semantic–based spatial data mining.

References

1. Bentley, J.L.: Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM **18**(9) (1975) 509–517
2. Nievergelt, J., Hinterberger, H., Sevcik, K.C.: The Grid File: An Adaptable, Symmetric Multikey File Structure. ACM Transactions on Database Systems **9**(1) (1984) 38–71
3. Seeger, B., Kriegel, H.: The Buddy-Tree: An Efficient and Robust Access Method for Spatial Data Base Systems. In: Proceedings of 16th International Conference on Very Large Data Bases, Brisbane, Australia (1990) 590–601
4. Berchtold, S., Keim, D.A., Kriegel, H.P.: The X-tree: An Index Structure for High-Dimensional Data. In: VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1996) 28–39
5. Künzi, C.: Spatial Indexing: Measure of Performances. Technical report, Information Management Institute, Neuchâtel, Switzerland (February 2008)
6. Guttman, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. SIGMOD '84: Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (1984) 47–57
7. Grenon, P., Smith, B.: SNAP and SPAN: Towards Dynamic Spatial Ontology. Spatial Cognition and Computation **4**(1) (2004) 69–104
8. Ciorascu, C., Künzi, C., Stoffel, K.: Designing Multi-Dimensional Spatial Ontologies for Spatial Data Mining. Technical report, Information Management Institute, Neuchâtel, Switzerland (October 2008)
9. Perry, M., Hakimpour, F., Sheth, A.: Analyzing Theme, Space and Time: An Ontology-based Approach. In: 14th Annual ACM International Symposium on Advances in Geographic Information Systems, ACM Press New York, NY, USA (2006) 147–154

Spatialized tags for building 3D shapes folksonomies

Laurent Moccozet

Information Systems Department, University of Geneva, Switzerland
Laurent.Moccozet@unige.ch

Abstract. In this paper we propose to define spatialized tags based on 3d hierarchical graphs in order to develop specialized taxonomies and folksonomies for 3D shapes. We aim at providing a framework that keeps the simplicity of use of tags and folksonomy for basic users and at the same capture the specific and intrinsic structure of 3D shapes. A prototype navigation system is proposed to demonstrate how the resulting folksonomy can be used to browse a 3D shape repository.

1 Introduction

In [1], Attene et al. propose a system to perform complex segmentations of 3D surface meshes and to annotate the detected parts through concepts expressed by an ontology. With the expansion of online 3D shapes repositories [2–6], we share the same analysis, regarding the needs and requirements for semantically characterizing 3D shapes for indexation [7–9]. The authors compare the two main strategies for annotation: keyword and ontology-based annotation. Keyword based annotations are currently widely used in Web 2.0 services for annotating online contents with folksonomies [10]. Attene et al. favor the second approach, which supports the viewpoint of domain experts. We do not aim at discussing and comparing both approaches. Their respective advantages and drawbacks are well-known and have been extensively discussed and compared. Tags and folksonomy based approaches have been particularly analyzed such as summarized in [11]. Our motivation here is simply to note the success gained by folksonomies and to propose a specialized approach for building and managing folksonomies for 3D shapes. Following Flickr[12], Del.icio.us[13] or Technorati[14] many Web 2.0 user-centered services are now providing folksonomy facilities in order to index and classify content. Content Management Systems (CMS), such as blogs (Wordpress[15], Dotclear[16]) or collaborative platforms (Drupal[17], Elgg[18]) provide features for annotating and browsing contents with tags and folksonomies. This situation demonstrates that although folksonomies provide a limited level of semantic, they have been widely adopted as a way to annotate, browse and navigate online contents. However, keyword and ontology based annotation systems are not antagonist. There are many existing and ongoing research efforts to connect and integrate both approaches such as proposed in [19, 20]. Therefore it seems natural that both approaches will finally collaborate and that they will benefit from each other.

Still recently, producing 3D content was limited to expert users because 3D authoring tools were requiring too many extra skills in 3D surface and shape geometry. With the advent of 3D authoring tools for large amateur audiences such as Sketchup [21], or users-created 3D online virtual environments such as Second Life [22], users are nowadays able and encouraged to produce directly their own 3D contents and publish them on the Web. We can reasonably expect that the number of available 3D shapes will soon explode. In parallel, we need to propose non-expert tools for appropriately annotating 3D contents directly by their creators. These tools should also take into consideration that the contents are produced by amateurs. Basic content creators usually produce 3D shapes that are pretty good for immediate display but that are of low to average geometric quality. The segmentation and shape analysis tools required for 3D annotation still require extra processing (for cleaning-up the shapes) and extra skills that are not immediately accessible to basic users and may prevent them from annotating their results when uploading.

Our approach is driven by two assumptions: the first one is that the success of folksonomies is coming from their simplicity for the users: no over efforts are required to understand how the annotation system works, no excessive extra efforts are required to annotate the contents, no extra efforts are required to understand a predefined classification or taxonomy Basic users are selectively lazy: they are ready to spend time to produce the content, but not to annotate it once they publish it. This may restrict ontology based systems to experts due to the resulting complexity of the annotation task. Moreover, semantic searching (such as the semantic search engine of the AIM@SHAPE shape repository) requires specific skills in ontologies. Basic 3D content producers create imperfect 3D shapes (close to polygon soup) whereas shape analysis tools for annotation expect specific geometric properties to process correctly. Our second assumption is that the 3D shape folksonomy model must be able to capture and express the intrinsic structure of 3D shapes such as raised in [1]: to achieve shape characterization, "the structural subdivision of an object into subparts, or segments, has proven to be a key issue. At a cognitive level, in fact, the comprehension of an object is often achieved by understanding its subparts". From our point of view, any framework for 3D shapes folksonomies must address these two aspects in order to be adopted by standard users and to achieve an accurate representation of 3D shapes semantic.

2 Spatialized Tags

Our purpose is to provide a simple way to express the main spatial relationships between the whole surface and its subparts. We therefore consider two usual relationships: hierarchy and connexion. The hierarchy association expresses the decomposition of parts into subparts: a part tagged with "body" label can be divided into subparts that will be tagged as "trunk", "legs", "arms" and so on. The connection association expresses the spatial connectivity between the features: a feature tagged "legs" will be spatially connected to another feature

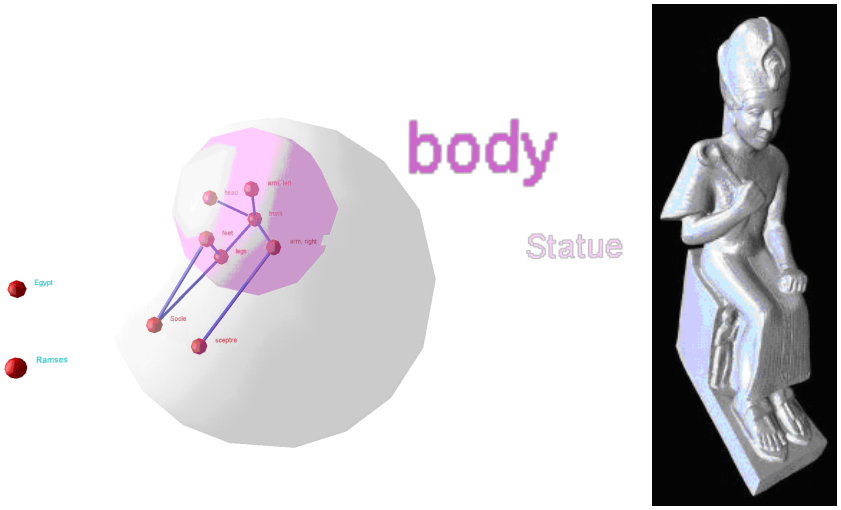


Fig. 1. tags graph associated to the Ramses statue (from the AIM@SHAPE shape repository [2])

tagged "feet" or "trunk". Such a representation can easily be conveyed with a hierarchical graph. As we want to semantically map the graph to the associated 3D shape, we adopt a 3D graph representation. WilmaScope [23] is a three dimensional interactive graph visualisation and viewing system. The Wilma design is quite versatile and makes it able to map out relationships between concepts, data, records, software entities, network nodes and many other contexts. A WilmaScope graph defines an object oriented model for a clustered graph with a recursive class definition [24]. WilmaScope also provides XML Graph (XWG) files that can be exported from or imported into the graph visualization system. Therefore we can interactively build the 3D graph with the visualization system, and export it as an XML file in order to process it for indexing and searching the 3D contents. For our purpose, each node of the graph corresponds to a semantically meaningful feature of the 3D shape to which one or more tags can be associated. Each edge of the graph represents a spatial link between the connected features. Finally, nodes can be grouped into clusters in order to reflect the hierarchical organization of complex 3D shapes and scenes (Fig. 1). The Ramses statue is tagged with 3 first level tags: Egypt, Ramses (both displayed as a red sphere) and Statue (displayed as a grey translucid sphere). Statue is associated to a cluster. This cluster is composed of three tags at the second level: scepter, socle and body. Body is itself defined as a cluster. In this cluster, the tag "head" is connected by an edge (displayed as a blue cylinder) with "trunk" as the two corresponding shape features are spatially connected. It is obvious by looking at the shape that it would be very difficult to segment the shape of the statue into arms, legs such as in [1]. Although the graph looks similar to a scenegraph [25], it has some major differences: scenegraphs are direct acyclic graphs whereas 3D

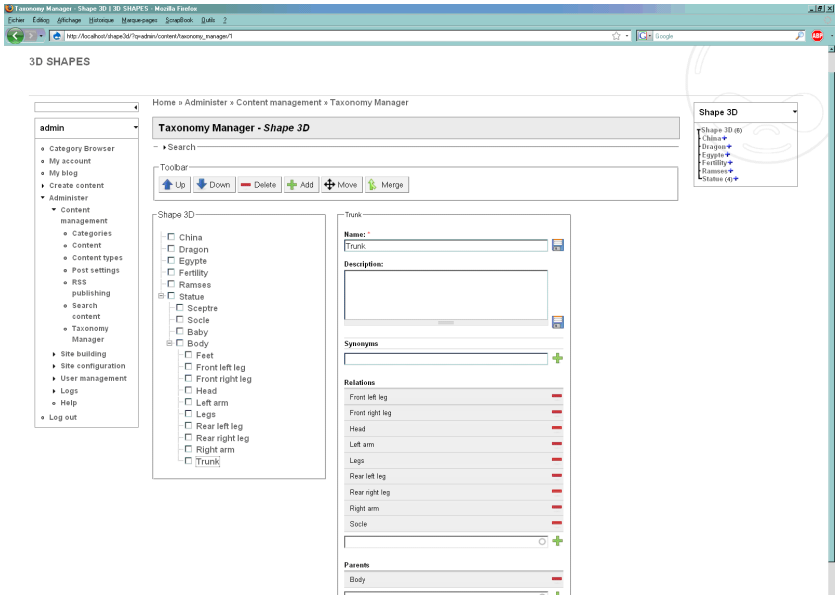


Fig. 2. Visualisation of the global vocabulary based on the 3D tags graphs of a few 3D shapes

tags graphs do not have such a restriction and scenegraphs are usually used to define compound objects by grouping multiple shapes whereas 3D tags graphs are used to annotate shape features. Our 3D tags graph can embed a semantic representation of scenegraphs.

We do not expect the 3D tags graph to be explicitly associated to specific parts of the annotated 3D shapes, although it could be extended for this purpose. In our framework, the 3D graph main goal is to provide an abstract representation of the 3D shape topology and structure. The main reasons are:

1. the annotation system must be kept as simple as possible in order to be adopted by users, which requires avoiding complex shape processing such as 3D segmentation;
2. it allows defining annotations that are not uniquely driven by geometric segmentation of the shape (some features may not be identified with shape segmentation) and that are not restricted to express spatial relationships.
3. it improves the reusability of existing 3D tags graphs, so that when a user is annotating a new shape, he/she can reuse a previously annotation for a similar shape by simply selecting it without having to explicitly recreate the spatial association between the new shape and the existing association. Giving the ability to reuse existing annotations is also an important issue to avoid over tagging.

There is however no doubt that providing explicit spatial links between tags and shape subparts would definitely improve the quality of the indexation. However,

from our point of view, it seems important that this should not be a requirement so that any user should be able to simply edit the 3D graph without having to explicitely segment and connect a 3D shape to the tags.

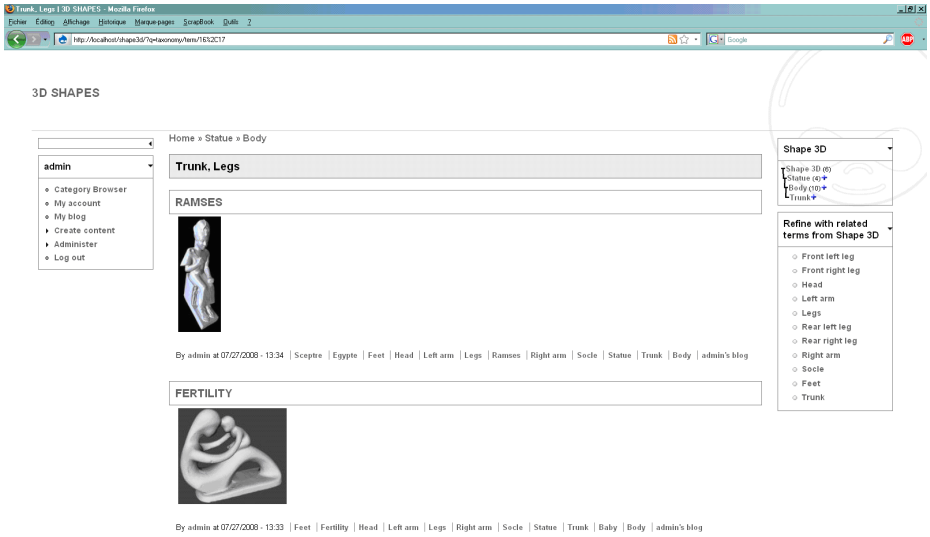


Fig. 3. Example of selection of 3D shapes labeled with "trunk" and "legs" related terms

For the user, the process is quite simple: when uploading a 3D shape, he/she just needs to freely describe the structure of the shape by creating the 3D hierarchical graph corresponding to the shape he/she is currently uploading. The resulting graph file is associated to the 3D shape as a metadata descriptor. Once the 3D graph is completed, it is added to the existing taxonomy by merging it to the global repository vocabulary. As we do not enforce tags to be explicitely attached to shape subparts, no extra processing of the 3D shape is required. Therefore, we maintain an appropriate extended approach of the taxonomy annotation process by keeping at the same the extra annotation work as direct as possible and by retaining the main aspects of the semantic of the shape structure.

3 Browsing 3D Shapes

We have implemented a prototype repository blog site for browsing 3D shapes using the proposed folksonomy model. It has been developed with the Drupal platform. The main reason is that this platform proposes the "Taxonomy" module, which defines vocabularies that can be organized and structured in such a way that it reproduces the structure of the 3D tags graph. According to [26], the Taxonomy module organizes taxonomies into vocabularies which consist of

one or more terms. The following principles apply to defining a vocabulary with the Taxonomy module:

- A vocabulary consists of a set of terms.
- Terms of a vocabulary can be ordered into hierarchies.
- A vocabulary may be free or controlled.
- A vocabulary allows defining synonyms and also related terms, similar to the "see also" in dictionaries.

In our implementation, graph nodes and clusters are represented by terms organized in a similar hierarchy and edges are represented by associating terms as "related terms" (Fig. 2). Following these simples translation rules, the XML graph description is converted and integrated into the vocabulary that is associated to the shape repository. To illustrate the principle, we have processed a few shapes from the AIM@SHAPE digital shape repository and inserted them into our prototype Drupal platform. In Fig. 2, we see how the 3D tag "trunk" is inserted in the vocabulary. It is hierarchically included inside a cluster tagged with "body", which is converted into a parent relationship in the vocabulary. It is spatially connected to various tags such as "legs" or "head", which is converted into a related relationship in the vocabulary.

Once the 3D graphs have been converted and merged into a hierarchical vocabulary, each blog entry corresponding to a 3D shape is tagged with all the associated terms. In addition to the traditional selection of contents with tags, it is then possible to provide various browsing strategies based on the terms hierarchy and terms relations. Logical expressions can be constructed by applying the AND and OR operators to the "hierarchy" and "related" relationships: we can select shapes that are tagged with "statue" and "trunk" and "head" with "trunk" related to "head". Such an expression will select statues with busts. Fig. 3 shows a simple example where the user can browse the terms hierarchy in conjunction with related terms. Whenever a term is selected, it is aggregated to previously selected ones with the AND operator. The top right panel displays the current vocabulary hierarchy and the top right one displays the terms that are related to the terms that have already been selected. In the center of the web browser window, the shapes matching the terms selection are displayed. The selection corresponds to shapes that are tagged with related terms "trunk" and "legs". The user can reach this selection by first selecting a first tag by browsing the tags hierarchy (and select "trunk"), then add a related term ("legs") from the "refine with related terms".

4 Discussion and Conclusion

The proposed framework still needs to be fully implemented and tested in an global interactive system for publishing, annotating and browsing 3D shapes online. However according to this preliminary prototype, we can expect it to provide an efficient framework for annotating 3D shapes that keeps the annotation process simple enough for easy adoption for standard user and that catches

the intrinsic structures of 3D shapes. In the future, we are expecting to propose a taxonomy query language that will integrate the hierarchy and related relationships for searching 3D shapes for more complex queries. The main issue is to evaluate how the folksonomy would evolve according to the scale of the repository. The introduction of the 3D tags graph naturally increases the risk of over tagging. However, when browsing a 3D shape repository it is obvious that they are organized into "families" that share the same global structures which is not the case with texts or photos. Therefore it should be possible to attenuate the over tagging effect by providing pre-defined 3D tags graphs for each shape family and let users select them for annotating new shapes.

5 Acknowledgements

The author would like to acknowledge the contibution of the various open source projects that have been used for this work and their authors, contributors and community: Wilma Scope, Drupal and the AIM@SHAPE digital shape repository.

References

1. Attene, M., Robbiano, F., Spagnuolo, M., Falcidieno, B.: Semantic annotation of 3d surface meshes based on feature characterization. In: International conference on semantic and digital media technologies (SAMT) 2007. (2007)
2. AIM@SHAPE: Aim@shape shapes repository `http://shapes.aim-at-shape.net/`.
3. Stanford Computer Graphics Laboratory: The stanford 3d scanning repository `http://graphics.stanford.edu/data/3Dscanrep/`.
4. Princeton Shape Retrieval and Analysis Group: Princeton 3d model search engine `http://shape.cs.princeton.edu/search.html`.
5. Reddy, M.: 3d models `http://lodbook.com/models/`.
6. AIM@SHAPE: Aim@shape digital shape workbench links `http://shapes.aim-at-shape.net/link.php`.
7. Falcidieno, B., Attene, M., Mortara, M.: The role of semantics in shape modelling and reasoning. In: Israel-Italy bi-national conf. on Shape Modeling and Reasoning for Industrial and Biomedical Applications. (2007)
8. Moccozet, L., Garcia-Rojas, A., Vexo, F., Thalmann, D., Magnenat-Thalmann, N.: In search for your own virtual individual. In: Semantic Multimedia, Springer Verlag (2006) 26–40
9. AIM@SHAPE: Shape annotator `http://shapeannotator.sourceforge.net/index.html`.
10. Quintarelli, E.: Folksonomies: power to the people. In: Incontro ISKO Italia - UniMIB. (2005) `http://www.iskoi.org/doc/folksonomies.htm`.
11. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata (2004) `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`.
12. Wikipedia Contributors: Flickr, Wikipedia, The Free Encyclopedia `http://en.wikipedia.org/wiki/Flickr`.

13. Wikipedia Contributors: Del.icio.us, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Del.icio.us.
14. Wikipedia Contributors: Technorati, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Technorati.
15. Wikipedia Contributors: Wordpress, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Wordpress.
16. Wikipedia Contributors: Dotclear, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Dotclear.
17. Wikipedia Contributors: Drupal, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Drupal.
18. Wikipedia Contributors: Elgg, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Elgg.
19. Specia, L., Motta, E.: Integrating folksonomies with the semantic web. In: European Semantic Web Conference (ESWC 2007). (2007) http://www.eswc2007.org/pdf/eswc07-specia.pdf.
20. Tanasescu, V., Streibel, O.: Extreme tagging: Emergent semantics through the tagging of tags. In: Workshop: International Workshop on Emergent Semantics and Ontology Evolution (ESOE 2007). (2007)
21. Wikipedia Contributors: Google sketchup, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Sketchup.
22. Wikipedia Contributors: Second life, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/SecondLife.
23. Wikipedia Contributors: Wilma scope, 3d graph visualization system, Wikipedia, The Free Encyclopedia http://wilma.sourceforge.net/.
24. Dwyer, T., Eckersley, P.: Wilmascope- an interactive 3d graph visualisation system. In: Graph Drawing, Springer Verlag (2006) 442–443
25. Wikipedia Contributors: Scene graph, Wikipedia, The Free Encyclopedia http://en.wikipedia.org/wiki/Scenegraph.
26. Drupal handbook contributors: Taxonomy: A way to organize content http://drupal.org/handbook/modules/taxonomy.

The Role of Contexts and Descriptors for Expressing Semantics in 3D Media

Francesco Robbiano, Michela Spagnuolo, and Bianca Falcidieno

CNR-IMATI-Genova, Via de Marini 6, 16149 Genova, Italy
{robbiano,spagnuolo,falcidieno}@ge.imati.cnr.it
http://www.ge.imati.cnr.it

Abstract. This paper deals with the enrichment of 3D media with semantic information. Currently one of the main goals is to provide clear, persistent and meaningful information for a user interacting with objects in different contexts. Tagging is a simple mechanism to attach explicit semantic information to digital objects, but there is no warranty that the attached information is still meaningful across different contexts. The state of the art presents several descriptor schemes that preserve some characteristics and discard other characteristics of the objects, depending on the viewpoints they cast. We show the setup of a system which will enable the analysis of different descriptor schemes and the extraction of their characteristics, ready to be matched with the contexts chosen by the user. An explicit conceptualization of the contexts on one hand can ease a searching mechanism in which the context is part of the query, and on the other hand can enable an *a priori* evaluation of the most suitable descriptor schemes to be used. Moreover, given a precise context, some quantitative (subsemantic) measurements on the objects can be properly translated into explicit semantics.

Keywords: 3D, annotation, description, semantics, context, Computer Graphics

1 Semantics and 3D Media

Enriching 3D media with semantics is a recently pursued aim in Computer Graphics. Semantics is intended as the association of a resource to a meaning, but as new research goals are set, the targeted meaning evolves accordingly. At an early stage, since the late seventies, the focus hinged on the representation schemes, and the semantics was intended as the connection between a syntactically valid representation to a semantically sound mathematical model of a solid. The addressed semantics did not go beyond geometry [1]. Now, in Computer Graphics the geometric representation schemes are quite well-established, and so this looks like a solved problem. Later, since the late eighties, especially in the CAD domain, the necessity of detecting geometric patterns and reusing them for flexible design tasks was felt. Another semantic layer was addressed, where the aim was to provide fixed taxonomies of features (feature definition), based on specific meanings and purposes (e.g. machining), and to match geometric patterns in the digital models with them (feature extraction) [2].

These are still open research issues, but are limited to a precise set of applications (e.g. CAD, CAM). Recently, since more and more 3D repositories are available (e.g. [3, 4, 5]) and the actual interaction of users with 3D media is increasing, it is important to characterise objects by properties perceivable by humans. *To represent* literally means "to present again", and therefore if the aim is to refer to objects as they were perceived by the human users, their representation should be enriched with high-level information, such as the linguistic category of the considered object, along with some of its main perceptual features (e.g. "ball, large", "table, elongated, low, with four legs"). This effort would ease a lot the tasks of search, retrieval and classification in large 3D repositories, and would also put the bases for a conceptual characterization of objects. An important step towards semantics has been performed by the Network of Excellence AIM@SHAPE [6], which in the span of time from 2003 to 2007 investigated the relationships between geometry, structure and semantics in 3D shapes, identifying different levels of expressiveness and conceptualizing some of the semantic characterisations typical of 3D digital models.

When the devised target is enriching 3D objects with language-based attributes, a simple tagging mechanism (e.g. stitching the tags "little" and "dog" to a digital model of a little dog) could look as the best choice, and often it is adopted, also for other widespread multimedia resources such as images [7], but it is neither simple to automate nor necessarily powerful. The aim is to equip 3D objects with conceptual information, either directly (*explicit semantic*) or indirectly (*implicit semantic*), and requirements at this stage are *clarity* and *persistence*. Clarity, because users are involved, and the added information must be clear, usable and shareable among them. Persistence, because the information has to be carried along with the digital model of the object, and so it must be suitable for any application involving the object itself. Tagging, as previously stated, is a very straightforward way to enrich a resource with semantics. It could be possible to explicitly tag the virtual little dog as "light", and this would fit the clarity requirement, as it would be immediately perceived as semantic information. But what about persistency? In a context in which we are considering objects to be put inside a shopping bag, we would tag any dog as "heavy", and so the semantic would simply not be persistent, i.e. not preserved across different contexts. Thus, it is important to integrate any direct, explicit semantic characterisation with implicit attributes that we will call "sub-semantic", because they do no have a precise meaning *per se*, but can be translated into explicit semantic when the context is ready to catch the carried meaning. For instance suppose that we have a measure of the roundness of a 3D object. The measure itself is persistent, even if the semantic characterisation that can be pulled out of it can be different in different contexts. The measure "0.9" could be interpreted as "round object" if the target is sorting different fruits and we know that "round" is something that can distinguish an orange from a banana, but can be interpreted as "not round" if the target is to understand if some ping-pong balls in a given set are irregular. These sub-semantic characterisations are persistent, but need to be interpreted through an elaboration layer in order to be meaningful and reduced to a proper semantic characterisation.

An example of this kind of characterisations is given by the so-called geometric segmentmeters [8], i.e. measurements that can be extracted from the geometric repre-

sentation of the objects (e.g. volume, area, volume of bounding box, length, height, width).

2 Contexts and Description Schemes

It is possible to assume that any user approaching a 3D object, both in real life and in digital applications, is intrinsically bound to the context in which he lives. If he has to fight monsters he would examine an object to decide if it can be thrown or used as a weapon, if he has to hold liquids he will care much about the presence of holes in the object, and if the important will be the resemblance of the object with a given model, the overall shape would be relevant. Even when the objects and the users are the same, the context may change, and the semantics that we would like to use for enriching the representation of the 3D media would change accordingly.

Therefore it is almost immediate to understand that no single way of encoding semantics is appropriate, unless we are bound to a unique and fixed context. It is possible, though, to exploit the expressive power of several descriptions on top of the same object.

Describing literally means "writing about", i.e. representing something just with the help of words, or in a wider sense, with the help of a different language. A shift of language is intended, therefore we are no more bound to presenting the target of the description exactly as it was originally (as in the re-presentation) but a layer of elaboration is allowed. In synthesis, a description is a looser form of representation in which a shift of language and an elaboration layer are allowed. Different layers of elaboration give the freedom of dealing with the object in different contexts.

In Computer Graphics several kinds of tools concur in providing descriptions of 3D objects, in the sense expressed above. Segmentation tools subdivide the object (usually its surface) into smaller parts, producing a structural characterization and a collection of subparts, each of which can be described in further details. Annotation tools are used to tag the objects, or their subparts [8], through conceptual tags, and possibly properties and relations with other resources. A particular kind of annotation is classification. Classification tools are meant to fit objects in one and only one conceptual class, which is useful when the aim is to discriminate among a fixed, well-defined and exhaustive set of categories. Very often, these processes are based not directly on the objects' representations but on signatures built on top of them. The signature can take into account global or local measurements, statistics for these measurements throughout the whole object (feature distribution), graphs representing their structure, 2D projections recording the view of the objects from different directions, and so on. For a complete survey on the object descriptors, and their implication in the retrieval task, refer to [9]. Each signature follows a precise description scheme, designed to capture specific characteristics from a 3D object. Different description schemes cast different viewpoints on the objects, being sensitive to some characteristics and invariant to others, therefore it is impossible to state that one descriptor scheme is better than another, unless a context is provided. Thus, far from considering the different techniques as competitors in a race, it could be possible to exploit the

distinctive feature of each of them to use only the one(s) which are most suitable in a given situation. The target is to match contexts with the proper description schemes, and there are two main approaches to achieve this goal.

The first is a sort of "black box" approach: several description schemes are applied, their effectiveness for retrieval is evaluated in any interesting context. From the outcome of the evaluation some conclusions can be drawn about the suitability, so that the matching between descriptors and contexts can be performed *a posteriori*. Nevertheless, in this case the knowledge about how the description schemes work is overlooked. The second approach can be regarded to as a "white box" approach, because the characteristics of the descriptors are used *a priori* to be matched against the characteristics of the contexts.

3 The Proposed System

The following is a preliminary description of the setup of a system that will support the exploitation of multiple descriptions and the selection of the most suitable in a given context. This system will serve also as the basis of the query formulation and the matching phases in a search engine for 3D objects. Our system will follow the second approach proposed above. In more details, the goals of the system are to characterise the descriptors, to characterise the contexts, and to provide the information necessary to match them. A framework allowing to run different description methods on several datasets is under construction. Experiments and analyses on the descriptions methods and their outcomes will be performed in order to have a clear characterisation of them. These characterisations have to be encoded in formalized conceptualizations, in such a way that they can be matched against the contexts. The conceptualizations of the descriptor schemes have to keep information about which characteristics are preserved and which are discarded in the proposed approach. Accordingly, the conceptualizations of the contexts will encode their desiderata about the characteristics that are important in a context and the ones that are not. Some examples of these characteristics are: overall shape, structure, pose, orientation, volume, presence of holes.

Characteristics	Length	Overall shape	Structure	Pose	Orientation
Contexts					
Short and long pencils	relevant	not relevant	not relevant	not relevant	not relevant
Detection of humans	not relevant	not relevant	relevant	not relevant	relevant
Resemblance of generic objects	not relevant	relevant	not relevant	relevant	not relevant
Descriptors					
Length of 1st Principal Component	preserved	discarded	discarded	discarded	discarded
Reeb Graph – Geodesic function	discarded	discarded	preserved	discarded	discarded
Shape Distribution	discarded	preserved	discarded	preserved	discarded

Table 1: An example of matching between contexts and descriptors, considering some relevant high-level characteristics. Descriptors that match very well some contexts may be highly inadequate in other contexts.

When descriptors and contexts are expressed in the same way, as shown in Table 1, it will be simpler to match them and select most suitable description scheme, or even to combine a number of them to fulfil the requirements of a single context. Clearly, our approach can be fruitfully combined with the *a posteriori* approach, as the actual performance of the description methods has to be tested and evaluated over shared benchmarks[10].

Acknowledgments. This work is supported by the FIRB project SHALOM (International Collaboration Italy-Israel, Contract N. RBIN04HWR8).

References

1. Requicha, A. (1980), Representation of rigid solids: theory, methods and systems, Computing Surveys. vol.12,
2. Shah, J. J. and Mantyla, M. (1995) Parametric and Feature Based Cad/Cam: Concepts, Techniques, and Applications. 1st. John Wiley & Sons, Inc.
3. The AIM@SHAPE Shape Repository, http://shapes.aimatshape.net
4. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jakobs,D (2003). A Search Engine for 3D Models. ACM Transactions on Graphics, Vol.22, 1, (pp.83-105)
5. CCCC – Dejan Vranic's 3D Search Engine, http://merkur01.inf.uni-konstanz.de/CCCC/
6. EC-FP6 IST NoE 506766 AIM@SHAPE Network of Excellence, http://www.aimatshape.net
7. FLICKR, http://www.flickr.com/
8. Attene, M., Robbiano, F., Spagnuolo, M., Falcidieno, B. (2007). Semantic Annotation of 3D Surface Meshes based on Feature Characterization. In Proc. of Semantics And digital Media Technology 2007, Genova, Italy.
9. Tangelder, J.W.H., Veltkamp, R.C. (2008). A survey of content based 3D shape retrieval methods. Multimedia Tools and Application , vol. 39 (pp. 441-471).
10. Veltkamp, R. C.; Ter Haar, F. B., (2008) SHape REtrieval Contest (SHREC) 2008, In Proceedings of Shape Modeling and Applications (SMI 2008), pp.215-216,