# LIGHTWEIGHT VIDEO OBJECT RECOGNITION BASED ON SENSOR FUSION

*László Czúni and Metwally Rashad*

Image Processing Laboratory, University of Pannonia, Veszprém, Hungary

## ABSTRACT

While there are several promising approaches for visual object recognition the application of lightweight devices under varying image conditions and using low quality images still causes lots of problems to be solved. We introduce a new retrieval mechanism including standard orientation sensors helping the visual recognition process. We apply a view based model of the objects and the matching of the query and candidate images is based on compact image descriptors coupled with relative orientation. Besides introducing the concept we show the effectiveness of our approach through two datasets using noisy and blurred images.

***Index Terms***— object recognition, view centered recognition, orientation sensor, image retrieval, COIL-100

## 1. INTRODUCTION

Optical recognition has many problems in general such as scaling, illumination changes, partial occlusion, and background clutter, in case of capturing 3D objects with mobile devices viewpoint variation and image noise (e.g. motion blur due to hand shaking in poor lighting conditions) can decrease the recognition rate tremendously. Numerous recognition algorithms have been developed, most of them apply single image-based recognition. Single view methods may easily fail when there is strong similarity between the captured images or when the background clutter or partial occlusion masks distinctive features. Video based approaches can use more views but suffer from the increased complexity. To avoid big data/cloud processing type of solutions rises a need for efficient lightweight but robust techniques that could run in cheap embedded systems without a high performance back end support. In our paper we discuss a multi-sensor approach for video-based object recognition where a user moves a mobile camera around a target object of interest, while keeping the object roughly in the center of the viewfinder. The extraction of image features and the retrieval algorithm (using orientation data) is running in the lightweight client. Two datasets were used to show the efficiency of the proposal.

978-1-4673-8457-5/15/$31.00 

## 2. RELATED WORKS

In an early paper of [1] recognition was achieved from video sequences by employing a multiple hypothesis approach. Appearance similarity, and pose transition smoothness constraints were used to estimate the probability of the measurement being generated from a certain model hypothesis at each time instant. A smooth gradient direction feature was used to represent the appearance of objects while the pose was modeled as a von Mises-Fisher distribution. Recognition was achieved by choosing the hypothesis set that has accumulated the maximum evidence at the end of the sequence. Unfortunately, the testing of the method was carried out on four objects only. In [2] 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 query frame with all components of all models. While the accuracy was about 83% in case of 25 objects, the complexity is high. In [3] also SIFT points were used as image features. The underlying topological structure of an image dataset was generated as a neighborhood graph. 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. Complexity is not discussed in the paper. In [6] in addition to the camera they used the accelerometer and the magnetic sensor to recognize the landscape. Clustered SURF (Speeded Up Robust Features) features were quantized using a vocabulary of visual words, learnt by k-means. For tracking objects the FAST corner detector was combined with sensor tracking. Because of the small storage capacity of the mobile device a server-side service was needed to store the large number of images. In [7] we showed that CEDD is quite tolerant for different noises and can be computed in today mobile platforms.

## 3. VIEW CENTERED RECOGNITION

There are two main approaches for the recognition of 3D objects. In object centered representations, such as structure from motion methods, object features describe the 3D structure or volume of the object. The main disadvantage of these methods is that they require the computationally com-

plex simultaneous calibration of camera and 3D reconstruction. Contrary, in case of view centered representations, the approach we follow, the outlook of the object is modeled from different viewpoints so there is no effort taken to reconstruct the (2D or 3D) structure of the object. Rather information (orientation changes of the camera and image features) is collected and organized to be used for object recognition.

### 3.1. Image Feature Extraction

We do not attempt to give a review on image feature extraction in our paper just list some possible methods we thought would serve as the basis of a robust recognition engine. In our previous tests [7] we investigated the following four types of descriptors: MPEG-7 based methods (MPEG7_CLD, MPEG7_EHD, MPEG7_SCD, MPEG7_Fusion); Local feature based methods (SURF, SURFVW [8], SIFT ); Compact Composite Descriptors [8] [9] (CompactCEDD, CEDD, CompactFCTH, FCTH, JCD, CCD Fusion, CompactVW); and others (Tamura texture descriptor, Color Correlogram and Correlation (ACCC) [10], MPEG7-CCD_Fusion [9]). Unfortunately, the SIFT based method ran extremely slow (about two orders slower than compact descriptors) in initial tests compared to others and its performance was not better than the average of all. Even it seemed to be very sensitive to motion blur so it was neglected in our further experiments. Please note that although there are several much faster local descriptors ([11]) than SIFT, the selection of the most appropriate one is out of focus of this paper. The chosen CEDD descriptor, found quite robust in previous works, 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 on CEDD see [8].

### 3.2. Similarity of Descriptors

According to previous tests the similarity of two CEDD descriptor vectors are efficiently given by the Tanimoto Coefficient [9]. Let $q_i$ be the descriptor of the $i$th frame from the query and $c_j$ be the descriptor of the $j$th frame of a candidate. The Tanimoto Coefficient is then:

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

Even if the sole of the object is fixed, the relative orientation of the camera (compared to the object) can be changed from time to time and thus the rotation of the camera can be described by pan, tilt, and roll. While we can get rid of the problem of different pan and tilt settings if object tracking is applied (see in later Section) camera roll should be handled differently. Basically, CEDD is not rotation invariant but with the modification of the Tanimoto distance rough rotation invariance can be achieved:

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

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.

### 3.3. Model Generation

In our model we have not only one but several CEDD descriptors of the objects extracted from different viewing directions (see Fig. 1 for illustration). In each case the object is located in the center of the image while the elevation and azimuth can be varied due to camera tilt, pan, and translation. Each descriptor is coupled with the orientation data giving the elevation and azimuth in degree measured with the digital compass and acceleration sensors. Azimuth angle should be considered as a relative value since the object can be rotated, that is we need an azimuth matching mechanism (built into a modified similarity function in the next Section). We used the built-in accelerometer and compass sensors to measure the orientation of the camera for each view.To reduce the database size several visually similar frames can be removed from the database. Let $Fi$ and $Fj$ be two consecutive frames taken at different azimuths. If the difference in $T(Fi, Fj)$ (measured by the Tanimoto Coefficient)is below threshold $th$ then $Fj$ is simply deleted. As shown in Fig. 2 the difference between $F1$ and $\{F2, F3, F4\}$ is less than $th$ but it is greater for $F5$, then frames $\{F2, F3, F4\}$ are deleted from the object model.

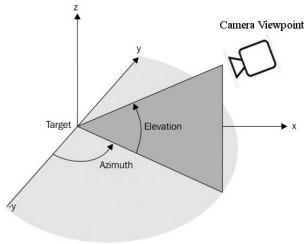


**Fig. 1**. Model generation setup

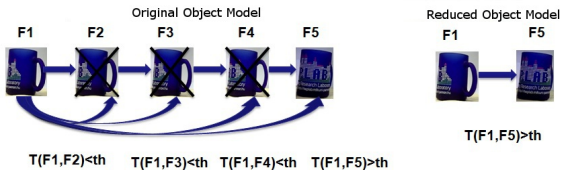


**Fig. 2**. An example of model size reduction.

### 3.4. Retrieval: The Similarity of Model and Query

In our application model an object is placed on a surface (e.g. table) and the camera moves around it keeping the object roughly in the center. The target object segmentation can be easily carried out by setting a target rectangle manually in the first frame then applying tracking such as Camshift [13] with low complexity. Without using temporal or orientational information one may use several frames from the query video to compute the average Tanimoto Coefficient resulting in complexity $O(N_c \cdot N_f^q \cdot N_f^c)$ where $N_c$ is the number of candidate objects, $N_f^q$ is the number of frames in query and $N_f^c$ is the number of frames in candidates (referring to object model size). Contrary, we show that testing only one frame from the query against all model frames then using the known relative orientation information for the other frames results in much lower complexity but similar hit rate. That is we defined the following similarity function:

$$T^{multi-sensor}(q,c) = \frac{\min_j T(q_i, c_j) + \sum_{\forall k, k \neq i} T(q_k, c_{\alpha(k)})}{N_f^q} \tag{3}$$

where $i$ is randomly selected (in our current implementation) and $\alpha(k)$ means the frame which is at the same (or very close) relative orientation in the candidate model to $j$ as $k$ to $i$ in the query. The complexity of the multi-sensor method can be described as: $O(N_c \cdot (N_f^c + (2 \cdot (N_f^q - 1))))$. Since there is no guarantee that we find a frame at the exact relative position in the candidate we used the best matching of the left and right neighbors in the closest available orientations explaining the multiplication by 2 in the above complexity. For comparisons we also tested a similarity function (multi-sensor, full search) where all frames from the query were used the same way:

$$T^{multi-sensor,fullsearch}(q,c) = \frac{\min_\beta \sum_{\forall k} T(q_k, c_{\beta(k)})}{N_f^q} \tag{4}$$

where $\beta(k)$ denotes a frame at orientation which is at $\beta$ degree relative to the orientation of the frame $k$.

## 4. EXPERIMENTS

### 4.1. Datasets

We had two datasets: The first included 16 objects (fully 3D-shaped) like some types of 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. Objects were centered and a bounding box was manually defined for each image. Image sizes and side ratios varied a lot as shown in Fig. 3. As we can see the object size, shape, color, contrast can vary from view to view. The background of the objects were only roughly uniform and the surface of objects was sometimes glossy. The second database is the COIL-100 dataset [14] which includes 100 different objects, where 72 images of each object were taken at pose intervals of $5^\circ$. Fig. 4 shows some examples objects from COIL-100.

**Fig. 3**. Test object examples in increasing ID order.

**Fig. 4**. Test object examples in COIL- 100 dataset.

The query dataset is composed of 10 randomly selected images of each object strongly distorted with motion blur and additive Gaussian noise (the queries were left out from possible candidates for testing reasons). We used the built in function in Matlab *imnoise* with standard deviation $sd = 0.012$ to apply additive Gaussian noise on the clear dataset and created also distorted images with motion blur by different length pixel $len = 15$), with an angle of $\theta$ degrees in a counterclockwise direction ($\theta = 20$). Some examples of the queries are shown in Fig. 5.

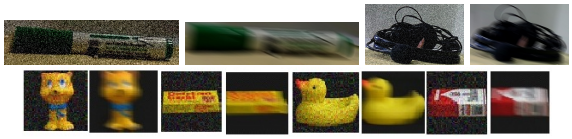

**Fig. 5**. Noisy and blurred query examples from the two datasets.

### 4.2. Model size

Fig. 6 contains the number of frames in the smaller dataset in case of each object category at different $Th$ threshold settings. The smallest number (7) at threshold 20 was found in case of the white-green bus while the largest number (20) for the green pen waht can be reasoned by the visual examination of the objects.

### 4.3. Retrieval Performance

We show the hit rate and the running time of the multi-sensor method compared to the method when all frames of

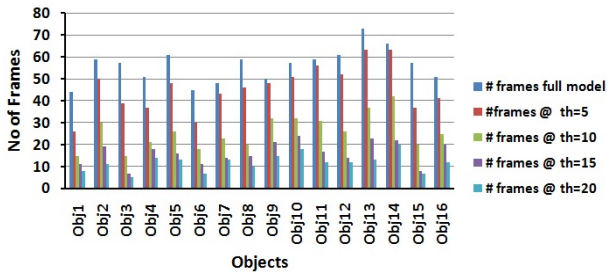


**Fig. 6**. Models sizes before and after frame reduction.

the query are used for retrieval without orientation data ("image" method) and compared to the method with all frames querying with orientation data ("Multi-sensor Full search" method). The effect of applying different $Th$-s and $N_f^q$-s is also explored. 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 in the tests. There are two graphs illustrating the hit rate vs. the number of frames in the query. As Fig. 7 shows for motion blurred images retrieval performance is greatly affected by the model size and as $N_f^q$ goes from 1 to 8 the hit rate increases about 5%. The strong additive noise resulted in lower values, especially when model size was reduced by $th$. In our retrieval we did

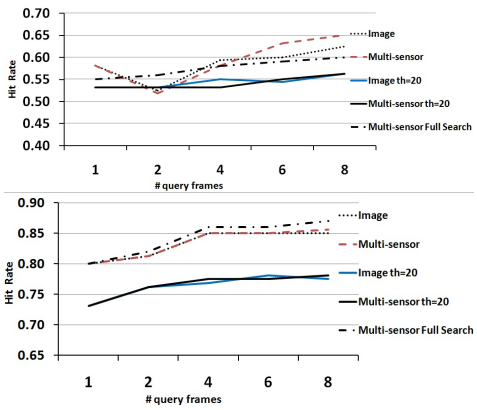


**Fig. 7**. Average hit rate for strong motion blur (top) and additive Gaussian noise (bottom).

not apply structuring of data, e.g. decision trees can even decrease the running time. Fig 9 illustrates the average running time (based on 10 queries) for the different retrieval methods. (Please note that the extraction of the CEDD descriptors, which is about 0.4 sec on the mobile platform, is not included in these data.) It is clearly visible that as the number of query frames is increasing the advantage of the multi-sensor method is growing while resulting practically the same retrieval rate. It means that using the multiple-sensor method at $N_f^q = 8$ we get the best performance at the running time of $N_f^q = 2$ of the only-images approach. (Please note that in case of Gaussian noise the highest hit rate would be above at $N_f^q = 8$, not investigated in this paper.)

Finally, we repeated our test on the COIL-100 database (see Fig. 8). We found our approach superior to other methods, unfortunately the analysis could not be included due to the limited length.

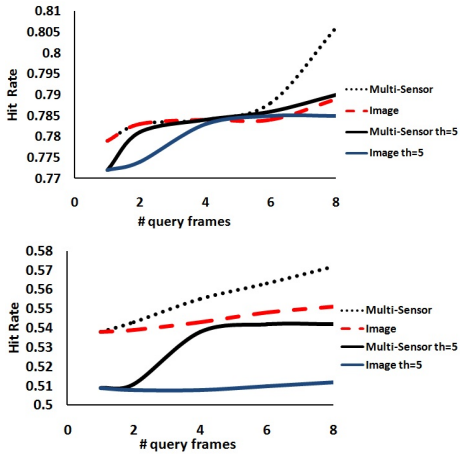


**Fig. 8**. Average hit rate for strong motion blur (top) and strong additive Gaussian noise (bottom) for the COIL-100 image set.

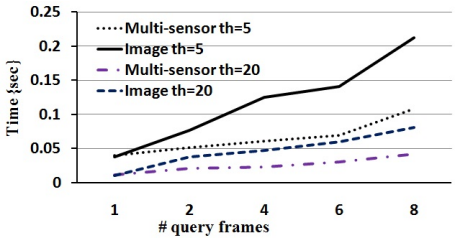


**Fig. 9**. Average running time for linear image search with different approaches and model size.

## 5. CONCLUSIONS

Our motivation was to create an object recognition model with lightweight solutions. Thus a compact global image descriptor was coupled with orientation data of the camera. This way the descriptor size and the number of matching steps could be kept low for video inputs. For evalution two datasets were used: the standard COIL-100 dataset and our own test data either with strong distortions. For both datasets we found that the multi-sensor method achieved the same or better hit rate than the full search with a fraction of running time.

# Acknowledgment

The work of L. C. was supported by Bolyai scholarship of the Hungarian Academy of Sciences.

## 6. REFERENCES


[1] O. Javed, M. Shah and D. Comaniciu. A probabilistic framework for object recognition in video. *In International Conference on Image Processing (ICIP)*, Page 2713-2716, 2004.

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

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

[4] Kumar, S.S. and Min Sun and Savarese, S.. Mobile object detection through client-server based vote transfer. *In IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, page 3290-3297, 2012.

[5] J. He, J. Feng, X. Liu, T. Cheng, T. Lin, H. Chung and S. Chang. Mobile Product Search with Bag of Hash Bits and Boundary Reranking. *In IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2012.

[6] S. Gammeter, A. Gassmann, L. Bossard, T. Quack and L. Gool. Server-side object recognition and client-side object tracking for mobile augmented reality. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, page 1-8, 2010.

[7] Czuni, L., Kiss, P. J., Lipovits, A., Gal, M.. Lightweight mobile object recognition. *In IEEE International Conference on Image Processing (ICIP)*, page 3426-3428, 2014.

[8] 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*, page 207-244, 2010.

[9] 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)*, page 134-140, 2009.

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

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

[12] Z. Chi,H. Yan and T. Pham. Fuzzy Algorithms: With Applications to image processing and pattern recognition. In Advances in Fuzzy Systems - Applications and Theory, Vol.10, 1996.

[13] A. R.J. Francois. CAMSHIFT tracker design experiments with Intel OpenCV and sai. *University OF Southern California Los Angeles Inst. for Robotics and Intelligent Systems*, 2004.

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