Content-Based 3D Object Retrieval

Benjamin Bustos
Center for Web Research, University of Chile, Chile

Daniel Keim and Dietmar Saupe
University Konstanz, Germany

Tobias Schreck
Technische Universitaet Darmstadt, Germany

Methods for automatically extracting descriptors from 3D objects are key to searching and indexing techniques in their growing repositories. The authors present two recently proposed approaches and discuss methods for benchmarking the 3D retrieval systems' qualitative performance.

Three-dimensional objects are an important multimedia data type with many application possibilities. For example, 3D models can present complex information, and content-based searching problems in large 3D object repositories arise in many practical fields. Example application domains include CAD/CAM, VR, medicine, molecular biology, military applications, and entertainment.

In content-based searching and organization, the problem is to define appropriate measures to automatically assess the similarity between any pair of 3D objects based on a suitable notion of similarity. The existence of such similarity measures is an important precondition for implementing effective search algorithms; it lets you query a repository of 3D objects for specific content and facilitates reusing 3D content. Also, similarity metrics let you organize 3D repositories by representing large object collections with limited cluster prototypes, or visualize the content of large databases by appropriate 2D mappings.

A similarity notion supports advanced automatic applications such as classifying shapes in industrial screening. For example, in medicine detecting similar organ deformations is useful for diagnostic purposes. Three-dimensional object databases also support CAD tools, which have many applications in industrial design and manufacturing—reusing standard parts can lead to reduced production costs.

Recently, researchers have proposed a range of methods for implementing similarity notions for 3D objects. In this article, we present a systematic overview of methods for characterizing 3D objects with descriptors suited for content-based 3D retrieval.

3D database retrieval concepts

A common characteristic of all applications in multimedia databases (and in particular for 3D object databases) is that a query searches for similar objects instead of performing an exact search, as in traditional relational databases. Multimedia objects cannot be meaningfully queried in the classical sense (exact search) because the probability is low that two multimedia objects are identical, unless they are digital copies from the same source. Instead, a query in a multimedia database system usually requests a number of objects most similar to a given query object or to a manually entered query specification. Therefore, one of the most important tasks in a multimedia retrieval system is to implement efficient similarity search algorithms.

Typically, multimedia data are modeled as objects in a metric or vector space, where a distance function must be defined to compute the similarity between two objects. Thus, the similarity search problem is reduced to a search for close objects in the metric or vector space. Two common similarity queries are the *range query* (which returns all the objects within some given distance ε to the query) and the *k nearest neighbors query* (which returns the k closest objects to the query).

The primary goal in 3D similarity search is to design algorithms that can effectively and efficiently execute similarity queries in 3D databases. Effectiveness is the ability to retrieve similar 3D objects while holding back nonsimilar ones. Efficiency is the cost of the search measured in CPU or I/O time. But, first you need to define how the similarity between 3D objects is computed. For this, the most widely used approach up until now has been to compare the global geometric similarity between two 3D objects.

One way to compute global geometric similarity is by direct geometric matching, measuring how costly it is to transform a given 3D object into another. The cost associated with the transformation process serves as the metric for similarity. However, directly comparing all 3D objects from a database with a query object may be a prohibitively time-consuming process because 3D objects can be represented in many different formats and might exhibit widely varying complexity.

The descriptor-based approach is another way to compute the similarity between 3D objects. With this approach, a retrieval system extracts numerical descriptors (also known as feature vectors) from the 3D objects

 Published by the IEEE Computer Society 0272-1716/07/$25.00

and uses them for indexing and retrieval purposes. A 3D feature vector usually characterizes a 3D object's global geometry. We can then compare this vector to other feature vectors to identify similar shapes and discard dissimilar ones (see the "Feature Vector Approach" sidebar for more details).

Apart from global geometric similarity, the notion of local or partial similarity might be important for some specific application domains. In this case, the problem is to find similarities in parts or sections of the 3D objects, or even to find complementary parts between solid object segments (as in protein docking). Although this is an important research field in 3D databases, it is still unclear how to design fast segmentation methods that lead to suited 3D object partitions, which could be compared pairwise. Another approach to define the similarity of 3D objects is based on comparing the 3D objects' topology, which can be done, for example, by comparing the skeletons derived from solid objects.

Content-based retrieval with descriptors

Candidate features for 3D description depend on the specific format in which the models in a considered database are given. A property explicitly coded in most representations is geometry, and consequentially, 3D descriptors usually rely only on geometry information. As Figure 1 illustrates, extracting shape descriptors is a multistage process,[1] In this process, a retrieval system first preprocesses a 3D object to achieve required invariance and robustness properties. Then it transforms the object so that its character is either of surface type or volumetric, or one of several 2D images captures it. Then, a numerical analysis of the shape takes place; the descriptor is formed from this result.

Feature extraction model

In the preprocessing stage, the goal is to achieve approximate rotation, translation, and scale invariance, as well as robustness of the descriptor with respect to noise. Ideally, an arbitrary combination of translation, rotation, and scaling operations applied to one object should not affect its similarity measure to another object, even in the presence of moderately noisy perturbations of the models. Invariance in anisotropic scaling might also be desirable. In some applications, even certain allowable shape deformations, as in articulated bodies, should be considered as an invariance requirement for a shape descriptor. Besides relying on preprocessing to provide these invariances, designing descriptors that provide certain invariances by definition (that is, in the numerical transform stage of the descriptor generation) is an option.

The next stage abstracts the model to one out of several different key characteristics that we can view in the 3D shape. The model might be an infinitely thin surface with precisely defined properties of differentiability. Alternately, we might see it as as a thickened surface that occupies some portion of volume in 3D space, or as the boundary of a solid. The transformation of a mesh into one of the latter forms is typical for volumetric abstractions. A third way to capture the mesh's character is to project it onto one or more image planes producing renderings, corresponding depth maps, silhouettes, and so on, from which we can derive the descriptors. In the numerical transformation stage, one of various methods captures certain main features of the models in one of the three abstraction types. Voxel grids and image arrays

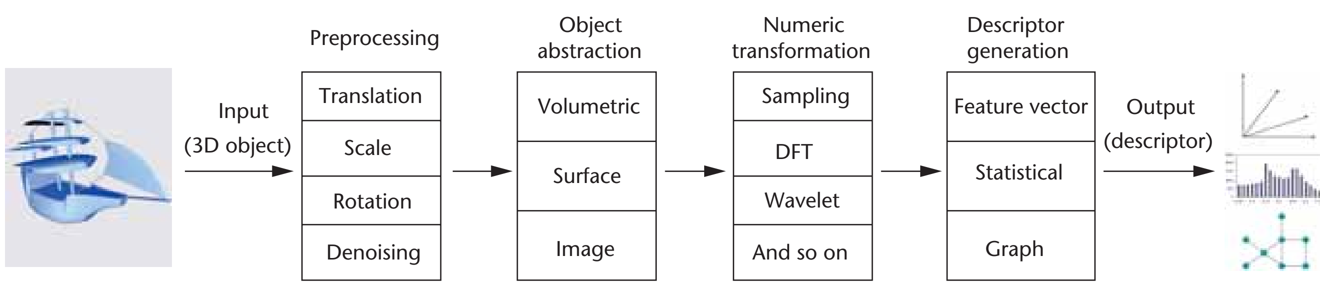
Feature Vector Approach

A *metric space* is a pair (X, δ) where X represents the universe of valid objects and $\delta : X \; X \rightarrow R^{+}$ is a function of object pairs, that returns non-negative real values (the distance between objects in the space) that hold the properties of a metric (strict positiveness, symmetry, and the triangle inequality). A vector space R^d is a particular type of metric space, composed by *d*-tuples of real numbers called vectors. That is, if $\in R^d$ then $x = (x_1, \ldots, x_d)$, $x_i \in R$, $1 \;\; i \;\; d$.

A widely used family of distance functions for vector spaces is the Minkowski distance (L_p), which we define as

$$L_p(x,y)=\left(\sum_{i=1}^{d}\left|x_i-y_i\right|^p\right)^{1/p}, p\geq 1$$

To model multimedia data as a vector space, we must use a transformation function, which depends on the multimedia data type. This function extracts important features from the multimedia objects and maps these values into *d*-dimensional feature vectors. Usually, the dimensionality *d* of the resulting feature vector is a parameter of the transformation function: By using higher values of *d* it is possible to obtain a better (finer) representation of the multimedia object. However, in practical applications there is usually a saturation point where adding more dimensions after reaching the saturation point does not considerably improve the quality of the object's representation. For most applications, the transformation is irreversible—that is, it is not possible to reconstruct the original multimedia object from its feature vector.



1 **A 3D feature vector extraction process model.**

Index Structures for Efficient Retrieval

A naive method to answer range and *k* nearest neighbors queries is to perform a sequential scan of the database, comparing each multimedia object directly against the query. However, this method might be too slow for real-world applications. An *index structure* can be used to filter out irrelevant objects during the similarity search without comparing them against the query, thus avoiding the sequential scan.

Researchers have proposed several index structures for metric and vector spaces. Metric access methods are index structures that use the metric properties of the distance function (especially the triangle inequality) to filter out the space's zones.[1] Spatial access methods are index structures especially designed for vector spaces which, together with the metric properties of the distance function, use geometric information to discard points from the space.[2] Usually, these indices are hierarchical data structures that use a balanced tree to index the database.

References

1. E. Chávez et. al, "Searching in Metric Spaces," *ACM Computing Surveys,* vol. 33, no. 3, 2001, pp. 273-321.
2. C. Böhm, S. Berchtold, and D. Keim, "Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases," *ACM Computing Surveys*, vol. 33, no. 3, 2001, pp. 322-373.

can be Fourier or Wavelet transformed, and we can adaptively sample surfaces. This yields a numerical representation of the underlying object, not necessarily allowing the object's reconstruction.

The final stage generates a descriptor. We can classify the descriptor type itself as feature vectors, statistical descriptors, or graph-based descriptors. The first two methods capture object features either in vectors of real or statistical summarizations, and defining distance functions for them is straightforward. Graph-based descriptions are more complex in nature, and are especially useful for representing structural properties when object features can be segmented in a meaningful and robust way. On the other hand, for graph-based representations, custom distance functions often have to be developed.

Other classifications for shape description and analysis methods are possible; see for example the surveys of Tangelder and Veltcamp[2] or Ramani et al.[3] The methods in the feature vector class are efficient, robust, easy to implement, and provide some of the most common and best approaches. This does not imply, however, that statistical or graph-based methods cannot be recommended; most of these methods have their particular strengths and may well be the ideal candidate for a specific application.

Desired properties of retrieval

An efficient and effective 3D search system has several desirable properties. Efficiency refers to the consumption of resources needed for storage and retrieval of the multimedia objects and is typically measured by system response times or storage utilization. Effectiveness typically relates to the quality of the answer objects that the search system returns, and is often assessed by metrics known from information retrieval. Quality of the answers measures the answers' relevance with respect to the query object. An effective retrieval system returns the most relevant objects from the database on the first positions of the *k* nearest neighbors query, and holds back irrelevant objects from this ranking.

A feature vector -based search system's effectiveness and efficiency are determined primarily by the implemented feature vectors. Regarding efficiency, we require that the system efficiently extracts the feature vector descriptors from the objects and efficiently encodes them, possibly by a representation that provides the embedded multiresolution property. Fast extraction makes it possible to perform database inserts on the fly, where the system calculates feature vectors for any new object to be inserted in real time. Efficiency of representations requires the vectors to consume minimal space in terms of number of vector components and number of bits used to encode the component values. Short feature vectors reduce the amount of required disk or memory space. They also speed up distance calculations and access to the vectors. Specifically, the performance of multidimensional index structures deteriorates quickly if the dimensionality of the indexed data grows.[4] Often there is a typical tradeoff between the feature vectors' resolution (size) and the provided discrimination power, in that higher dimensionality leads to better retrieval precision. Therefore, the embedded multiresolution property is desirable. Feature vectors with this property encode progressively more object information inside a given vector. So, by considering subsets of dimensions in embedded multiresolution feature vectors, we can choose the object description's level of detail. (For additional information on efficiency aspects, see the "Index Structures for Efficient Retrieval" sidebar).

Regarding system effectiveness, we prefer to have descriptors that provide sufficient discrimination power as well as certain invariance properties the application requires. Discrimination power requires that an appropriate distance function defined in the feature vector space effectively captures the similarity relationships present in object space by distances in the feature vector space. Also, the descriptors should be robust with respect to small changes in the input 3D objects. Depending on the application, certain invariances of the search might be desired, meaning that distances in the feature vector space should be invariant with respect to certain object transformations that leave the similarity relationships unchanged. Robustness is another requested effectiveness criterion, implying that small variations in the multimedia objects, caused by noise, should not dramatically alter the resulting distance between the objects in the feature vector space.

Image- and graph-based descriptors

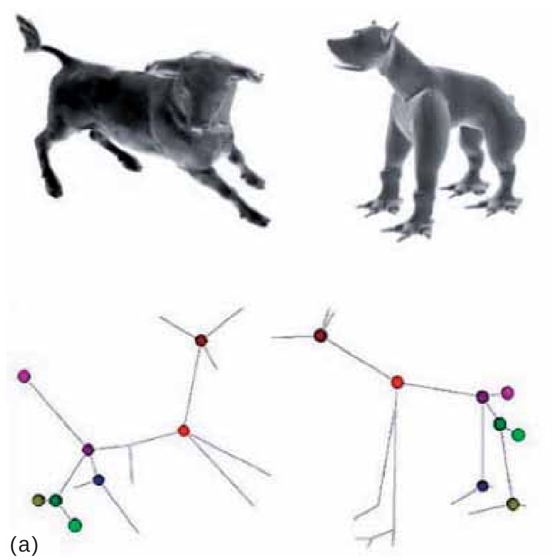
As recent surveys indicate, a wealth of different approaches to describe 3D shape for retrieval systems exist.[1-3] The situation is comparable to content-based image retrieval, where many different descriptors have been proposed over recent years. Many of the 3D descriptors in existence are heuristically introduced, motivated by techniques and practices from computer graphics (projection-based descriptors), geometry (descriptors based on surface curvature statistics), or signal processing (descriptors representing object samples in the frequency domain). Usually, it is unclear beforehand which of the potentially many different features should be preferred for addressing the general 3D retrieval problem. Each of the descriptors captures specific model information, and their suitability for effective retrieval must be experimentally evaluated. Two exemplary 3D descriptors have recently been proposed.[5,6] We chose these descriptors to give you a feeling for the types of approaches used for shape matching.

Skeletons derived from solid objects are intuitive object descriptions, possibly capturing important structural object information. For 3D object retrieval, we need to devise suitable skeletonization algorithms and similarity functions defined on skeletons. Sundar et al. use skeletons obtained by connecting clusters of object voxels left after an appropriate thinning of the model voxels has taken place.[5] The thinning method, based on the voxel grid's Euclidean distance transform, identifies salient object voxels**.** Clusters of salient voxels connect to form a skeleton graph, where information on the underlying voxel clusters as well as local topological properties of the skeleton enrich the graph nodes . Together with an intelligent graph-matching scheme, we can then calculate the dissimilarity between any two 3D models for which skeletons have been determined (see Figure 2a). Sundar et al. note the method's suitability for matching articulated objects as well as the potential for finding partial matches between objects.

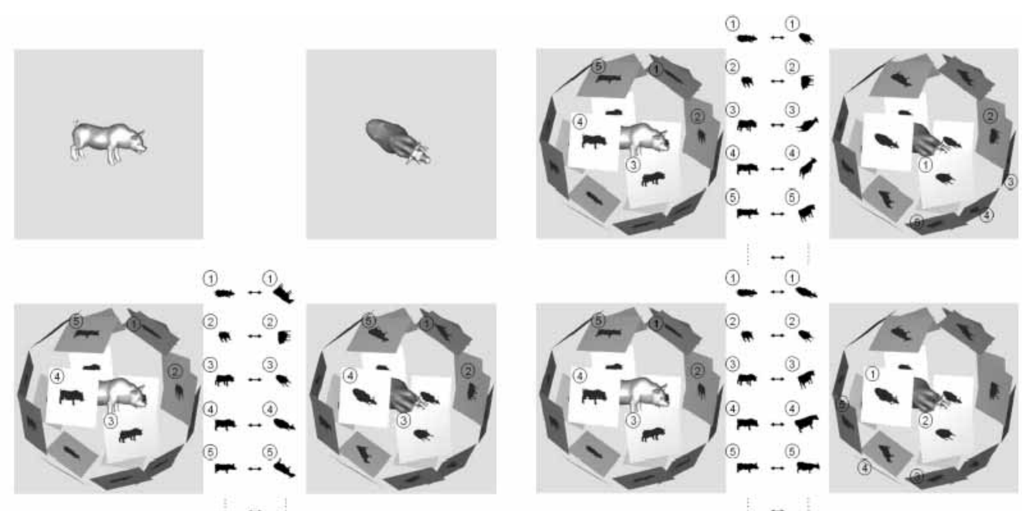
Chen et al. demonstrate how intelligent retrieval of 3D models can successfully leverage 2D shape description approaches.[6] The authors calculate the similarity between a pair of 3D models by comparing sets of 2D projections rendered from the model. To this end, they considered a system of virtual cameras distributed regularly on an imaginary sphere enclosing a 3D model. Each camera renders a 2D image of the model through parallel projection (see Figure 2b). Each projection is then described by image features extracted from the corresponding 2D silhouettes. The similarity between two objects is defined as the minimum of the sum of distances between all corresponding image pairs over the rotation of one camera system relative to the other. Together with an efficient multistage filtering approach that gives increasingly more detail information from the silhouette descriptors, the system supports retrieval in large 3D databases and provides implicit rotational invariance not requiring object orientation preprocessing. In benchmark-based precision-recall experiments, the system provided excellent retrieval performance.

Evaluating retrieval quality using benchmarks

To evaluate a search engine's retrieval quality, the information retrieval community has defined and proposed several measures. Two well-known effectiveness measures are *precision* and *recall*. Precision is the fraction of the retrieved objects that is relevant to a given query, and recall is the fraction of the relevant objects that the database retrieved. We can use precision val-



(a)



(b)

2 **(a) A pair of mutually best-matching objects from a 3D database, using graph-based shape description.[5] (b) The LightField descriptor determines similarity between 3D objects by the maximum similarity when aligning sets of 2D projections obtained from an array of cameras surrounding the object.[6]**

The SHREC 3D Retrieval Contest

Following examples in other retrieval disciplines, researchers in the 3D field have established an international shape retrieval contest. In 2006, chaired by Remco Veltkamp of the European Community-funded Network of Excellence Aim@Shape, the 3D Shape Retrieval Contest (SHREC) debuted at the IEEE International Conference on Shape Modeling and Applications. The initial contest was designed around the Princeton Shape Benchmark of 2004[1], and in 2007 specialized toward problems involving, for example, watertight models, CAD content, and partial similarity retrieval tasks.

SHREC is expected to become an objective forum for evaluating and comparing 3D retrieval algorithms, and to stimulate research on new, challenging aspects of 3D shape retrieval.

ues at several recall levels to produce precision versus recall plots. These plots let us easily compare the effectiveness of similarity search algorithms. Another widely used effectiveness measure is the *R-precision* (also called first-tier precision), which is defined as the precision for retrieving *N* objects, where *N* is equal to the number of relevant objects to the query stored in the database. The R-precision gives a single number to rate a retrieval algorithm's performance. This effectiveness measure is similar to the *bull eye percentage* (also called second-tier precision), defined as the recall for retrieving 2*N* objects from the database.

To compare different retrieval algorithms against each other using such evaluation measures requires benchmark databases with reference queries and associated relevance information. Among several 3D benchmarks proposed earlier, the well-known Princeton Shape Benchmark (PSB)[7] is one of the most popular such benchmarks to date. It consists of a carefully compiled collection of about 1,800 3D models harvested from the Internet. The models represent real-world objects such as vehicles, buildings, animals, or plants, and are classified according to function and shape on multiple levels of abstraction. Based on such benchmarks, experimental evaluation of 3D retrieval methods can take place. Bustos et al. provide a thorough experimental effectiveness evaluation of several different 3D descriptors.[8] This work showed that many of the proposed descriptors for 3D objects have good average effectiveness, and are well suited for general-purpose 3D content represented by the benchmarks. Also, the international 3D Shape Retrieval Contest (Shrec) launched in 2006, was initially built around the PSB benchmark. See "The Shrec 3D Retrieval Contest" sidebar for more information.

Future work

Many important open problems in the research area of content-based description and retrieval of 3D objects exist. For example, domain-specific model databases (such as CAD parts or models from visualization) may show specific requirements and restrictions that must be taken into account to perform the similarity query (for example, invariance with respect to local deformations in geometry and topology or invariance with respect to anisotropic scaling). Thus, the similarity model used to perform the search must reflect these additional constraints or requirements.

Most of the retrieval methods developed to date restrict themselves on geometric aspects of 3D models. Conceptually, we can associate additional important object attributes such as color, material properties, and texture with 3D models. Depending on how the model was created, we can also consider the structural object or machining process information. While these attributes offer additional information that we could exploit for content-based retrieval, the absence of a widely accepted, versatile, and powerful 3D representation format makes research into multi-aspect 3D retrieval difficult in practice. Havemann and Fellner provide a discussion of the format problem, along other pressing challenges in managing growing amounts of 3D object data.[9]

The definition and effective implementation of partial similarity search notions among multimedia objects remains a big challenge. This problem is far more complex than the global geometry similarity search problem, because in partial similarity only a fraction of the 3D object is considered for the match. Even the concept of "match" in this context must be properly defined—for example, we might want to look for similar, or complementary parts.

Acknowledgments

This work was partially funded by the German Research Foundation (DFG) under Projects No. KE 740/6-1/8-1 and SA 449/10-1 within the strategic research initiative on "Distributed Processing and Delivery of Digital Documents". It was also partially funded by the Millennium Nucleus Center for Web Research, Grant P04-067-F, Mideplan, Chile.

References

1. B. Bustos et al., "Feature-Based Similarity Search in 3D Object Databases," *ACM Computing Surveys,* vol. 37, no. 4, 2005, pp. 345-387.
2. J. Tangelder and R. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods," *Proc. Int'l Conf. Shape Modeling and Applications (*SMI 04*)*, IEEE CS Press, 2004, pp. 145-156.
3. N. Iyer et al., "Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends," *Computer-Aided Design*, vol. 37, no. 5, 2005, pp. 509-530.
4. C. Böhm, S. Berchtold, and D. Keim, "Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases," *ACM Computing Surveys*, vol. 33, no. 3, 2001, pp. 322-373.
5. H. Sundar et al., "Skeleton Based Shape Matching and Retrieval," *Proc. Shape Modeling International 2003* (SMI 03), IEEE CS Press, 2003, pp. 130-142.
6. D.-Y. Chen et al., "On Visual Similarity Based 3D Model Retrieval," *Computer Graphics Forum*, vol. 22, no. 3, 2003, pp. 223-232.
7. M. Kazhdan et al., "The Princeton Shape Benchmark," *Proc. Shape Modeling International 2004* (SMI 04), IEEE CS Press, 2004, pp. 167-178.

8. B. Bustos et al., "An Experimental Effectiveness Comparison of Methods for 3D Similarity Search," *Int'l Journ. Digital Libraries,* vol. 6, no. 1, 2006, pp. 39-54.
9. S. Havemann and D. Fellner, "Seven Research Challenges of Generalized 3d Documents," *IEEE Computer Graphics and Applications*, vol. 27, no. 3, 2007, pp. 70-76.

Benjamin Bustos *is an assistant professor in the Department of Computer Science at the University of Chile. He is also a researcher at the Millennium Nucleus Center for Web Research. His research interests are database index structures and multimedia information retrieval. He has a doctoral degree in natural sciences from the University of Konstanz, Germany. Contact him at bebustos@dcc.uchile.cl.*

Daniel Keim *is a professor and head of the Databases and Visualization group in the Department for Computer and Information Science at the University of Konstanz, Germany. He is also the coordinator of the recently established strategic research initiative on "Scalable Visual Analytics: Interactive Visual Analysis Systems for Complex Information Spaces," funded by the German Research Foundation (DFG). His research interests include data mining, databases, knowledge discovery, visual analytics, and information visualization. He has a doctoral degree from the University of Munich, Germany. Contact him at keim@informatik.uni-konstanz.de.*

Dietmar Saupe *is a professor of computer science and also chair of the PhD program "Explorative Analysis and Visualization of Large Informaion Spaces" at the University Konstanz. His research interests include multimedia signal processing, computer graphics, and visualization. He has a doctoral and habilitation degree in applied mathematics from the University Bremen. Contact him at dietmar.saupe@uni-konstanz.de.*

Tobias Schreck *is a postdoc researcher with the Interactive Graphics Systems Group in the Department for Computer Science at Technische Universitaet Darmstadt, Germany. His research interests are multimedia information retrieval and visual analytics. He has a doctoral degree in natural sciences from the University of Konstanz, Germany. Contact him at tobias.schreck@gris.informatik.tu-darmstadt.de.*