View Centered Video-based Object Recognition for Lightweight Devices

László Czúni*, Metwally Rashad†
*University of Pannonia, 8200 - Veszprém, Hungary
czuni@almos.uni-pannon.hu - metwally.rashad@virt.uni-pannon.hu

***Abstract*—Abstract—Video-based object recognition faces the problem of multi-view object variance, noisy conditions, and limited computational resources. In our previous work, we introduced a multi-view recognition approach with a compact global image descriptor coupled with orientation sensor data. Since our purpose is to run all computations in a handheld device, contrary to more intensive deep learning approaches, now we investigate the efficiency of our approach using a full representation image model with KD-Tree indexing. Experimental results show the effectiveness of our approach through three databases using noisy images.**

***Index Terms*—object recognition, view centered recognition, orientation sensor, image retrieval, KD-Tree.**

I. Introduction

Recently used multilayer deep learning recognition approaches discover intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation of the previous layer. While there are such successful techniques for object recognition in large databases [1], [2], these techniques require tremendous performance regarding processing power and memory.

In contrast we aim to research lightweight methods which can run in embedded systems without high performance backend support. More conventional (not multilayer) methods focus on the problem of local and/or global feature based object recognition. While local feature descriptors represent an image by multiple descriptors, sampled at different locations in the image, global descriptors describe the image as a whole with limited amount of data. The most important attributes of the global approach is the relatively low computational power needed for extraction, small size and less intensive comparison in image search engines.

We are to find proper tools for the recognition of 3D objects in noisy environments with handheld devices. We have chosen a view centered recognition model where the compact color and edge directivity descriptor (CEDD) [3] [4] is used for visual comparisons. As we have already shown 3D orientation sensors can enhance the performance of such approaches [5], now we also show the effectiveness of KD-Tree structures with a ranking method and give comparisons with previous techniques.

II. Previous Works

Without trying to make a comprehensive review of this ever improving area we focus on some close selected papers. Handheld 3D object recognition is a difficult task due to changing viewpoints, varying 3D to 2D projections, and possible different noises (e.g. motion blur, color distortion). In [6] authors created object models with the help of SIFT points which are tracked from frame to frame. Video matching is based on the comparison of every view of the query with all components of the optimized models of candidates. While the accuracy was about 83% in case of 25 objects, the complexity can be considered still high. In [7] also SIFT points were used as image features. The underlying topological structure of an image dataset was generated as a neighborhood graph of features. Motion continuity in the query video was exploited to demonstrate that the results obtained using a video sequence are much robust than using a single image. The ratio of correct retrieval increased to 80% with the method from only 20% of single image queries in case of 100 objects while the complexity was not discussed. Video based object recognition approaches can use thousands of views and thus can easily suffer from high complexity. Kd-Tree is an efficient data structure, established by Friedman, Bentley and Finkel [8], and is often used for fast indexing and retrieval. In [9] authors improved the Kd-Tree for a specific usage: indexing a large number of SIFT and other types of image descriptors. They also extended priority search to priority search among multiple trees in a simultaneously way. In [10] parallel Kd-Trees were explored for ultra large scale image retrieval in databases containing dozens of millions of images. In our paper we also use Kd-Trees, however, the number of candidate views (typically below 100,000) does not require the use of such multiple tree solutions.
In [5] authors introduced a novel retrieval mechanism using the camera's orientation sensors. Our paper is a step forward to extend this model with tree indexing and candidate ranking to find the best balance in retrieval rate and computational demands.

III. The Proposed View Centered Recognition

There are two main approaches for the recognition of 3D objects: object and view centered. In object centered representations (e.g. structure from motion) object features must describe the 3D structure. The main disadvantage of

978-1-4673-9554-0/16/$31.00

these methods is that they require the computationally complex simultaneous camera calibration and 3D reconstruction.

We have chosen view centered representations, where the outlook of the object is modeled from different viewpoints with multiple 2D images so there is no effort taken to reconstruct the 3D structure. The issue of choosing the features to be extracted should be guided by the following concerns: to carry enough information to distinguish images; to be invariant to distortions; to be subject of fast and robust comparisons. In our previous tests [11] we investigated different types of descriptors in real-life circumstances: MPEG-7 based methods (MPEG7_CLD, MPEG7_EHD, MPEG7_SCD, MPEG7_Fusion); Local feature based methods (SURF, SURFVW [3], SIFT); Compact Composite Descriptors [3] [4] (CompactCEDD, CEDD, CompactFCTH, FCTH, JCD, CCD Fusion, CompactVW); and others (Tamura texture descriptor, Color Correlogram and Correlation (ACCC) [12], MPEG7-CCD_Fusion [4]). While we know that there are always newer and better global and local descriptors [13], the selection of the most appropriate one is out of focus of this paper. Based on previous quantitative evaluations we have chosen the CEDD descriptor which combines color and texture information of a rectangular region in histograms in a vector of length 144. Texture information of image blocks is modeled by classifying them into six classes: non-edge, vertical, horizontal, 45-degree diagonal, 135-degree diagonal and nondirectional edges. Each class is described by 24 bin color histogram based on fuzzy color selection. For more details about CEDD see [3].

In our model we have not only one but several CEDD descriptors of the objects extracted from different viewing directions. In each case the object is located in the center of the image while the elevation and azimuth can be varied due to camera motion [5]. The similarity between two CEDD vectors is efficiently given by the Tanimoto Coefficient [4]. Let q_i be the descriptor of the ith view from the query and c_j be the descriptor of the jth view of a candidate. The Tanimoto Coefficient is then:

$$T(q_i,c_j) = \frac{q_i^T c_j}{q_i^T q_i + c_j^T c_j - q_i^T c_j} \quad (1)$$

where q_i^T is the transpose vector of the descriptor q_i. In case of absolute congruence of the vectors, the Tanimoto coefficient takes the value 0, while in case of maximum deviation the coefficient tends to 1. Please note, that we need a modified Tanimoto distance to achieve rough rotation invariance:

$$T^R(q_i,c_j) = \min_{roll} T(q_{i,roll},c_j) \quad (2)$$

where $roll \in 0°,45°,90°,135°$ and $q_{i,roll}$ means that orientation specific texture classes are shifted with some positions within the CEDD vector.

Our basic Full Representation (FR) recognition approach with Kd-Tree indexing is outlined in Figure 1. Object knowledge database contains several views of all known objects with the extracted CEDD vectors. For all object views we build only one Kd-Tree where the leaf nodes contain a number of similar views of possibly different objects. Given a set of query CEDDs (generated from multiple views in the query sequence) we travel the tree to its leaf node and measure the Tanimoto Coefficient between query's CEDD and all views found there, generating a limited length (e.g. $l = 4$) retrieval list.

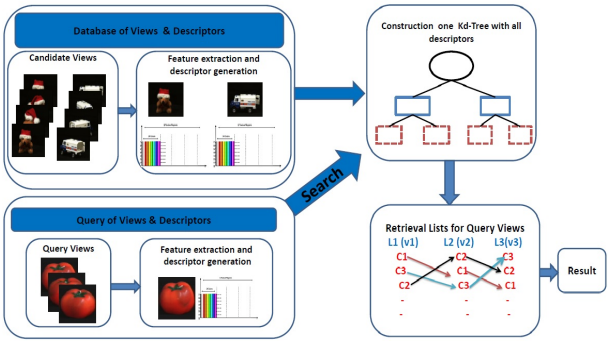

Figure 1. Summary of the proposed method (see text for details).

That is we have a sequence of retrieval lists, one for each query view, containing candidate object labels. Now, for each candidate, occurring in any of the lists, we compute the accumulated rank. If an object is not on a list it gets rank $l+1$. The candidate object which has minimum accumulated rank is chosen as the result of the search.

IV. Experiments and Results

A. Databases

We have three datasets: The first is our small database (sUP) including 16 objects (fully 3D-shaped) like some types of toy cars, headset, books, coffee cups, stapler, plastic bags, computer mouse, pens. Between 44-73 views per object were captured from the same elevation but from different azimuth leading to approximately 900 images. Image sizes and side ratios varied a lot as shown in Figure 2. The second database is the COIL-100 database [14] including 100 different objects, where 72 images of each object were taken at pose intervals of 5°. The third one is the ALOI database [15] including 1000 small objects, where 72 views of each object were recorded by rotating the object in the plane at 5° steps (Figure 2). The query dataset is composed of 10 randomly selected images of each object, strongly distorted with motion blur or additive Gaussian noise. We used the built-in function of Matlab *imnoise* with standard deviation $sd = 0.012$ to generate additive Gaussian noise and made motion blur by *fspecial* with parameters $len = 15$, and angle $\theta = 20$ degrees. Some examples of the queries are shown in Figure3.

B. Retrieval Performance

The purpose of the tests were to see the hit rate and the running time of the FR approach compared to other approaches introduced in [5]. The method labeled "Image" used a sequential full search of all views of the queries. The

978-1-4673-9554-0/16/$31.00 ©2016 IEEE

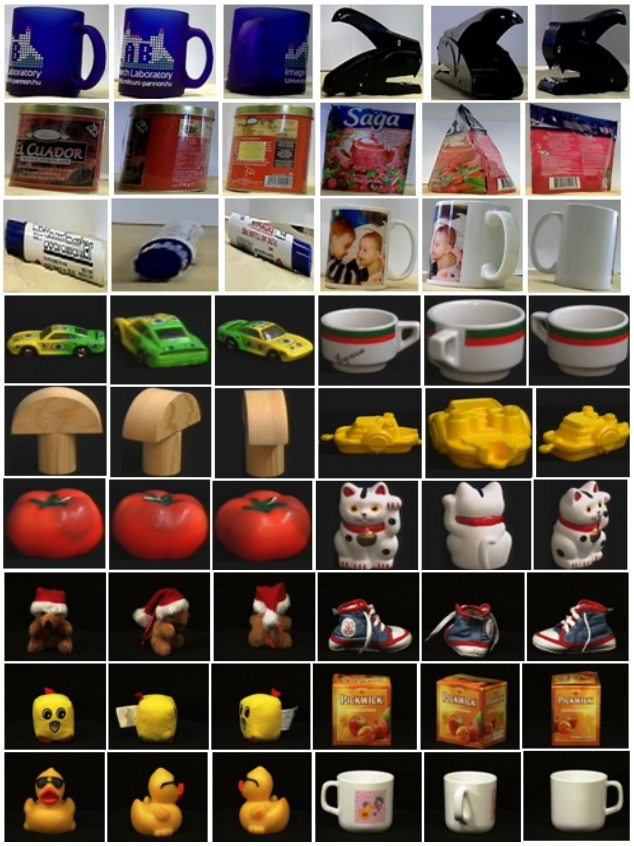

Figure 2. Object examples. Top: sUP, middle: COIL-100, bottom: ALOI databases.

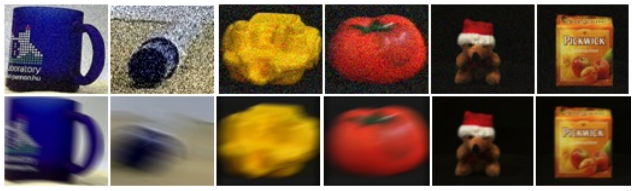

Figure 3. Noisy and blurred query examples from the three databases.

candidate with the lowest average Tanimoto Coefficient was retrieved as the matching one. The method "Multi-sensor" used only the first view of the query to run a full search among candidates, other views of the query were matched to the appropriate pair of the candidates based on the difference in orientation. Multi-sensor method showed similar hit rate while giving 2-3 times speed up in case of 8 query views [5]. For recent tests a Samsung SM-T311 tablet equipped with Android 4.2.2 Jelly Bean, 1 GB RAM, and ARM Cortex A9 Dual-Core 1.5 GHz Processor was used. For each test database there are separate graphs for motion blur and Gaussian noise illustrating the hit rate vs. the number of frames in the query (see Figures 4, 5, and 6). We can observe that the FR method outperforms the previous approaches in all cases as the number of query views reaches a certain value. This value is typically lower for Gaussian noise than for motion blur. Figure

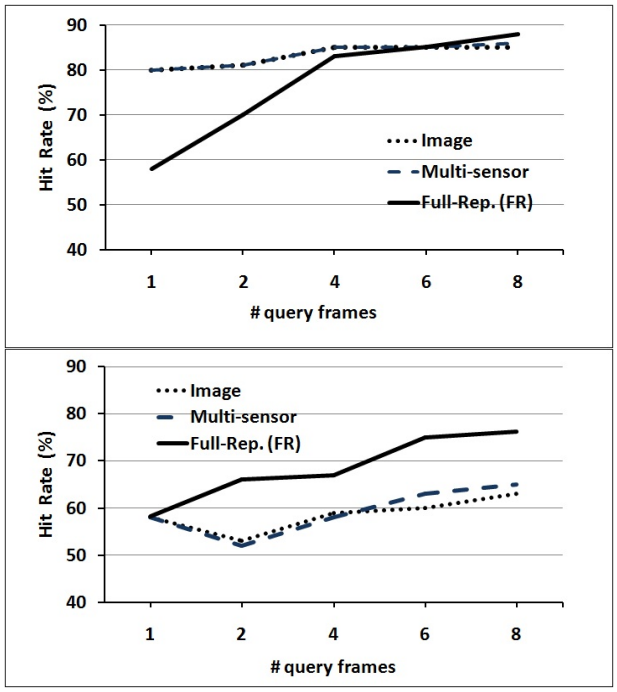

Figure 4. Average hit rate for strong motion blur (top) and additive Gaussian noise (bottom) for the sUP image set.

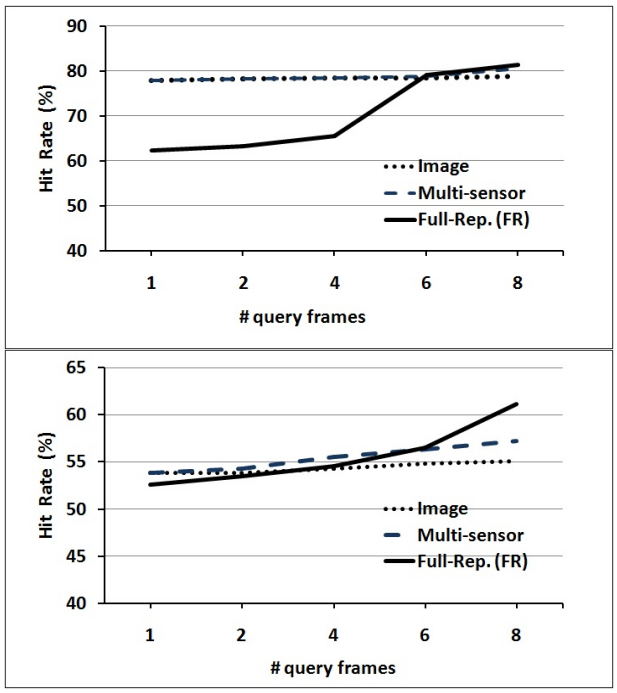

Figure 5. Average hit rate for strong motion blur (top) and strong additive Gaussian noise (bottom) for the COIL-100 image set.

7 illustrates the average running time (based on 10 queries) of the different retrieval methods in case of sUP giving no doubt about its efficiency. Please note that the extraction of the CEDD descriptors, which is about 0.4 sec on the mobile

978-1-4673-9554-0/16/$31.00 ©2016 IEEE

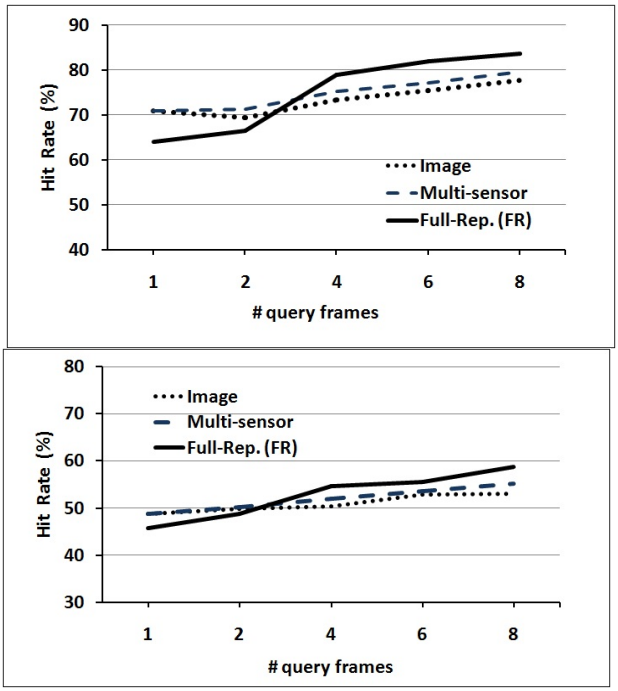

Figure 6. Average hit rate for strong motion blur (top) and strong additive Gaussian noise (bottom) for the ALOI image set.

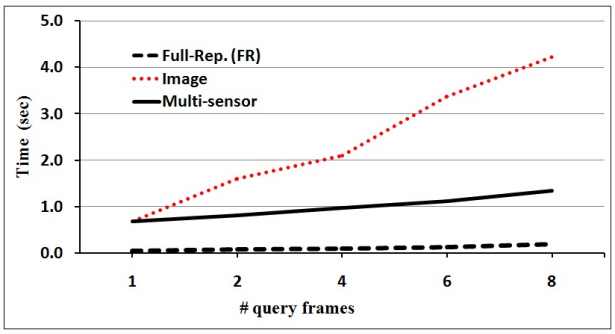

Figure 7. Average running time of the three methods for the sUP image set.

platform, is not included in these data.

Finally, Figure 8 illustrates the average running time (based on 10 queries) of the FR retrieval method in case of two sets. Even if the number of possible objects is high (100) and 7200 views are considered, we can apply the FR approach below 8 seconds for 8 queries proving the feasibility of the approach for lightweight tools in noisy conditions.

V. Conclusion

Our motivation is to create a multi-view object recognition technique that is capable to achieve real-time recognition of 3D objects with handheld devices. To achieve our goal we already investigated the use of sensor fusion and now tested Kd-Tree indexing of global image descriptors. The realistic evaluations showed that increasing the number of views, resulting in better hit rate but also requiring higher computation power, can

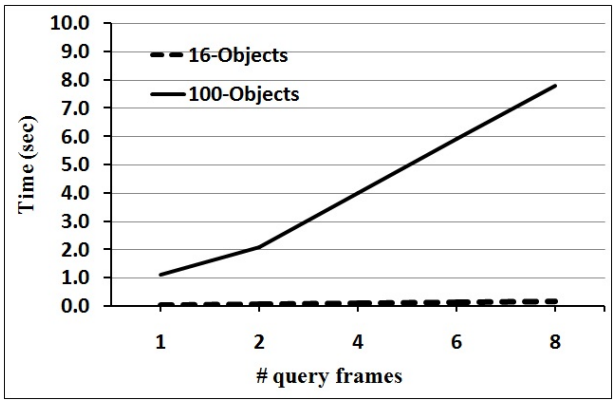

Figure 8. Average running time for the FR image search.

be handled if motion sensors and/or tree indexing is used. In future we plan to combine the information fusion with indexing techniques to achieve even better results.

References

[1] Szegedy, Christian, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015

[2] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105, 2012.

[3] S. A. Chatzichristofis and Y. S. Boutalis. Accurate Image Retrieval based on Compact Composite Descriptors and Relevance Feedback Information. *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 207–244, 2010.

[4] S. A. Chatzichristofis, Y. S. Boutalis and M. Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. *In Sixth IASTED Int. Conf. on Signal Processing, Pattern Recognition and Applications (SPPRA)*, pp. 134–140, 2009.

[5] L. Czúni and Metwally Rashad. Lightweight Video Object Recognition based on Sensor Fusion. *In International Workshop on Computational intelligence for multimedia understanding (IWCIM)*, pp. 1–5, 2015.

[6] A. Bruno, L. Greco and M. Cascia. Video Object Recognition and Modeling by SIFT Matching Optimization. *In ICPRAM*, pp. 662–670, 2014.

[7] NOOR, Humera, et al. Model generation for video-based object recognition. *In Proceedings of the 14th annual ACM International Conference on Multimedia*, pp. 715–718, 2006.

[8] J. Friedman, J. Bentley and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, Vol. 3, pp. 209-226, 1977.

[9] C. Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. *In IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2008.

[10] M. Aly, M. Munich and P. Perona. Distributed Kd-Trees for Retrieval from Very Large Image Collections. *In British Machine Vision Conference (BMVC)*, 2011.

[11] L. Czúni, P. J. Kiss, A. Lipovits, M. Gal. Lightweight mobile object recognition. *In IEEE International Conference on Image Processing (ICIP)*, pp. 3426–3428, 2014.

[12] A. Tungkasthan, S. Intarasema and W. Premchaiswadi. Spatial Color Indexing using ACC Algorithm. *In 7th International Conference on ICT and Knowledge Engineering*, pp. 113–117, 2009.

[13] O. Miksik, K. Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. *In 21st International Conference on Pattern Recognition (ICPR)*, pp. 2681–2684, 2012.

[14] S. A. Nene, S. K. Nayar and H. Murase. Columbia Object Image Library (COIL-100). *Technical Report CUCS*, 1996.

[15] J. Geusebroek, G. J. Burghouts and A. W. M. Smeulders. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, pp. 103–112, vol. 61, 2005.

978-1-4673-9554-0/16/$31.00 ©2016 IEEE