Signal, Image and Video Processing (2021) 15:315–322
https://doi.org/10.1007/s11760-020-01743-y

ORIGINAL PAPER

Active multiview recognition with hidden Markov temporal support

Amr M. Nagy[1,2] · Metwally Rashad[2] · László Czúni[1]

Received: 31 December 2019 / Revised: 11 June 2020 / Accepted: 13 July 2020 / Published online: 31 July 2020

Abstract
Our paper deals with active multiview object recognition focusing on the directional support of sequential multiple shots. Since inertial sensors are easily available nowadays, we propose the use of them to estimate the orientation change of the camera and thus to estimate the probability of relative poses. With the help of relative orientation change, we can compute transition probabilities between possible poses and can use a hidden Markov model to evaluate state (pose) sequences and can thus increase the recognition rate. Furthermore, we can plan our next viewing position to minimize the risk of misclassification, resulting in higher overall recognition rates. Besides giving the theoretical details, we use two datasets to illustrate the performance of our model through several tests including occlusion, blur, Gaussian noise, and to compare to a solution with a long short-term memory network.

Keywords Multiview recognition · Active vision · Hidden Markov model · Inertial measurement unit · Long short-term memory

1 Introduction

Object recognition is important in many applications such as robot vision, autonomous vehicles or helping the visually impaired. While there is a long history of optical object recognition only in the last few years we saw significant improvements with the evolution of neural networks. It is a natural assumption that multiple shots can decrease ambiguity and if those shots are from different directions, the amount of information gathered from the object also increases. In this later case, we soon arrive to relative pose estimation where the appearance of the 3D object corresponds to its relative pose. Due to natural ambiguities, such as noise, occlusion and geometrical distortions, this estimation can be ill posed. Nowadays, LSTM (long short-term memory) networks are popular techniques to include temporal domains in the deep neural network (DNN) frameworks. While DNNs can be very accurate in recognition, their training, computational, and memory requirements are high. Unfortunately, when considering lightweight solutions, which can be crucial of futures' sensors technology, the temporal combination of multiple views is much less investigated. In our paper, we propose a lightweight variational approach, which can be combined with any single shot detection technique, including DNNs. The proposed model implements information fusion since besides 2D images it uses IMU (inertial measurement unit) sensors for the estimation of change in the relative orientation of the camera. The advantage of this fusion in recognition rates and in active vision is shown in our paper.
The barrier to use hidden Markov models (HMMs) in object recognition was always the real lack of ordered sequential information. Talking about sequential multiple observations the order of shots is determined by the actual behavior of the observer, resulting in a weak certainty in state transition probabilities. We resolve this issue with the utilization of IMUs, giving useful hints to estimate transitions on the fly. The advantages of our proposal can be summarized as:

We acknowledge the financial support of the Széchenyi 2020 program under the project EFOP-3.6.1-16-2016-00015 and the Hungarian Research Fund, Grant OTKA K 120367. We are grateful to the NVIDIA Corporation for supporting our research with GPUs obtained by the NVIDIA GPU Grant Program.

✉ Amr M. Nagy
amr.nagy@virt.uni-pannon.hu

Metwally Rashad
metwally.rashad@fci.bu.edu.eg

László Czúni
czuni@almos.vein.hu

[1] Faculty of Information Technology, University of Pannonia, Egyetem u. 10, Veszprém, Hungary

[2] Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt

– the number of shots can vary from one to any number and can be changed dynamically (in our experiments, we test 2, 4, 6, and 8 consequential queries),
– it can use any single shot recognition technique to evaluate the individual query images,
– it can easily incorporate active vision by proposing the next view to decrease uncertainty of recognition,
– it relies on low-cost IMU sensors available in many mobile and imaging devices.

1.1 Overview and contribution

The advantage of deep convolutional neural networks in object recognition is that in case of large number of training samples and proper training they can find a good combination of feature detectors and classifier sub-networks. Contrary, variational approaches can find good results (solutions with high probabilities) even if the conditions are far from optimal and these conditions could not be approximated during training. Our first attempt to use HMMs for multiview object recognition is in [6]. Now, we show our improved model and results, where we can plan the next viewing position to improve recognition rates. Also, we extended the evaluations and included test cases where the objects to be recognized are partly occluded. Our experiments, on these occluded data, with LSTMs and HMMs support the above theory regarding the vulnerability via improper training. In numerical evaluations, we use two standard datasets with the sum of 1100 object classes. Color and edge directivity descriptors (CEDDs) [2], as compact global feature descriptors, are calculated for individual shots.
In the next section, we overview related papers, and then in Sect. 3, we explain the proposed method. In Sect. 4, the used datasets are introduced, while the experiments and evaluations can be found in Sect. 5. Finally, we conclude our article in the last Section.

2 Related papers

In computer vision, the problem of recognition of 3D objects from different views can be approached in many ways; there are numerous topics where we can find related solutions. The keywords *video-based recognition*, *3D object recognition*, *multiple view recognition* all can be considered, but also specific domains such as face detection [16] or human skeleton-based recognition [3] can apply similar techniques. Naturally, also special 3D sensors (such as Kinect or Lidars) can also be utilized to help the detection, segmentation, and recognition of 3D objects [4]. Due to the lack of space, we cannot discuss such approaches. In our overview, we concentrate only on a few papers which we considered the most relevant.

The modeling of objects from different views was proposed by early techniques; an often used one is the so-called aspect graphs (e.g., [5,19]). For example, in [19], instead of recovering a full 3D geometry, parts are connected through their mutual homographic transformation. In this approach, a canonical view is a collection of canonical parts (colored patches of objects) that share the same view. We can interpret a canonical view as a subset of the overall linkage structure of the object category, and the linkage structure is to connect each pair of canonical parts. The resulting model is a compact summarization of both the appearance and geometry information of the object class. In [20], the method is similar, but probabilistic models are generated to capture the relative position of parts within each of the discretized viewpoints. While our purpose is similar, in our case the linkage is based on the fusion of visual and orientation information utilizing global descriptors and a modified HMM framework. Naturally, local descriptors are more efficient in generic object classifications, but in case of specific objects we found global descriptors sufficient and also very cost effective considering computational time and memory requirements [7].
Liu et al. [13] attack the problem of generating large amount of training data and the common problem of hand-crafted features on various texture-less and surface-smooth objects. A hand-crafted 3D feature descriptor with center offset and pose annotations, called view-specific local projection statistics (VSLPSs), is proposed supported by a voting strategy to transform the feature-point matching problem into the problem of voting an optimal model-view in the 6-DOF space. Various experiments on three public and their own dataset demonstrate its good performance.
The application of HMMs for object recognition is restricted to such cases where statistically ordered sequences of features can be constructed. It is natural that in some activities the order of motion patterns determines the class of activity (see, for example, [1]), but in case of static objects the sequence of observed features is not easy to generate. An early example for static objects is in [9] where the contours of objects were used with some invariant features. Unfortunately, in the experiments only four objects were investigated. In [8], authors presented an approach for face recognition using singular value decomposition (SVD) to extract relevant face features and HMMs as classifiers. In order to create the HMM model, the two-dimensional face images had to be transformed into one-dimensional observation sequences. 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 and the proposed spatial order modeling, to represent structural relations, seems to be unnatural. Considering lightweight approaches and HMMs, we should mention

[10], where first the low-dimensional spatial domain feature (SDF) descriptors were clustered and then used as state representation of objects views in HMMs. Unfortunately, no information is given how transition probabilities were estimated.

A recent CNN-based framework can be found in [12], where six-degree-of-freedom (6-DoF) pose for a large number of object classes was determined from single or multiple views. To learn discriminating pose features, three new capabilities are integrated into a deep CNN: an inference scheme that combines both classification and pose regression based on a uniform tessellation of the special Euclidean group in three dimensions (SE(3)), the fusion of class priors into the training process via a tiled class map, and an additional regularization using deep supervision with an object mask. While it is shown that the proposed framework improves the performance of the single-view network, the incremental-temporal use of the network is not discussed.

LSTMs are efficient techniques for the sequential linkage of observation data. In computer vision, they are mostly utilized for the processing of dynamically changing data such as motion behavior [21] and tracking of objects [11]. Not only temporal data can be processed by LSTMs: In [22], apple diseases and pests are detected. Here, the purpose of LSTMs was to combine the features of three deep models, namely AlexNet, GoogleNet, and DenseNet201. Yuan et al. [24] apply a much more interesting approach to address action-driven weakly supervised object detection. The proposed temporal dynamic graph LSTM architecture recurrently propagates the temporal context on a constructed dynamic graph structure for each frame. That is, temporal action information pattern can help the recognition of visual objects. Similarly to our approach [15] combines the output of independent detection but not with HMMs but with LSTMs called Association LSTM.

3 The proposed method

The overview of our algorithm is shown in Figs. 1 and 2. An object is being captured by several shots from different directions. These different views can be also considered as different relative poses. As the camera moves, we get the change of relative pose ($\Delta\alpha_i$) by the IMU sensors. Each captured image is independently evaluated, and the probability of all possible objects is estimated. Analyzing the retrieval list, we can compare the most relevant candidates and can determine the next move of the camera to minimize the ambiguity. To combine the retrieval lists, we start from the assumption that we see the same object continuously. That is, we have to determine the most probable state sequence for each object with the Viterbi algorithm. The object having the highest similarity (when evaluating its highest probability state sequence) is retrieved as the recognized object.

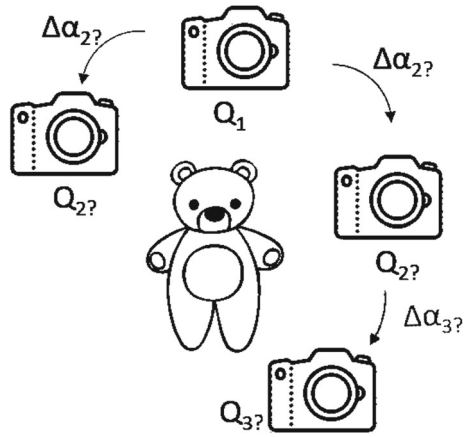

Fig. 1 Object is recognized continuously in a sequence of queries. New queries (Q_2 and Q_3) are planned by the analysis of previous shot(s) (Q_1)

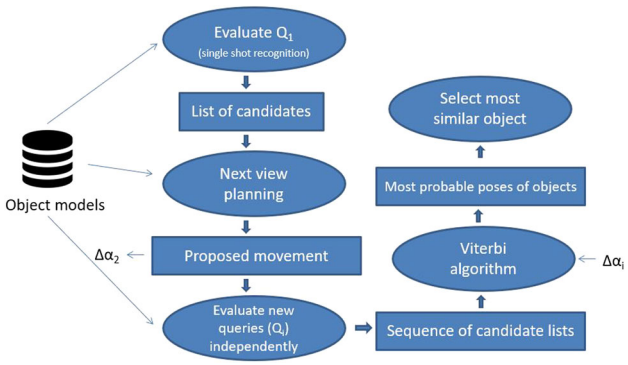

Fig. 2 Overview of the proposed multiview method

3.1 HMM object models

An HMM is defined by:

- the set of N possible hidden states $S = \{S_1, ..., S_N\}$,
- transition probabilities between states S_i and S_j (see Eq. 2),
- emission probabilities based on observations ($P(o)$, see Eq. 7),
- initial state probabilities (π_i).

The observation of objects with multiple views is a process where in each tth step this model is in one $q_t \in S$ state, where $t = 1, ..., T$. To achieve object retrieval will need to build HMM models for all elements of the set of objects (M) where the states correspond to different poses. Then, based on the sequence of observations, we find the most probable state sequence for all object models.

3.2 Object views as states in a Markov model

In our approach, the states can be considered as the 2D views (poses) of a given object model. Observations of these (hidden states) can be easily imagined as the camera is targeting toward an object from a given elevation and azimuth. In our experiments, we use static subdivision of the circle of 360° into 8 uniform sectors 45° each at the same elevation. We define the initial state probabilities $\boldsymbol{\pi} = \{\pi_i\}_{1 \le i \le N}$ based on the opening angle of the views:

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

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

3.3 State transitions

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

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

where i and j indexes refer to hidden states of the HMM. The transition probability matrix is denoted by $\mathbf{A} = \{a_{ij}\}_{1 \le i,j \le N}$, where $a_{ij} \ge 0$, and for a given state $\sum_{j=1}^{N} a_{ij} = 1$ holds.
Building a hidden Markov model means the definition of hidden states and learning its parameters ($\boldsymbol{\pi}$, $\mathbf{A}$, and emission probabilities introduced later) by examining typical examples. However, our problem does not allow such a training process: The probability of going from one state to another severely depends on the user's behavior. Contrary, we 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 (o_t and o_{t-1}):

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

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

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

where $S_i^{\min}$ and $S_i^{\max}$ denote the two (left and right) terminal positions of state S_i. The back-projected aperture interval angle is the range of orientation from where the previous observation could originate:

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

Now, to define the transition probability of coming from state S_i, we compute the ratio of opening angles of the intersection L_j and R_j and of the opening of L_j:

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

3.4 Recognition of objects from multiple views

The emission probability of a particular observation o_t for state S_i is defined as:

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

In [7], we have shown that the area-based CEDD [2] is a robust low-dimensional descriptor for object recognition. CEDD classifies pixels into one of six texture classes (horizontal, vertical, 45° and 135° diagonal, non-edge, and non-directional edges) using the MPEG-7 Edge Histogram Descriptor. For each texture class, a normalized 24-bin color histogram is generated, where each bin represents colors obtained by the division of the HSV color space. CEDD's advantage is that it uses only a short vector (length of 144) as a global descriptor, but naturally it can be less robust under various circumstances. More sophisticated (but also computationally expensive) single shot recognition techniques can also be used within our framework such as SSD [14] or Yolo [17]. In our experiments, we use the combination of CEDD and Tanimoto coefficient to approximate the emission probabilities of states. Emission probability of Eq. 7 can be given as:

$$b_i(o_t) = \frac{\mathcal{T}(\mathcal{C}(S_i), \mathcal{C}(o_t))}{\sum_{j=1}^{N} \mathcal{T}(\mathcal{C}(S_j), \mathcal{C}(o_t))} \tag{8}$$

where $\mathcal{C}$ stands for the CEDD descriptor generating function and $\mathcal{T}$ stands for the Tanimoto coefficient. Since each state of the object models 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. The sequence of retrieval lists, generated by independent queries, is evaluated by the Viterbi algorithm to combine the values of Eqs. 1, 6, and 8 to get the most probable state sequences. 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. To select the winner object $\hat{k}$, we have to compare the observations with the most probable state sequence:

$$\hat{k} = \arg\max_{\forall k \in M} \left(\frac{\sum_{i=1}^{T} \mathcal{T}(\mathcal{C}(o_i), \mathcal{C}(\hat{S}_{k,i}))}{T} \right) \tag{9}$$

where k denotes object k in M.

3.5 Active recognition

Active recognition is a relatively old idea in pattern recognition, and it is typical to extend non-active methods. Without discussing such techniques, we refer the reader to the survey in [18]. Active vision systems can be classified, according to their next view planning strategy, into two groups:

1. Systems that take the next view to minimize an ambiguity function;
2. Systems incorporating explicit path planning algorithms.

We have chosen the first strategy, and here we discuss a method that is very close to human's behavior to move around an object to become acquainted with its appearance from different directions. Based on a rapid evaluation of the first observations, we hypothesize which objects have high probability and we plan the following movements to find those views that can reduce ambiguity. Now, based on the preliminary models, each object k will be represented with N_k descriptors computed as the average of descriptors within a given viewing range:

$$\tilde{c}_{k,i} = 1/N_k^i \sum_{l=1}^{N_k^i} c_{k,l} \tag{10}$$

where $c_{k,l}$ stands for the descriptors of object k within interval i. The similarity between these average views can be computed with the Tanimoto coefficient and can be stored in matrix S of size $NN_k \times NN_k$. After making the very first observations, we are to evaluate the retrieval list(s) $\mathcal{L}$, and as $\alpha(\tilde{c}_{k,i})$ provides the estimate of orientation for the most probable object k in state i, we can also compute the similarity of object views to the left (and to the right accordingly):

$$S_{\text{left}} = \sum_{\tilde{c}_j, \tilde{c}_l \in \mathcal{L}, j \neq l} \mathcal{T}(\tilde{c}_{j,\text{left}}, \tilde{c}_{l,\text{left}}) \tag{11}$$

where $\tilde{c}_{j,\text{left}}$ is the closest $\tilde{c}_j$ view left to $\alpha(\tilde{c}_{j,k})$ being in the already existing retrieval list $\mathcal{L}$. Finally, we should move the camera either to the left or to the right depending on the similarity of views of the possible candidates:

$$\text{Decision} = \begin{cases} \text{Move to left} & \text{if } S_{\text{left}} \leq S_{\text{right}} \\ \text{Move to right} & \text{if } S_{\text{left}} > S_{\text{right}} \end{cases} \tag{12}$$

resulting in the more discriminating direction. The performance of this *active* approach will be compared to the *non-active* recognition in Sect. 5.

4 Datasets

The following datasets were used to generate the HMM models and to run the different tests. All of our models consisted of 8 states.

4.1 Coil-100 dataset

The COIL-100 dataset 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 Fig. 3. To simulate real-life scenarios, we created the Occluded COIL-100 dataset containing the same 100 objects, but with partial occlusion over the object areas (for illustration see Fig. 3).

4.2 ALOI-1000 dataset

The ALOI-1000 dataset includes 1000 different small objects recorded against a black background. Each object was recorded by rotation in the plane at 5° steps. For evaluation under different conditions, we used the same distortion settings, including occlusions, as described for the COIL-100

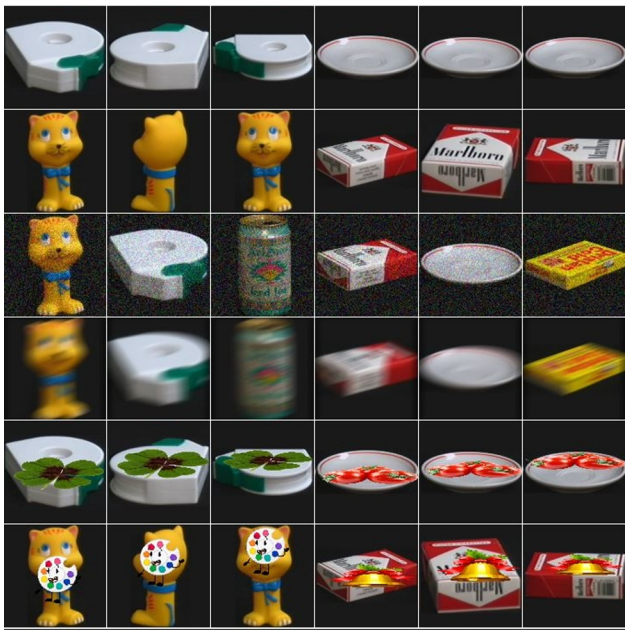

Fig. 3 First two lines: clear samples from COIL-100. Third line: example queries loaded with Gaussian additive noise. Fourth line: example queries loaded with motion blur. Fifth and sixth lines: occluded examples

Fig. 4 First two lines: clear samples from ALOI-1000. Third line: example queries loaded with Gaussian additive noise. Fourth line: example queries loaded with motion blur

dataset. Please note that while the resolution of images in COIL is 128 × 128, it is 384 × 288 for ALOI. (This explains the less visible Gaussian noise and motion blur in Fig. 4.)

5 Experiments and evaluations

All the above-introduced variations in the datasets were generated to show how our temporal methods can improve the performance of the weak classifiers under different circumstances. Since CEDD mainly relies on edge-like features, strong additive noise or (motion) blur can influence results. Charts are generated by taking the average of 10 experiments with randomly generated queries with all 100 and 1000 objects of COIL and ALOI datasets. (That is, the total number of queries was the multiple of 11,000.)

In all measurements, we see the advantage of using multiple queries: all curves monotonically increase as the number of queries increases. The first two charts of Fig. 5 help the understanding of our idea for active vision on some test data where all queries were occluded. The continuous curves show whether the ground truth (GT) objects are within the top 10 candidates of the retrieval lists ($\mathcal{L}$). Since our next view planning makes its decision based on the 10 most probable candidates of the first two retrieval lists, we expect to get results below this curve but above the non-active approach. We could measure performance gain over non-active recognition between 6.2 and 13.8% in these experiments.

Figures 6 and 7 show other experimental results regarding the COIL and ALOI datasets, respectively. In these tests, either all queries were loaded with Gaussian noise (GN) or motion blur (MB), or the first two queries were partially occluded, while the remaining ones were loaded with MB and GN (these are denoted with 2O_MB and 2O_GN). In all cases, the increase of the number of queries resulted in higher hit-rate and active vision outperformed the non-active.

For an alternative presentation of some parts of the above data, we included a table (Table 1) of results for the 8 queries

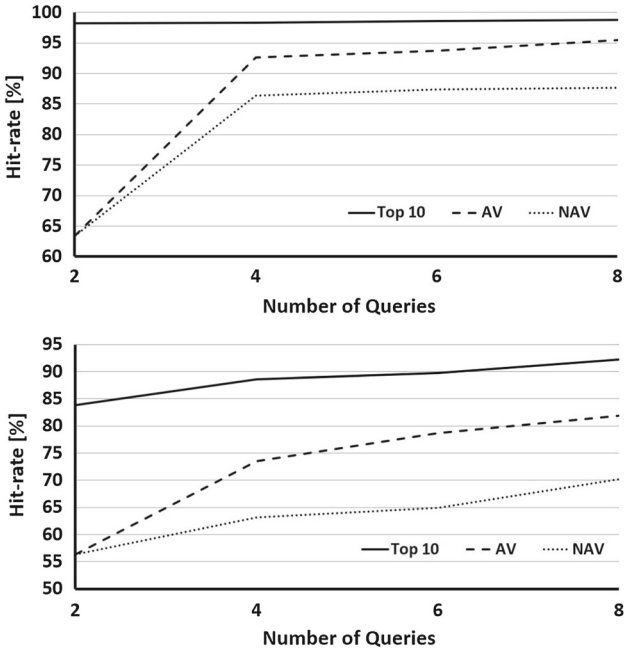

Fig. 5 Comparison of non-active and active recognition when all queries are occluded. Top graph: COIL-100, bottom graph: ALOI-1000 datasets. Continuous curves show the GT being in the top 10 items of the retrieval list

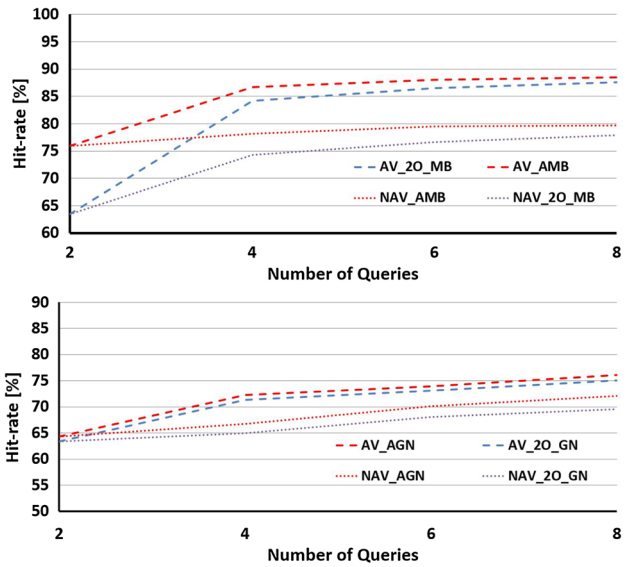

Fig. 6 Comparison of active and non-active recognition on the distorted COIL dataset. First graph: motion blur, second graph: Gaussian noise

cases. It is clear to see that while in case of the smaller dataset (COIL-100 with 100 object classes) there is a significant advantage of the active method, contrary, in case of large number of object classes, this evaporates to around 1% in general. While this effect is natural, it is less significant in case of good quality images as we can read from Fig. 5 where only occlusion happened but no other type of noise.

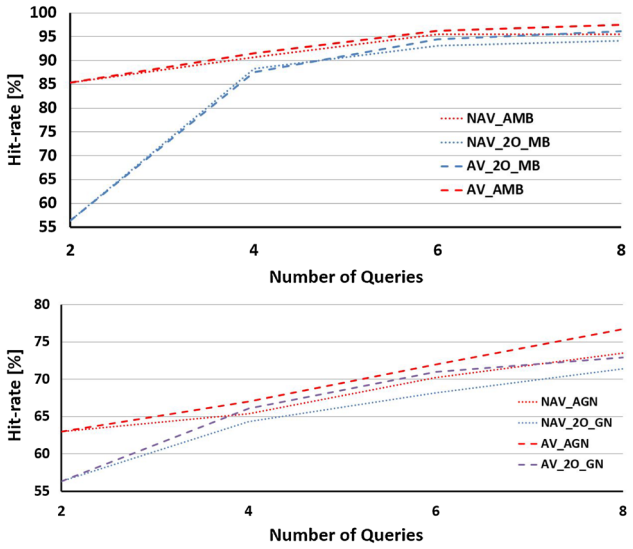

Fig. 7 Comparison of active and non-active recognition on the distorted ALOI dataset. First graph: motion blur, second graph: Gaussian noise

Table 1 Average hit-rates (%) in case of different query distortions applying 8 sequential queries

	COIL-100		ALOI-1000	
	AV	NAV	AV	NAV
All queries occluded (AO)	95.5	87.7	81.9	70.2
2 qrs. occ. oth. GN (2O_GN)	75.1	69.6	72.9	71.4
All queries GN (AGN)	76.1	72.1	76.7	73.5
2 qrs. occ. oth. MB (2O_MB)	87.6	77.9	96.2	94.2
All queries MB (AMB)	88.5	79.7	97.5	95.5

Table 2 Memory and running time requirements of the HMM and LSTM models

Number of queries	2	4	6	8
Method	HMM			
Memory	30 KB	60 KB	89 KB	120 KB
Time (s)	0.0127	0.0288	0.0385	0.0512
Method	LSTM			
Memory	392 MB	787 MB	1.2 GB	1.6 GB
Time (s)	0.0263	0.0422	0.0598	0.0751

5.1 About space and time complexity

While any single shot feature extraction technique can be applied in the proposed framework, in our article we used the very compact CEDD descriptor. It occupies 144 bytes per image, while the orientation information requires not more than 4 Bytes. Running on plain CPUs (Intel Core i7), the memory and running time requirements are given in Table 2.

5.2 Comparison to LSTM

Since DNNs are known for high performance in object recognition, we implemented a so-called ConvLSTM network accepting several query frames based on the technique given in [23]. The overview of the framework, after our modification, is illustrated in Fig. 8. It can process query frames in a directional sequential order (either left or right), the 10 convolution kernels have size 3 by 3. It is known that DNNs are sensible for the training: High number of sample images are required under similar viewing conditions to those at inference. To accomplish this, either sophisticated augmentation techniques are required or large synthetic datasets are used relying on the CAD model of the objects. In our experiments, we trained the LSTM on the whole COIL dataset and tested its recognition performance on the partially occluded version. We already showed the hit-rates of our proposed framework in Fig. 5. For comparisons with LSTM, Fig. 9 shows the mAP (mean average precision) values. It can be interpreted that the HMM can handle much better the untrained occluded queries. The running time and memory usage of the LSTM model are given in Table 2. Please also consider that the training took about half an hour for 100 epochs with an NVIDIA Quadro P6000 GPU with 24 GB RAM.

6 Summary

The main contribution of our paper is to show how active perception and information fusion can help the recognition

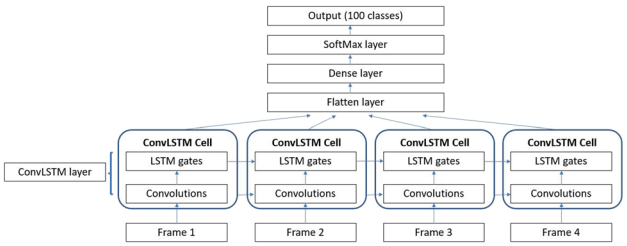

Fig. 8 Overview of the tested ConvLSTM framework in case of four sequential queries

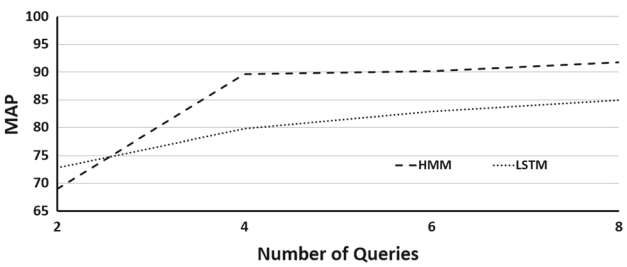

Fig. 9 Comparison of active HMM and ConvLSTM on occluded queries

of 3D objects in a HMM framework if only weak classifiers are applied. The possible application of such approaches can be important in embedded systems or if sensors with limited resources are to be used, for example, in future's autonomous or IoT devices. The proposed HMM technique is computationally lightweight, requires limited memory, and can incorporate other classifiers, not only the presented CEDD. The effectiveness of the method was tested with large number of experiments in various conditions and with comparisons with an LSTM implementation.

Acknowledgements Open access funding provided by University of Pannonia.

References

1. Ahmad, M., Lee, S.W.: HMM-based human action recognition using multiview image sequences. In: International Conference on Pattern Recognition, pp. 263–266. IEEE (2006)
2. Chatzichristofis, S.A., Zagoris, K., Boutalis, Y.S., Papamarkos, N.: Accurate image retrieval based on compact composite descriptors and relevance feedback information. Int. J. Pattern Recognit. Artif. Intell. **24**(02), 207–244 (2010)
3. Chiu, H.P., Kaelbling, L.P., Lozano-Pérez, T.: Virtual training for multi-view object class recognition. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
4. Cupec, R., Vidović, I., Filko, D., Durović, P.: Object recognition based on convex hull alignment. Pattern Recognit. **102**, 107199 (2020)
5. Cyr, C.M., Kimia, B.B.: A similarity-based aspect-graph approach to 3D object recognition. Int. J. Comput. Vis. **57**(1), 5–22 (2004)
6. Czúni, L., Rashad, M.: The fusion of optical and orientation information in a Markovian framework for 3D object retrieval. In: International Conference on Image Analysis and Processing, pp. 26–36. Springer (2017)
7. Czúni, L., Rashad, M.: Lightweight active object retrieval with weak classifiers. Sensors (2018). https://doi.org/10.3390/s18030801
8. Dinkova, P., Georgieva, P., 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), Lisbon, Portugal, pp. 144–149 (2014)
9. Hornegger, J., Niemann, H., Paulus, D., Schlottke, G.: Object recognition using hidden Markov models. Mach. Intell. Pattern Recognit. **16**, 37–44 (1994). Elsevier
10. Jain, Y.K., Singh, R.K.: Efficient view based 3-D object retrieval using hidden Markov model. 3D Res. **4**(4), 5 (2013)
11. Kulkarni, P., Mohan, S., Rogers, S., Tabkhi, H.: Key-track: A lightweight scalable LSTM-based pedestrian tracker for surveillance system. In: International Conference on Image Analysis and Recognition, pp. 208–219. Springer, Cham (2019)
12. Li, C., Bai, J., Hager, G.D.: A unified framework for multi-view multi-class object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 254–269 (2018)
13. Liu, H., Cong, Y., Sun, G., Tang, Y.: Robust 3-d object recognition via view-specific constraint. IEEE Trans. Syst. Man Cybern. Syst., (2020). https://doi.org/10.1109/TSMC.2020.2965729
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
15. Lu, Y., Lu, C., Tang, C.K.: Online video object detection using association LSTM. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2344–2352 (2017)
16. Ng, J., Gong, S.: Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In: Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No. PR00378), pp. 14–21. IEEE (1999)
17. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv:1804.02767 (2018)
18. Roy, S.D., Chaudhury, S., Banerjee, S.: Active recognition through next view planning: a survey. Pattern Recognit. **37**(3), 429–446 (2004)
19. Savarese, S., Fei-Fei, L.: Multi-view object categorization and pose estimation. In: Cipolla, R., Battiato, S., Farinella, G.M. (eds.) Computer Vision, pp. 205–231. Springer, Berlin (2010)
20. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3D object classes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1247–1254. IEEE (2009)
21. Tang, J., Shu, X., Yan, R., Zhang, L.: Coherence constrained graph LSTM for group activity recognition. IEEE Trans. Pattern Anal. Mach. Intell., (2019). https://doi.org/10.1109/TPAMI.2019.2928540
22. Turkoglu, M., Hanbay, D., Sengur, A.: Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. J. Ambient Intell. Humaniz. Comput. (2019). https://doi.org/10.1007/s12652-019-01591-w
23. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
24. Yuan, Y., Liang, X., Wang, X., Yeung, D.Y., Gupta, A.: Temporal dynamic graph LSTM for action-driven video object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1801–1810 (2017)