The Fusion of Optical and Orientation Information in a Markovian Framework for 3D Object Retrieval

László Czúni and Metwally Rashad

University of Pannonia, Veszprém 8200, Hungary,
czuni@almos.uni-pannon.hu,
WWW home page: http://keplab.mik.uni-pannon.hu

Abstract. In this paper we introduce a new 3D object retrieval model inspired by some well-known mechanisms of the human brain: viewer-centric recognition, Markovian estimations, and fusion of information originating from the visual and vestibular subsystems. We have built a Hidden Markov Model (HMM) framework where 2D object views correspond to states, observations are coded by compact edge and color sensitive descriptors, and orientation sensors are used to secure temporal inference by estimating transition probabilities between states. Our first evaluation results, over a database of 100 3D objects, are very encouraging: the fast and memory efficient new method outperformed previous models.

Keywords: object retrieval, information fusion, HMM, viewer-centric models

1 Introduction and Motivation

We introduce an efficient bio-inspired 3D object retrieval approach which can be implemented with very limited memory and processing power. Our motivation is to use ideas (viewer-centric object models with Markovian inference and information fusion) originating from the operation of the brain but also to avoid the complexity of hierarchical deep neural networks as it would be a direct copy of nature's successful mechanisms. In our research we focus on a relatively simple task: how to recognize/retrieve 3D objects by several 2D views taken from different directions.

Humans have only access to a limited subset of reality due to the limitation of attentional capacity and of sensitivity. As a result our experiences do not replicate the real world but rather create a construction or representation of it with prediction and estimation. One example is the temporal difference (TD) learning algorithm which has received attention in the field of neuroscience a long time ago [16]. TD mechanisms consider that subsequent predictions are correlated in some sense: TD learning adjusts predictions to match other predictions about the future. Evaluation of Markovian processes can be considered as a rough approximation of this, moreover Hidden Markov Models are able to make efficient

predictions considering the difference between the real world and its sensations with the help of probability functions. How we make representations of uncertainty greatly depends on the integration of data over time. The extent to which past events are used to represent uncertainty seems to vary over the cortex: primary visual cortex responds to rapid perturbations in the environment, while frontal cortices encode the longer term contexts within which these perturbations occur [9].

Information fusion is also very important in the creation of the brain's representations. Several examples for this are the different phenomena of vestibular and visual information co-processing. For example during long-drawn head rotations with the eyes closed, the elasticity of the cupula (a structure in the vestibular system providing the sense of spatial orientation) gradually restores it to its upright position. Thus the drive to the optokinetic response stops (misleadingly informing the brain that there is no motion). When opening the eyes in such situations, the world is seen moving and people feel giddy.

While the exact mechanisms of interaction between the different modalities are not always clear for the researchers, we will fuse orientation and visual information in a viewer-centered object recognition model. In cognitive science the recognition of objects from different views are described by two competing theories: according to the so-called object-centered approach [1] the structural description of simple parts play important role without explicit object representation from the different views. This can be imagined similar to the computer vision algorithms for object recognition with SIFT-like features [12]. In contrary, viewer-centered theory, supported e.g. by [7], suggests that this is done based on matching specific views to a set of templates, which requires explicit viewer-specific object representations.

Convolutional Neural Networks have very strong biological motivation and have been extensively used for image-based recognition, detection, retrieval, and image segmentation. However, their complexity (also energy and memory requirements) is quite large to be applied for real-time image based recognition in embedded or mobile systems. What we knew before and has been shown experimentally in recent developments is that simple approximations to input or internal data representations can still result in satisfactory performance. For example the so called XNOR-Networks, where both the filters and the input to convolutional layers are binary, run 58× faster convolutional operations and show 32× memory savings [15].

Following the above bio-inspired concepts we introduce a retrieval model with the following features:

- it is viewer-centered with the storage of very limited number of 2D views,
- it fuses visual and orientation information,
- it utilizes the inference in temporal sequences of signals (Markovian),
- the relation of observations and hidden model states can be estimated with simple correlations,
- it relies on compact descriptors computed very fast,
- it can be successfully used for real-time video object retrieval with lightweight devices.

There are two main reasons we are not using deep neural network models. First, we can implement the aimed concepts (Markovian inference, information fusion, viewer-centric object models) very efficiently in a HMM framework. Second, we have knowledge of efficient compact descriptors, and can use the orientation information directly in the Markovian model for the temporal support (as explained later), i.e. there is no need for time consuming training and optimization of millions of parameters of the neural structures.
In the next Section we give a short overview of related papers. Then the proposed object views, as hidden states of a Markov model, state transitions, observable features, and the decoding and retrieval steps are defined in consecutive subsections. Section Experiments and Evaluations contains experimental data and analysis followed by Summary.

2 A Brief Overview of Related Papers

Optical object retrieval and recognition is a very large topic with thousands of theoretical articles and applications, now we focus only on some which are closely related to our aims and motivations.
HMMs are often used in different recognition problems such as speech, musical sound, or human activity recognition but we relatively rarely meet them in the recognition of 2D or 3D visual objects. This is natural since ordered sequences of features are needed to construct HMM models. In [10] affine invariant image features are built on the contours of objects, and the sequence of such features are fed to the HMM. This approach is interesting but seemed to be too unnatural to have later followers.
In [5] authors presented an approach for face recognition using Singular Values Decomposition (SVD) to extract relevant face features, and HMMs as classifier. In order to create the HMM model the 2 dimensional face images had to be transformed into 1 dimensional observation sequences. For this purpose each face image was divided into overlapping blocks with the same width as the original image and a given height, and the singular values of these blocks were used as descriptors. A face image was divided into seven distinct horizontal regions: hair, forehead, eyebrows, eyes, nose, mouth and chin forming seven hidden states in the Markov model. While the algorithm was tested on two standard databases, the advantage of the HMM model over other approaches was not discussed.
The method of Torralba et al. [17] seems to be more close to a real-life temporal sequence: HMM was used for place recognition based on the sequences of visual observations of the environment created by a low-resolution camera. It was also investigated how the visual context affects the recognition of objects in complex scenes. There is no doubt that this approach has real cognitive motivation and relevance compared to those above.
Gammeter et al. [8] used accelerometer and magnetic sensor to help the visual recognition of the landscape. Clustered SURF (Speeded Up Robust Features) features were quantized using a vocabulary of visual words, learnt by k-means clustering. 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 used to store the necessary information for the algorithm. It is obvious that video gives much more visual information about 3D objects than 2D projections. Local feature descriptors (like SIFT, FAST, etc.) are often used for view-centered recognition. In [14] the underlying topological structure of an image set was represented as a neighborhood graph of local features. Motion continuity in the query video was exploited for the recognition of 3D objects.
The most similar viewer-centered HMM based 3D object retrieval method to ours was published by Jain et al. [11]. However, there are many differences to our work and many ambiguous details in [11]: it is not clear how the crucial emission and transition probabilities were estimated and also the dimension of the applied image descriptor (13) seems to be too small for real-life applications. The dataset in their tests included only gray-scale CGI without texture and no orientation sensor was used during the recognition.
Our early work, to utilize orientation information for object retrieval, can be found in [3]. Later we modified our method [4] to maximize a fitness function over a sequence of observations, based on the Hough transformation paradigm. While, as we have demonstrated by the above examples, the use of HMMs for object recognition is often a bit unnatural, turning our previous Hough framework to HMM is obvious and is also biologically motivated. As will be shown, our recent HMM model has better hit-rate and smaller complexity and encapsulates the bio-inspired concepts described above.

3 Object Retrieval with HMM

To achieve object retrieval will need to build HMM models for all elements of the set of objects (M). Then, based on observations, we find the most probable state sequence for all objects models. The state sequence among these, which is the most similar to the observation sequence, will belong to the object being retrieved.

3.1 Object Views as States in a Markov Model

Let $S = \{S_1, \cdots, S_N\}$ denote the set of N hidden states of a model. In each t index step this model is described as being in one $q_t \in S$ state, where $t = 1, \cdots, T$.

In our approach the states can be considered as the 2D views (or the average of some neighboring views) of a given object model. This can be easily imagined as a camera is targeting towards and object from a relative elevation and relative azimuth. The number of possible states should be kept low, otherwise the state transition matrix ($\mathbf{A}$) would contain too small numbers and finding the most probable state sequence would be too unstable. On the other hand, small number of states would mean that quite different views of some objects should be represented by the same descriptors, resulting in decreased similarity of model these views and actual test observations. Thus it is easy to see that the generation of states should be designed carefully. Often Gaussian mixtures are

used to combine the views of similar directions. Now we use static subdivision of the circle of 360°, into 2, 4, 6, and 8 uniform parts with 180°, 90°, 60°, and 45° correspondingly, with surprisingly good results as given in Section 4.
We define the initial state probabilities $\pi = \{\pi_i\}_{1 \leq i \leq N}$ based on the orientation range of states:

$$\pi_i = P(q_1 = S_i) = \frac{\alpha(S_i)}{360} \tag{1}$$

where $\alpha(S_i)$ is the size of orientation aperture of state S_i given in degree.

3.2 State Transitions

Between two steps the model can undergo a change of states according to a set of transition probabilities associated with each state pairs. In general the transition probabilities are:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{2}$$

where i and j indices refer to states of the HMM, $a_{ij} \geq 0$, and for a given state $\sum_{j=1}^{N} a_{ij} = 1$ holds. The transition probability matrix is denoted by $\mathbf{A} = \{a_{ij}\}_{1 \leq i,j \leq N}$.
To build a Markov model means learning its parameters (π, $\mathbf{A}$, and emission probabilities introduced later) by examining typical examples. However, our case is special: the probability of going from one state to an other severely depends on the users's behavior, interest and also on the frame rate of the camera. Thus we can not follow the traditional way, to use the Baum-Welch algorithm for parameter estimation based on several training samples, but can directly compute transition probabilities based on geometric probability as follows.
First define $\Delta_{t-1,t}$ as the orientation difference between two successive observations:

$$\Delta_{t-1,t} = \alpha(o_t) - \alpha(o_{t-1}). \tag{3}$$

Now define R_i as the aperture interval belonging to state S_i by borderlines:

$$R_i = [S_i^{min}, S_i^{max}[. \tag{4}$$

The back projected aperture interval is the range of orientation from where the previous observation should originate:

$$L_j = [S_j^{min} - \Delta_{t-1,t}, S_j^{max} - \Delta_{t-1,t}[. \tag{5}$$

Now we have arrived to estimate the transition probability by the geometrical probability concept applied on the intersection of L_j and R_j:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) = \frac{\alpha(L_j \cap R_i)}{\alpha(L_j)}. \tag{6}$$

Please see Figure 1 for illustration.

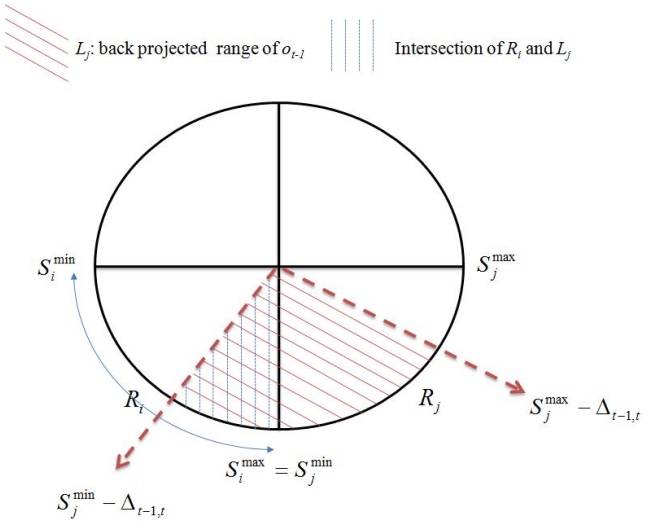

Fig. 1. Geometrical interpretation of transition probabiliites.

3.3 Hidden States Approximated by Observations with Compact Descriptors

The appearance of objects may significantly differ from those made during model generation under controlled circumstances. The changes in illumination, color balance, viewing angle, geometric distortion and image noise can result in heavily distorted feature descriptors. Thus observations only resemble the descriptors of the model states. Let $O = \{o_1, o_2, \cdots, o_T\}$ denote the set of observation sequence. The emission probability of a particular o_t observation for state S_i is defined as

$$b_i(o_t) = P(o_t|q_t = S_i) \tag{7}$$

In [4] we have shown that the CEDD (Color and Edge Directivity Descriptor) [2] is a robust low dimensional descriptor for object recognition. Being area based, pixels are classified into one of 6 texture classes (non-edge, vertical, horizontal, 45 and 135 degree diagonal, and non-directional edges). For each texture class a (normalized and quantized) 24 bin color histogram is generated, each bin representing colors obtained by the division of the HSV color space, resulting in feature vectors of dimension 144 (6×24). The similarity of CEDD vectors is computed by the Tanimoto coefficient:

$$T(e_i, c_j) = \frac{e_i^T c_j}{e_i^T e_i + c_j^T c_j - e_i^T c_j} \tag{8}$$

where e_i^T is the transpose vector of the query descriptor and c_i denotes the descriptors of object views. Rotational invariance can be achieved as given in [3]. Now eq. 7 can be rewritten as:

$$b_i(o_t) = \frac{T(C(S_i), C(o_t))}{\sum_{j=1}^{N} T(C(S_j), C(o_t))} \tag{9}$$

where C stands for the descriptor generating function of CEDD. Since each model state can cover a large directional range we will use the average CEDD vector, of available model samples within, to represent the whole state with a single descriptor.
Now we have the complete set of parameters of all HMMs denoted by $\lambda_k = (\mathbf{A}, b, \pi)$, $k \in M$. The task is to find the most probable state sequence $\hat{S}_k$, for all possible candidate objects, based on observations.

3.4 Decoding for Retrieval

We use the well-known Viterbi algorithm to get the state sequence with the maximum likelihood. The variable δ_t gives the highest probability of producing observation sequence $o_1, o_2, \cdots, o_t$ when moving along a hidden state sequence $q_1, q_2, \cdots, q_{t-1}$ and getting into $q_t = S_i$, *i.e.*

$$\delta_t(i) = \max P(q_1, q_2, \cdots, q_t = S_i, o_1, o_2, \cdots, o_t|\lambda) \tag{10}$$

It can be calculated inductively as

1. Initialization:
$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \tag{11}$$
2. Recursion:
$$\delta_{t+1}(j) = b_j(o_{t+1}) \max_i [a_{ij}\delta_t(i)], \quad 1 \leq j \leq N \tag{12}$$

Finally, we can choose the most probable state $\hat{i}$ ending at T:

$$\hat{i} = arg \max_i [\delta_T(i)] \tag{13}$$

To achieve object retrieval we have to find the most probable state sequence $\hat{S}_k$ with the above steps for all possible candidate objects. Now, to select the winner object, we have to compare the observations with the most probable state sequence:

$$\hat{k} = arg \max_{\forall k \in M} \left(\frac{\sum_{i=1}^{N} T(C(o_i), C(\hat{S}_{k,i}))}{N}\right) \tag{14}$$

4 Experiments and Evaluations

4.1 Test Dataset

The COIL-100 dataset [13] includes 100 different objects; 72 images of each object were taken at pose intervals of 5°. We evaluated retrieval with clear and heavily distorted queries using Gaussian noise and motion blur. The *imnoise*

function of Matlab, with standard deviation $sd = 0.012$, was used to generate additive Gaussian noise (GN) while motion blur (MB) was made by *fspecial* with parameters $len = 15$, and angle $\theta = 20°$. Some examples of the queries are shown in Figure 2.
For different tests different numbers (2, 4, 6, 8) of hidden states were generated by equally dividing the full circle. Each state was represented with its average CEDD descriptor vector.
To estimate the relative orientation of the camera with used the same built in IMU (Inertial Measurement Unit) sensor as in [4] with around 4.5° average absolute error with a 5.25° variance. The evaluation of our method with textured and varying backgrounds is for future work.

Fig. 2. First three lines: Clear samples from COIL-100. 4^{th} line: Example queries loaded with Gaussian additive noise. 5^{th} line: Example queries loaded with motion blur.

4.2 Hit-rate

The hit-rate of retrieval is measured by taking the average of 10 experiments with all 100 objects with randomly generated queries (the orientation angle of subsequent queries were increased monotonically). As shown in Figure 3 for different quality queries, as the number of queries increases the hit-rate increases monotonically. It is also true that higher number of states gives better results.

We tested no more states than 8, where it reached the maximum performance in most cases.
For comparison with the method of [4] we included the best results of the Voting Candidates algorithm denoted by VCI. There is an obvious 2-6% gain over VCI observable. Please note, that the same visual descriptors and orientation sensor was used by VCI in previous tests.

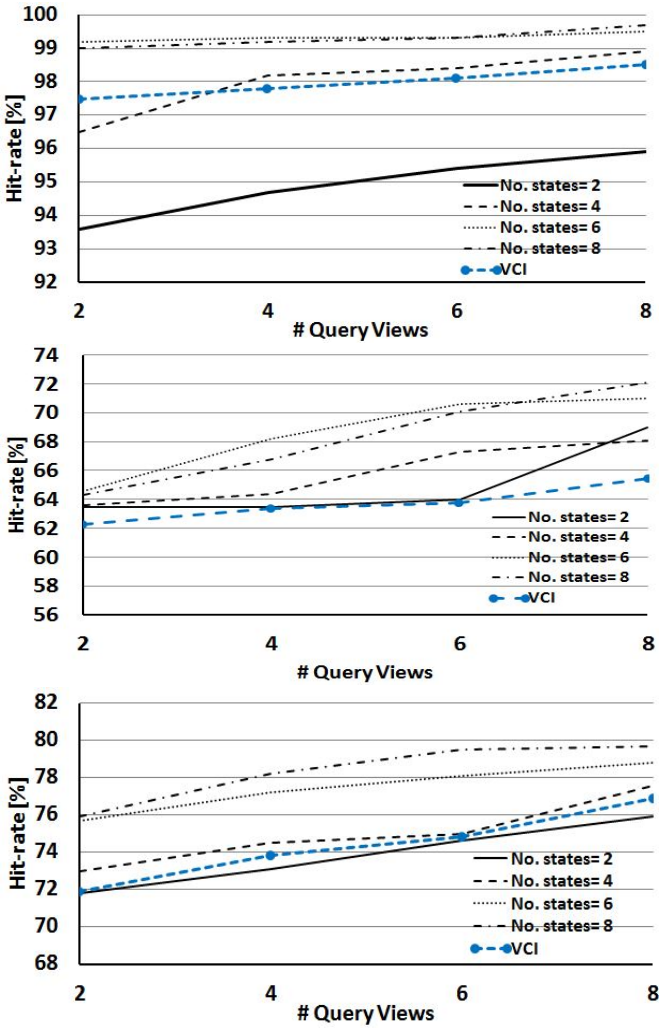

Fig. 3. First graph: average hit-rate with clear samples from COIL-100. Second graph: Queries loaded with Gaussian additive noise. Third graph: Queries loaded with motion blur.

4.3 Running Time and Memory Requirements

Tests were run on 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. No code optimization or parallelism was carried out and only the CPU was used during calculations. As given in Table 1 even for 8 queries the whole processing chain is within 1 second on the specified mobile computing hardware. This is a fraction of the complexity of VCI [4].

Table 1. Running times in seconds for the retrieval of one object from 100.

Phase	Number of Query Views (N_f^q)			
	2	4	6	8
CEDD generation	0.08	0.16	0.24	0.32
HMM evaluations	0.11	0.15	0.18	0.23
SUM	0.19	0.31	0.42	0.55

The advantage of using compact descriptors is the very limited memory requirement of object models. A CEDD descriptor occupies 144 Bytes in memory and orientation can be stored in 4 Bytes. For 100 objects and 8 states we need to store roughly 120 KB (100×8×148 Bytes).

5 Summary

The main purpose and contribution of our paper is twofold:

- building a bio-inspired object retrieval framework with Markovian inference and multimodal information fusion in a viewer-centric model, and
- showing that its implementation is robust and resource efficient to be used in mobile devices.

We presented our first results over a dataset of 100 3D objects with 7200 views using clear and noisy queries. While results are better than with our previous model, still there is a lot to do: we are developing a clustering technique to build optimal states instead of the uniformly distributed states and should work also on automatic object segmentation and tracking.

Acknowledgements The work and publication of results have been supported by the Hungarian Research Fund, grant OTKA K 120367 and by the framework of Széchenyi 2020 Programme, within project EFOP-3.6.1-16-2016-00015.

References

1. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological review, 94(2), 115 (1987)
2. Chatzichristofis, S. A., and Boutalis, Y. S.: Accurate Image Retrieval based on Compact Composite Descriptors and Relevance Feedback Information, International Journal of Pattern Recognition and Artificial Intelligence, 207–244 (2010)
3. Czúni, L. and Metwally, R.: View Centered Video-based Object Recognition for Lightweight Devices, *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 1–4 (2016)
4. Czúni, L., Metwally, R.: The use of IMUs for video object retrieval in lightweight devices, Journal of Visual Communication and Image Representation, Vol. 48, 30–42 (2017)
5. Dinkova, P., Georgieva, P., and Milanova, M.: Face recognition using singular value decomposition and hidden markov model. In 16th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (MAMECTIS 2014), 144–149 (2014)
6. Edelman, S., and Blthoff, H. H.: Orientation dependence in the recognition of familiar and novel views of three-dimensional objects. Vision research, 32(12), 2385–2400 (1992)
7. Fang, F., and He, S.: Viewer-centered object representation in the human visual system revealed by viewpoint aftereffects. Neuron, 45(5), 793–800 (2005)
8. Gammeter, S., Gassmann, A., Bossard, L., Quack, T., and Van Gool, L.: Server-side object recognition and client-side object tracking for mobile augmented reality. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, 1–8 (2010)
9. Harrison, L., Bestmann, S., Rosa, M. J., Penny, W., and Green, G. G.: Time scales of representation in the human brain: Weighing past information to predict future events. Frontiers in Human Neuroscience, 5, 37. (2011)
10. Hornegger, J., Niemann, H., Paulus, D., and Schlottke, G.: Object recognition using hidden Markov models. Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems, 16, 37–44 (1994)
11. Jain, Y. K., and Singh, R. K.: Efficient view based 3-D object retrieval using Hidden Markov Model. 3D Research, 4(4), 5 (2013)
12. Lowe, D. G.: Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, Vol. 2, 1150-1157 (1999)
13. Nene, S. A., Nayar, S. K. and Murase, H.: Columbia Object Image Library (COIL-100), *Technical Report CUCS* (1996)
14. Noor, H., Mirza, S. H., Sheikh, Y., Jain, A., and Shah, M.: Model generation for video-based object recognition. In Proceedings of the 14th ACM international conference on Multimedia, 715–718 (2006)
15. Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In European Conference on Computer Vision, Springer International Publishing, 525–542 (2016)
16. Schultz, W., Dayan, P., and Montague, P. R.: A neural substrate of prediction and reward. Science, 275(5306), 1593–1599 (1997)
17. Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A.: Context-based vision system for place and object recognition. Proceedings Ninth IEEE International Conference on Computer Vision, Vol. 1, 273–280 (2003)