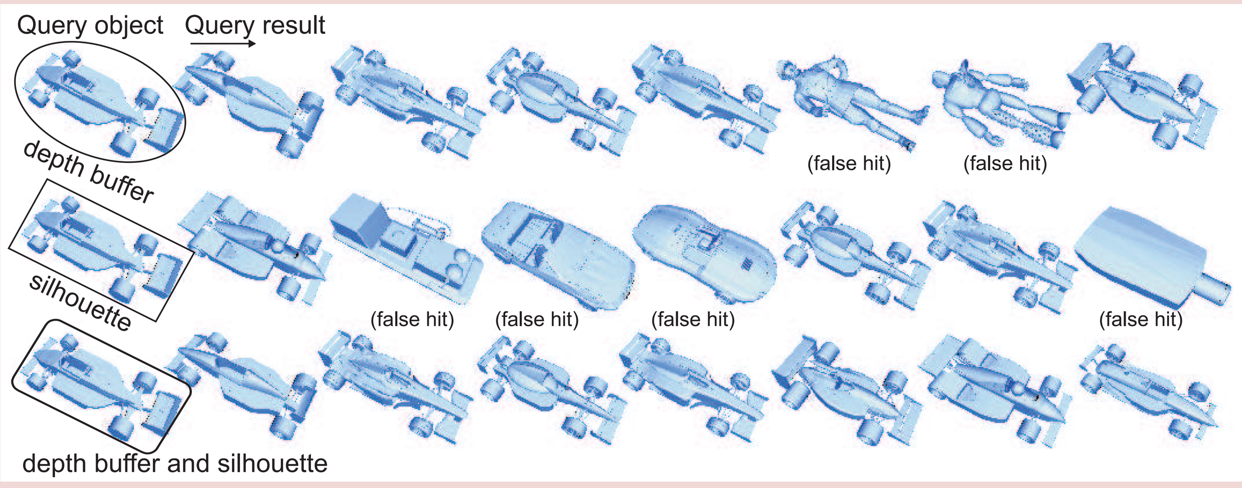
Dynamic Similarity Search in Multi-Metric Spaces

Benjamin Bustos
Department of Computer and Information Science
University of Konstanz, Germany
bustos@informatik.uni-konstanz.de

Tomáš Skopal
Department of Software Engineering
Charles University in Prague, Czech Republic
tomas.skopal@mff.cuni.cz

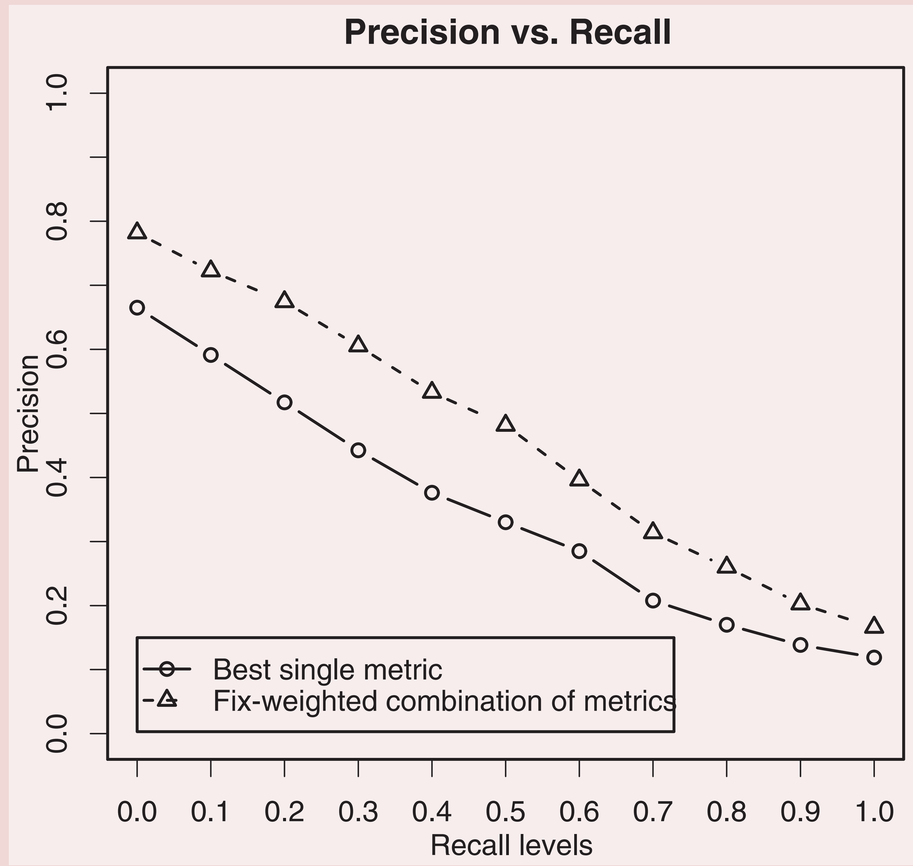
Multi-metric spaces



Example of a simple combination of two different metrics.
Each metric alone retrieves some relevant objects but also some **false hits**. By using a **combined metric**, the similarity search retrieves only relevant objects.

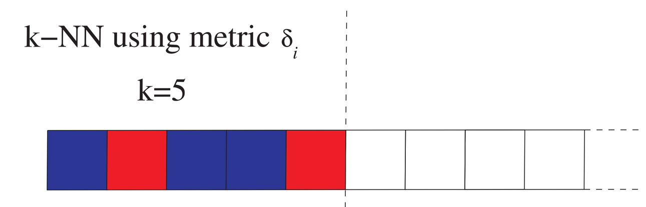
Better improvements in the effectiveness of the similarity search are achievable by using more metrics and different weights for each one (fixed weights with low (0), medium (1) or high (2) values). By testing all possible weight sets, it is possible to find the optimal one. The precision vs. recall diagram shows that the best **fix-weighted combination** improves considerably the effectiveness of the search compared with the best single metric [1].

However, static combinations have some disadvantages:
The best weight set is highly database-dependent, it is expensive to compute (testing all possibilities), and we observed that depending on the query object the best set of weights to use was different. Thus, we proposed a method to compute **dynamic weights** depending on the **query object** [2], which leads to the concept of multi-metric space.



Computing the weights

I. Perform k-NN in training dataset



Three objects belong to the blue class and two objects belong to the red class.

II. Entropy impurity

P_{ω_i} : fraction of objects that belong to model class i

$$entropy(\delta_i) = -\sum_{i=1}^{|\#classes|} \begin{cases} P_{\omega_i} \cdot \log_2(P_{\omega_i}) & \text{if } P_{\omega_i} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The entropy impurity of metric δ_i is equal to 0 if all objects belong to the same class, and has a maximum value ($\log(k)$) if each object belongs to a different class.

III. Weights

$$w_q(\delta_i) = \frac{\log_2(k) - entropy(\delta_i)}{\log_2(k)} \in [0,1]$$

The weight is 0 if the metric has maximum entropy impurity, and is 1 if the metric has entropy impurity equal to 0.

R-precision values

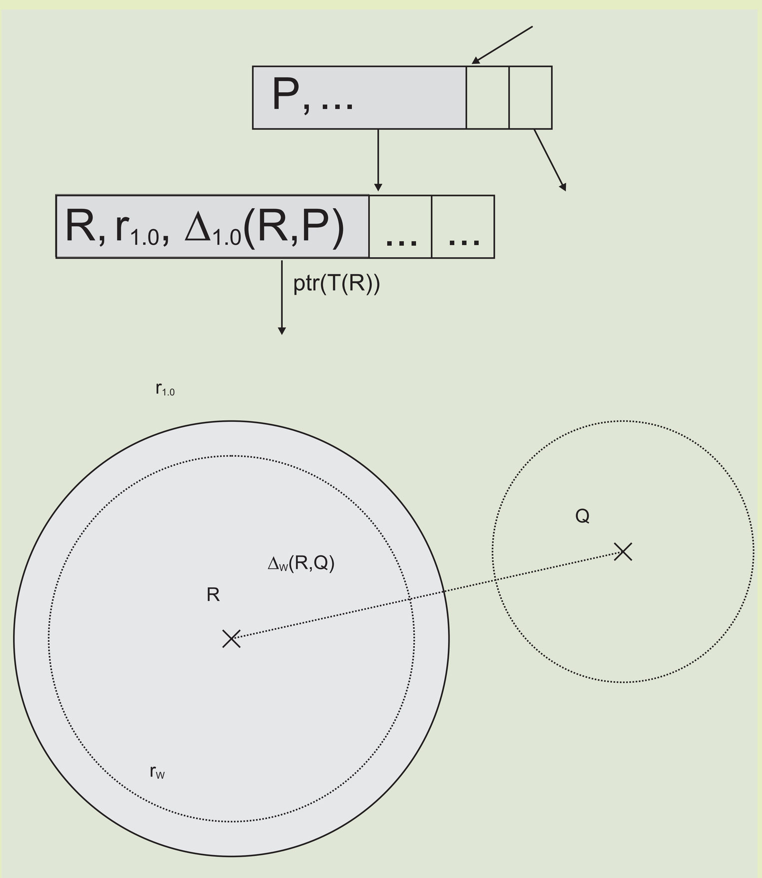
Method	Average R-prec.	Relative improv.
Best single metric	0.3220	0%
Best fix-weighted combination	0.4263	32%
Entropy impurity weighted combination (k=3)	0.4550	41%

M³-tree: index structure for fast retrieval

Linear multi-metric:

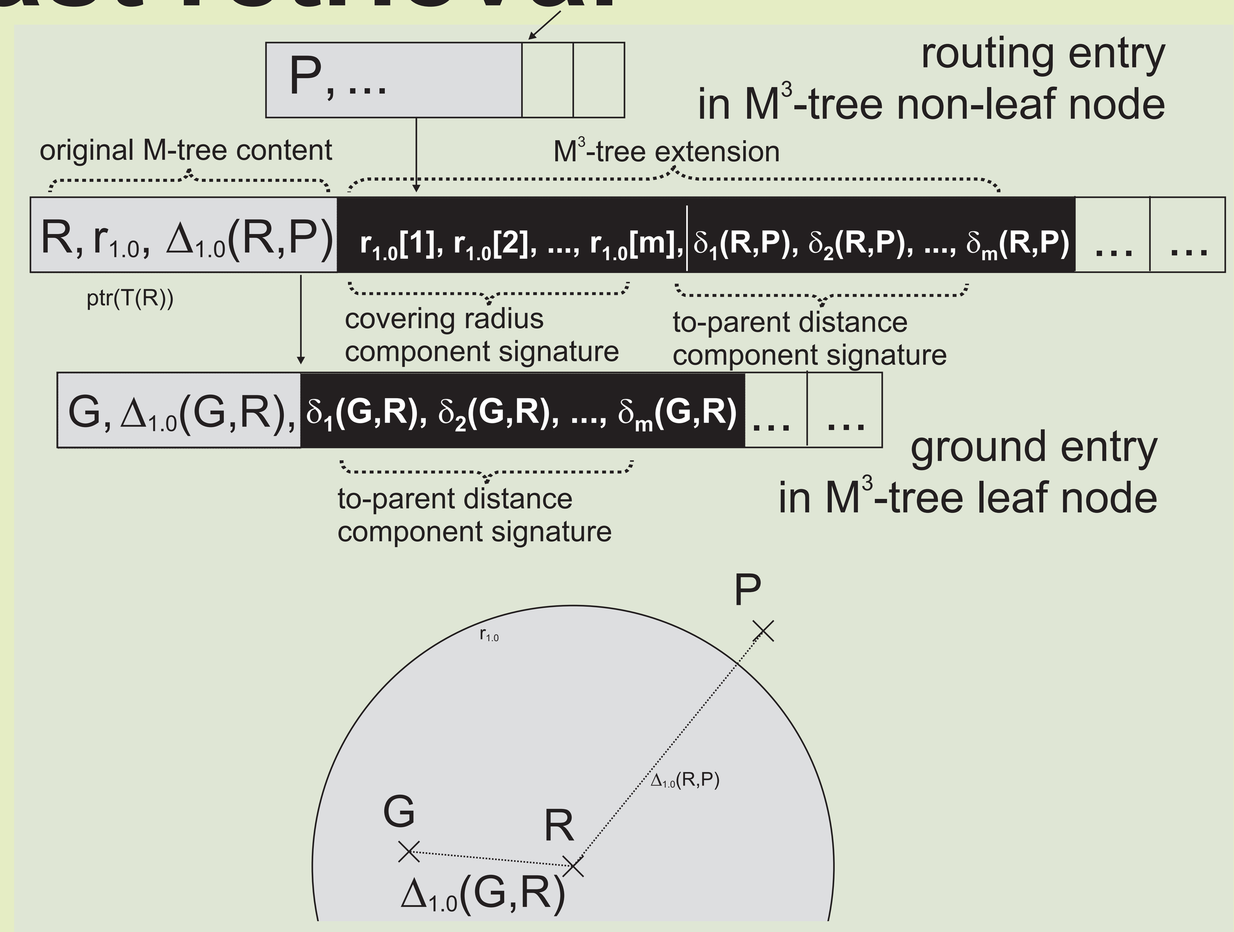
$$\Delta_{\mathbb{W}}(O_1, O_2) = \sum_{i=1}^{m} w_i \cdot \delta_i(O_1, O_2)$$

Adapted M-tree:
The 1-weighed combination is used as the index distance. Since the weights are maximal, the distance is an upper-bound to any query distance (for which the weights are lower than 1).
No structural changes to M-tree are needed [3].
Advantage:
Single index is sufficient to use query-weighted distances.
Disadvantage:
The indexing upper-bound distance is not very tight with respect to small weights. In such cases the querying performance deteriorates.

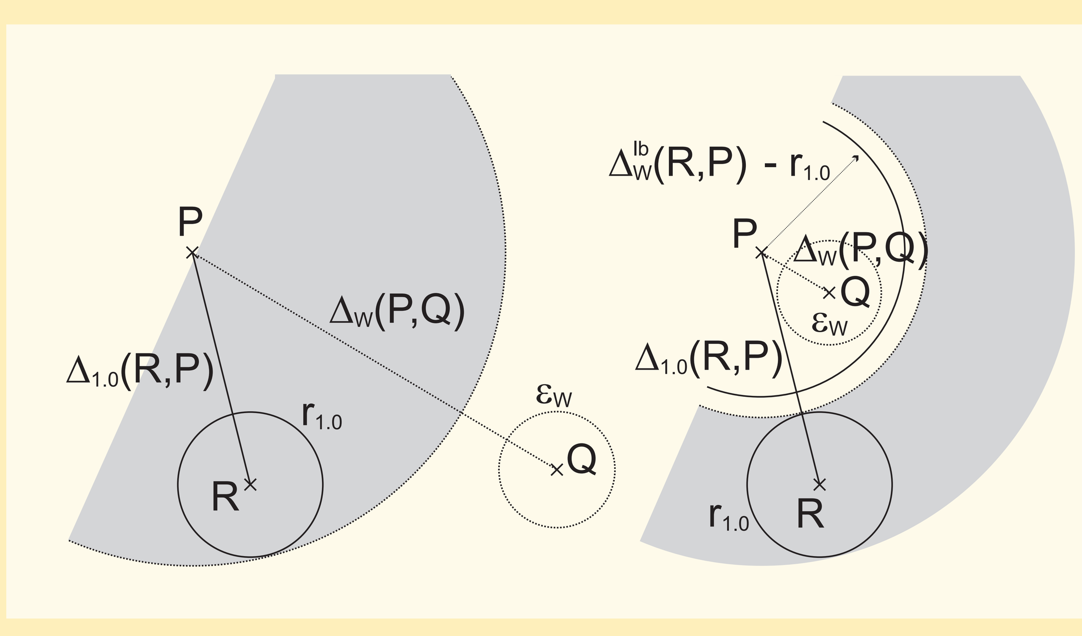


M³-tree:
The adapted M-tree is further extended such that partial distances or radii are stored separately. To achieve compact representation, the distance/radii components are stored as signatures. Due to partial distances we are able to establish much tighter upper bounds to the query distances. Moreover, the tightness of the upper bound is no more weight-dependent. The construction is not modified, we must just adjust the M-tree insertion and splitting routines to keep the partial components up-to-date.

Modified query algorithm is presented (proceedings).



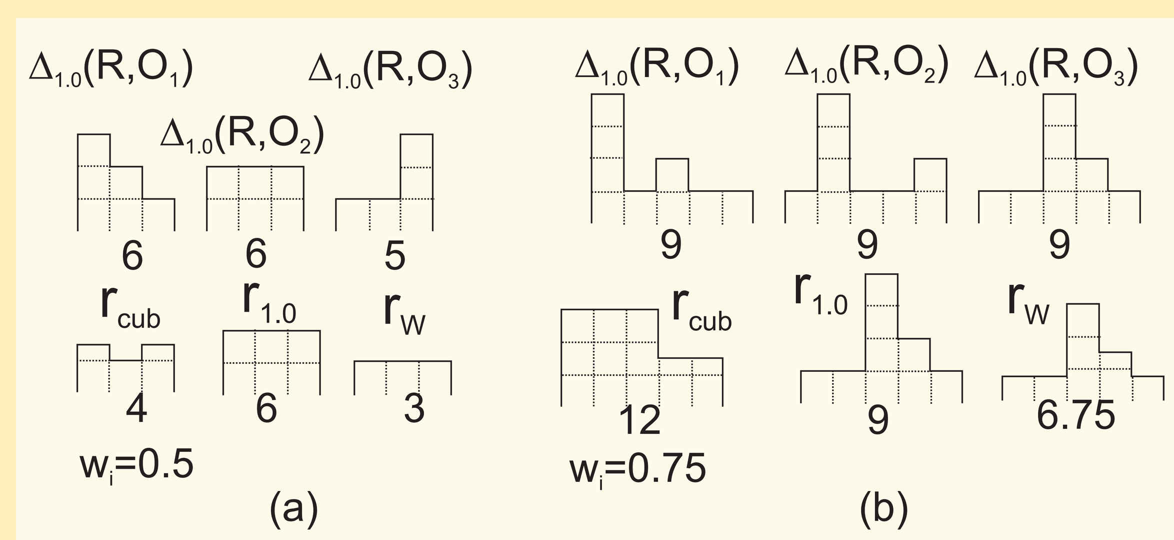
Query processing



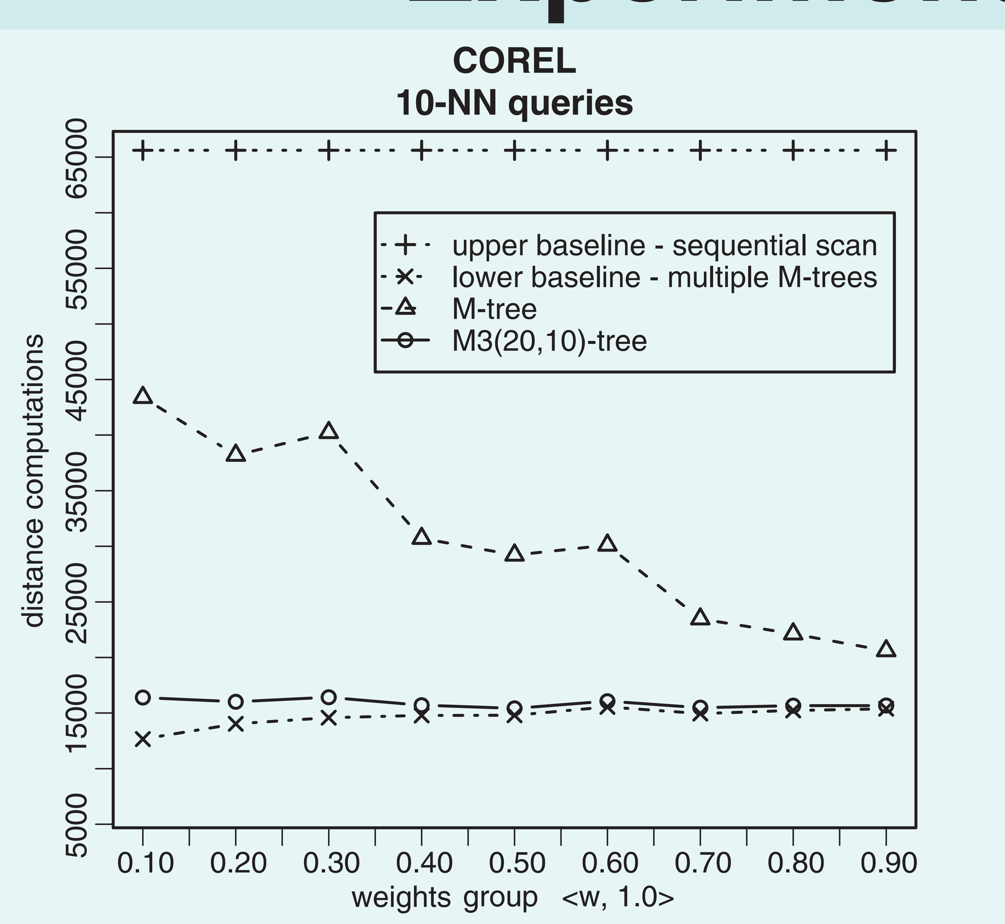
The query algorithms use upper/lower bounds to the actual query distance. The filtering makes use of overlap check whether the "overscaled" region radii intersect the query ball. The second filtering method makes use of the to-parent distances.

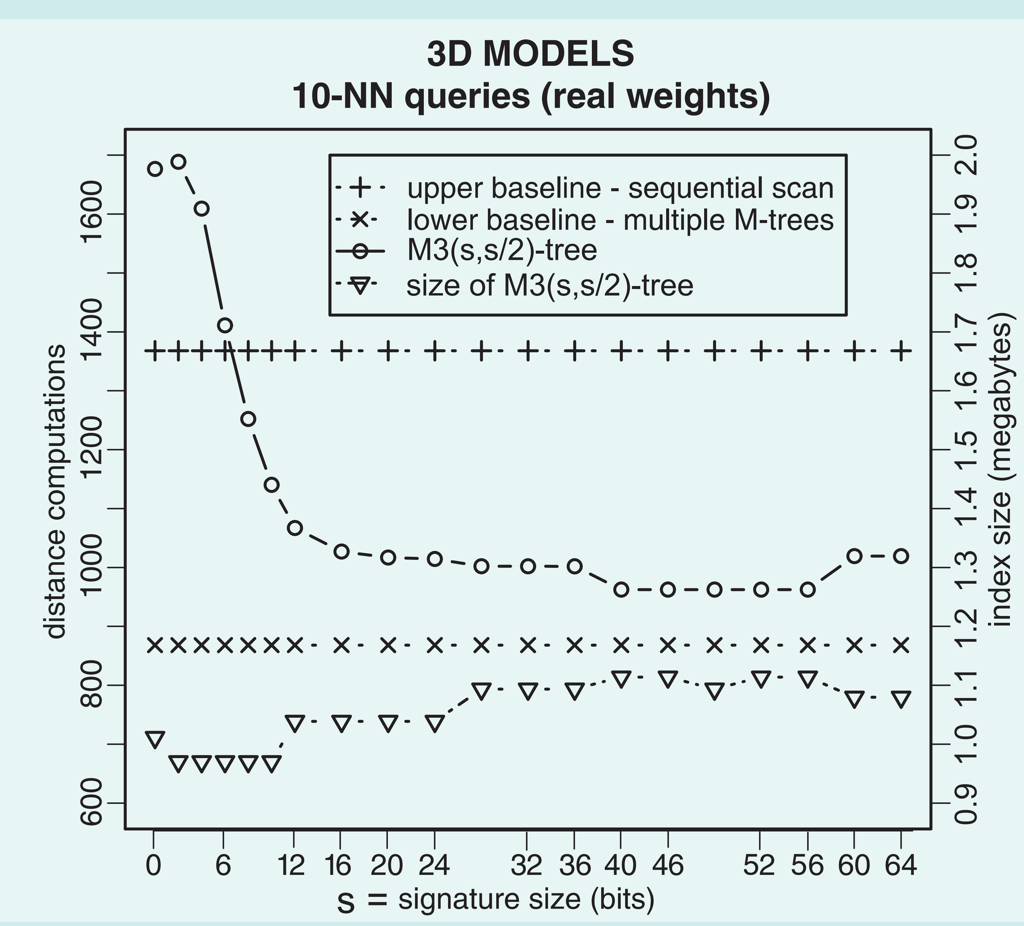
The tightened upper/lower bounds are utilized in all the original filtering checks, i.e. in the basic filtering as well as the parent filtering.

The main improvements are observed by using the (almost precise) to-parent distances. The usage is radii components is beneficial as well, however, due to "region nesting" the upper bound radius could not be as tight as the to-parent distances.



Experimental evaluation





References

[1] B. Bustos et al. Automatic selection and combination of descriptors for effective 3D similarity search. In *Proc. Intl. Workshop on Multimedia Content-based Analysis and Retrieval*, pp. 514-521. IEEE CS, 2004.

[2] B. Bustos et al. Using entropy impurity for improved 3D object similarity search. In *Proc. Intl. Conf. on Multimedia and Expo*, pp. 1303-1306, 2004.

[3] T. Skopal et al. Nearest neighbours search using the PM-tree. In Proc. 10th Intl. Conference on Database Systems for Advanced Applications, LNCS 3453, pp. 803-815. Springer, 2005.

Acknowledgments

This research has been partially supported by Czech grants GAČR 201/05/P036 and Information Society 1ET100300419 (second author).
The first author is on leave from the Department of Computer Science, University of Chile.