EFFECTIVE FISHER VECTOR AGGREGATION FOR 3D OBJECT RETRIEVAL

Jean-Baptiste Boin[⋆], *André Araujo*[⋆], *Lamberto Ballan*[⋆†], *Bernd Girod*[⋆]

[⋆]Department of Electrical Engineering, Stanford University, CA
[†]Media Integration and Communication Center, University of Florence, Italy

ABSTRACT

We formulate the task of 3D object retrieval as a visual search problem where a database containing videos of objects captured manually from different viewpoints is queried using a single image. We propose to aggregate visual information of similar views and use the Fisher vector (FV) framework to compactly represent a database of objects. Large-scale experiments on an existing video dataset that we complemented with image queries, shows that our aggregation schemes significantly outperform standard retrieval techniques. When representing our database with only 4 FVs per object, our approach performs with a mean average precision (mAP) of 73.0% on our dataset while the baseline (no aggregation) only reaches a mAP of 43.8%. It can also reach a 72.0% mAP level with a 10× smaller database than the baseline.

Index Terms— Image retrieval, Visual search, 3D object retrieval, Fisher vector aggregation

1. INTRODUCTION

Real-world objects that we interact with in the everyday life, such as a sculpture in a museum or a product in a mall, are intrinsically 3D. Mobile visual search methods [1, 2, 3, 4], and commercial applications such as Google Goggles [5], offer the capability to automatically identify the objects. They typically compare a query image captured by the camera against a large database of images of annotated objects. However, the appearance of an object with complex geometry varies considerably when the viewpoint changes, which makes the problem of 3D object retrieval from a single image query extremely challenging. Previous work in the area does not take advantage of the 3D nature of the data [1, 2, 6], and so they usually fail when query and database images are taken from significantly different viewpoints.

On the other hand, there is a vast literature on model-based and shape-based 3D object retrieval [7, 8, 9, 10, 11, 12]. In this line of work, the goal is to design 3D-shape descriptors and retrieval schemes which are able to effectively retrieve a particular 3D model from a given query. Their main assumption is to have access to accurate scans of the object at both train and query time. This is not a trivial process and the data acquisition is usually performed in a very controlled setup. Some recent approaches attempt to introduce a less constrained scenario by estimating the object pose and selecting query views [13, 12, 14], thus allowing also hybrid 2D-3D retrieval schemes. However, although they require less constraints on the query side, the database collection still needs accurate and controlled object scans. Thus it is very difficult to scale up the process to large collections of objects.

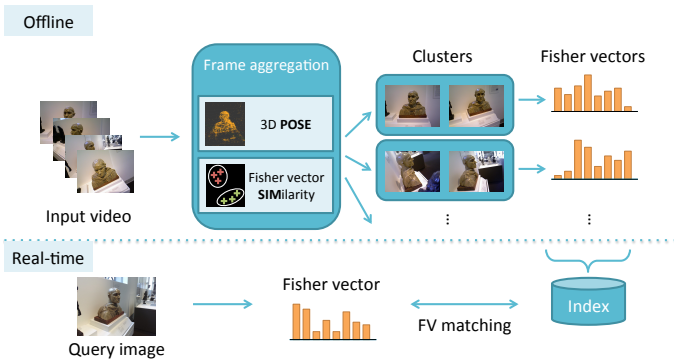

Fig. 1: Overview of the indexing and retrieval process. The frames of the video for each object that we want to recognize are clustered using a predefined aggregation scheme. A FV is then generated from each cluster, which makes the representation very compact. The database can be queried with a single image by generating its FV and matching it against the FVs in the index. Our contribution is to introduce aggregation schemes that achieve good retrieval performance despite high compression factors (POSE and SIM).

In this work, we try to get the best of both worlds. First, our approach targets the 3D object retrieval problem in the context of mobile and large-scale visual search. It is thus important to represent the data associated to each object in a compact way [1, 4]. To this end, we rely on the Fisher Vector (FV) framework that has been shown to efficiently query a large database of images or videos [15, 16, 17, 18, 19]. Second, our framework enriches the image representation such that the different poses of the 3D object are well encoded. That is partially inspired by the high performance obtained in 2D-3D matching and registration for recognizing landmarks [20, 21] or large (city scale) scenes [22, 23, 24]. In contrast to previous works, we study the specific 3D object retrieval scenario in which a precise 3D scanning rig is not available and we just use noisy video scans (i.e. RGB videos). This is an important aspect at a large scale, where the data collection might be crowdsourced, and so it is desirable that the object videos be captured manually in an unconstrained fashion.

Our key technical contribution is to generate a compact

signature of a video sequence that aggregates the features coming from coherent object poses. We evaluate our two proposed aggregation methods and compare them with three others in large-scale experiments. Retrieval experiments are performed by using an existing video dataset [25], enhanced with a new set of query images. When representing our database with only 4 FVs per object, we show that our proposed approaches perform with a mean average precision (mAP) of 73.0% and 72.7%, respectively, while the baseline (no aggregation) only reaches a mAP of 43.8%. Moreover, we can reach a 72.0% mAP level with a $10\times$ smaller database than the baseline. This is particularly important in a mobile visual search scenario where all or part of the database is expected to be stored on the device.

2. POSE-AWARE AND SIMILARITY-BASED FISHER AGGREGATION

Traditionally, mobile visual search (MVS) systems have enabled retrieval of objects that are near-planar or have one canonical view, such as book covers, and that can be represented in the database using a single image. Here, the focus is on using a single image query to retrieve objects with a more complex 3D structure, like sculptures. The retrieval system has access to different views of the object at indexing time.

A naive retrieval system would directly build upon traditional MVS techniques, by storing each different view of the object in the database and indexing them independently. At query time, the system would use the image query to search for the best match against all views for each object. The drawback of this approach is that the database is required to store a large number of items, which leads to scalability problems. In this work, we address this issue by combining different views to construct a much more compact representation of the object.

In our setup, the objects are captured in a semi-controlled way: the annotators were instructed to collect a short video to represent an object of their choice. The collection process used simple handheld cameras, and each object was captured in a different environment.

The system architecture is illustrated in Fig. 1. The main focus of this work is on the frame aggregation. We show that the choice of aggregation scheme is critical in order to get higher retrieval performance.

POSE: *pose-aware aggregation.* We propose to aggregate different object views such that those with similar camera positions are clustered together. Intuitively, this leads to a robust aggregation technique. For example, consider the case where the query image presents an object view which does not exist in the database. It is likely that the query view is similar to some database views which are captured around nearby camera positions. By aggregating database views that are captured in the same region of the 3D space, we incorporate many visual elements that could be visible at these positions or nearby positions.

Due to the semi-constrained collection process, the camera positions were not pre-defined, so it is necessary to estimate them automatically from the different object views. On the other hand, the estimates might not need to be perfect: since our objective is to use the camera poses to select which views are aggregated together, a rough estimate might be sufficient.

In the offline stage, we start by using a standard structure from motion (SfM) algorithm, by Snavely et al. [26], to compute a 3D model of the object and find the camera positions. We do not have access to the cameras that were used to collect the data so we estimate the camera intrinsic parameters by using the estimate of the field of view from the nominal specifications. This is sufficient under some reasonable assumptions (no lens distortion, no skew, principal point located at the center of the image). In this process, some views may not be properly registered so an estimate of the camera positions cannot be obtained. This may happen if the view is too dissimilar to the rest of the sequence, for example due to motion blur or lack of texture. Such views, called invalid, were discarded — we consider that they did not provide relevant information.

After obtaining an estimate of the camera extrinsic parameters (position and rotation), the valid views are clustered into K clusters by using their estimated 3D camera positions and the K-means algorithm. Finally, the views in each cluster are aggregated into a global signature which describes all of these different views. We make use of Fisher vectors (FV) [15], a state-of-the-art global image descriptor. In this case, the local features from all of the different views are pooled together to generate one FV that efficiently represents all of these views. We call this aggregation technique **POSE**.

SIM: *similarity-based aggregation.* Although the SfM stage makes intensive use of the underlying structure of the data (different views of a fixed object captured by a mobile handheld camera), it introduces additional complexity and strongly relies on a successful SfM system. For this reason we design an alternative version of this algorithm that we call **SIM** that aims at aggregating views based on visual similarity. The process is as follows: we first extract FVs from all views independently and cluster these high-dimensional vectors using K-means. This generates K sets of views that we aggregate into one FV per set.

3. EXPERIMENTS

3.1. A new dataset for query-by-image 3D object retrieval

The new dataset by Choi et al. [25] is particularly fitting for this study. It contains videos of various 3D objects that were captured by different operators, using an unconstrained RGB-D camera. There is only one video per object. We focus on the "sculptures" part of this dataset and we complemented it by capturing our own set of query images. This allow us to test the robustness of our approach to different capture conditions (different mobile device, lighting conditions, etc.). After

duplicate removal, this subset contains 453 videos captured at a resolution of 640 × 480 and a frame rate of 30 FPS. In this work, the depth channel of these videos was ignored. We captured queries for 35 of these objects, taking 5 photos per object with a mobile device camera, for a total of 175 queries[1].

3.2. Experimental protocol

FV parameters. Local features are extracted using a Hessian-affine detector [27] and described with SIFT descriptors [28]. These 128D descriptors are then projected to a 32D space using PCA, as in [18, 29, 30]. When generating the FVs from these descriptors, we use 512 Gaussians, as in previous work [18, 29]. We also evaluate results using 256 and 128 Gaussians to study whether our findings are also valid for more compact — but lower accuracy — representations. We use state-of-the-art binarized FVs [30], which were selected by the MPEG Compact Descriptors for Visual Search (CDVS) standardization effort for their scalable retrieval performance.

Indexing and retrieval. For our experiments, we extracted FVs using each of our schemes for various values of K. The frames used at that stage are sampled from the original video sequences at 1 FPS. These FVs correspond to our database-side index. Then, each query image is processed individually and its corresponding FV is matched against all the FVs in the index. The list of FVs is then sorted by weighted Hamming distance [30] to generate a ranked list of object. Retrieval performance is assessed using mean average precision (mAP) on that list.

Baselines. We compare the proposed aggregation schemes against three baselines:

INDEP: *individual views.* In this scheme, which is the naive approach mentioned in the previous section, we do not perform feature aggregation from multiple views, but we only generate FVs out of the features from single views independently. When a value of K is given, we select a subset of K views equally spaced (in time) over the whole sequence and generate one FV per view to represent our object.

RAND: *random view aggregation.* We create K sets of views by randomly assigning each frame to one set. We then aggregate all the features from the views of each set into a FV.

TEMP: *temporal view aggregation.* We divide a video sequence into K segments of approximately equal length and we generate FVs from the views of each segment.

It is important to note that for any value of K, RAND, TEMP and SIM use information from all the views of the sequence and POSE uses information from all views considered valid after the SfM stage. INDEP typically uses the least amount of information compared to the other schemes, especially for low values of K since $N - K$ views are ignored (where N is the number of views in the sequence), but the memory requirements and retrieval speed is still the same as the other schemes for equal values of K. A visualization of the clusters associated to each frame of an example sequence is given in Fig. 3.

[1]We will release these queries with publication of this work.

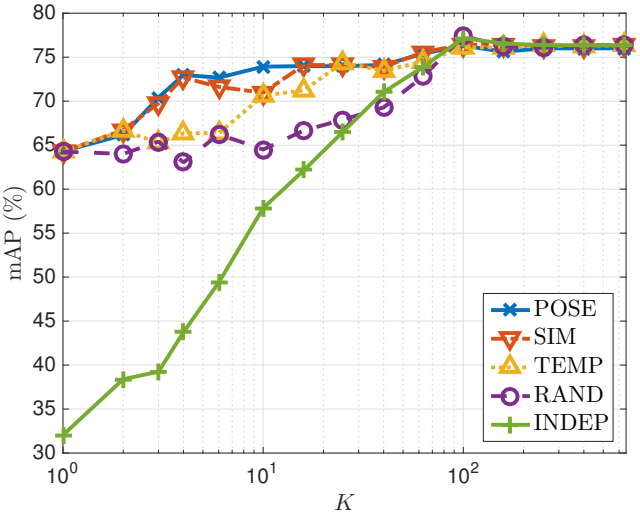

Fig. 2: Retrieval results: mean average precision (mAP) as a function of the number of database signatures per video (K), for different aggregation methods. FVs are generated using 512 Gaussians.

Sampling rate	1 FPS	3 FPS
First quartile	9.91%	52.58%
Median	34.62%	90.15%
Third quartile	78.44%	99.42%

Table 1: Comparison of statistics on the proportion of valid views after the SfM stage when sampling the videos at 1 FPS and 3 FPS. At 3 FPS, half of the videos get more than 90% valid views.

3.3. Experimental results

Structure from motion results. Choosing the frame rate used for the database videos was critical for the SfM stage and the quality of the reconstructions highly depended on that. Processing all the frames in the sequence would give the highest quality reconstruction, because the minimal appearance changes from one view to the next would allow for reliable registration of consecutive views. However, using a very high frame rate is not practical because the complexity considerably increases with the number of input images. On the other hand choosing a frame rate too low is detrimental to the quality of the reconstruction. For these sequences, we found that a sampling rate of 3 FPS was a reasonable trade-off between camera pose quality and processing time. The SfM system can work poorly for some videos, since some of them are very challenging (e.g. videos with low contrast) but at this frame rate we can get a high proportion of valid views for most of them. We compare statistics on this quantity in Table 1.

Retrieval results. We show the results of our experiments using 512 Gaussians in Fig. 2. It is to be noted that by design

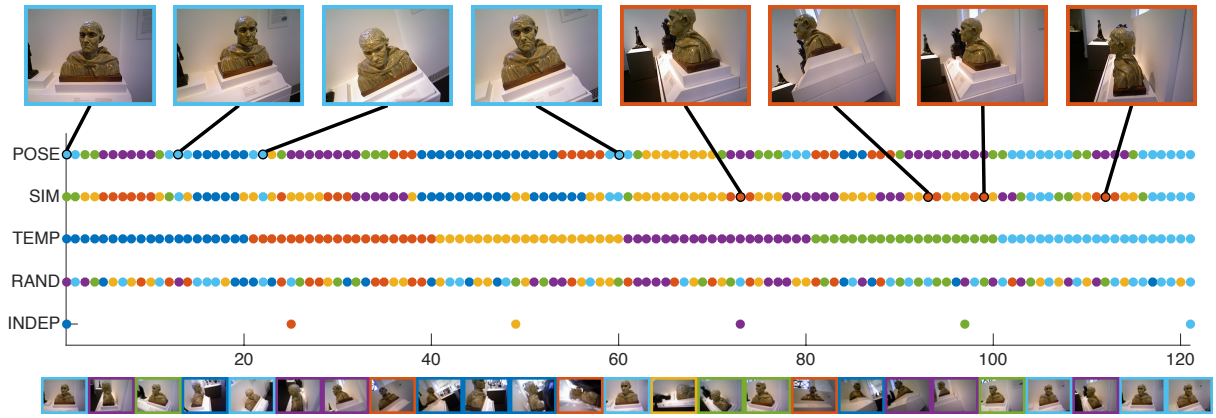

Fig. 3: Example of view aggregation for $K = 6$ on a sequence of 121 frames. Each dot corresponds to one view and the views aggregated together are shown in the same color. The images shown at the bottom represent a subset of the views in chronological order, the outline corresponding to the color of the set each view belongs to in POSE. At the top, we present typical views belonging to the same clusters for POSE and SIM.

the behavior of these schemes is expected to be very similar for high values of K. Indeed, with a large number of clusters, each cluster contains one or a few frames, so performance should converge to the same level.

The goal of this work is to achieve high accuracy with a low value of K, so it is desired to have the retrieval accuracy reach its maximum value as fast as possible. It is clear that aggregating views is an excellent strategy: INDEP performs considerably worse than RAND, TEMP, SIM and POSE for low values of K, and the mAP only becomes comparable with the base values for the aggregated schemes when using more than 10 views. Regarding the other schemes, RAND performs the worst and POSE and SIM deliver the highest performances, with no clear better option among the two.

We illustrate in Fig. 3 how views are clustered for the different schemes within one video. The different clusters obtained with POSE capture very distinct appearances. For example (Fig. 3 bottom) the light blue cluster contains all the frontal views of the object, the purple cluster the ones from the right, etc. The clusters obtained by SIM follow a similar distribution even if the frame assignment looks slightly more noisy. This can be explained by the variation of the set of features when there is motion blur, or if the background is altered, which can considerably alter the FV for a given view. Still, with a similar performance but lower complexity compared to POSE, this scheme could be preferred for some applications: the visual similarity can confidently be used as a proxy for the position if computing the camera poses is impractical.

For $K = 1$, RAND, TEMP and SIM are perfectly equivalent: all views are aggregated together using a single FV. The difference for POSE is that the views considered invalid after the SfM stage are dropped. This only changes the mAP in a negligible way, which confirms that the frames dropped do not contribute to useful information for an object.

The best gains for POSE and SIM compared to the other schemes are obtained for $K = 4$, as reported in Table 2 for different numbers of Gaussians used for the FVs. The mAP for our proposed approaches is typically boosted by 60 to 80% compared to INDEP. The benefits of aggregating views in a non-naive way are also the most striking for this value of K: for all number of Gaussians, aggregating views using POSE or SIM allows for a 8 to 10 p.p. mAP boost compared to the random aggregation. Another way to interpret our results is that our proposed approaches reach a similar retrieval accuracy level compared to other schemes while using a smaller database index: when using 512 Gaussians a mAP of 71% can be achieved with a $10\times$ smaller database for POSE or SIM ($K = 4$) compared to the naive INDEP scheme ($K = 40$). Even TEMP does not reach this accuracy target until $K = 16$, *i.e.*, obtaining 4X more memory-efficient index. This is crucial for mobile visual search applications where part of the database may be stored on a memory-constrained mobile device [4].

Method	128 Gauss.	256 Gauss.	512 Gauss.
INDEP	37.79%	38.75%	43.75%
RAND	56.68%	62.51%	63.04%
TEMP	58.86%	64.34%	66.36%
SIM (ours)	**66.58%**	**70.65%**	72.69%
POSE (ours)	64.63%	69.40%	**72.98%**

Table 2: Best results for a fixed database size constraint. We compare the mAP for a fixed value $K = 4$. Note that the values reported for 512 Gaussians are the same as those reported in Fig. 2.

4. CONCLUSIONS

This work explores the task of 3D object retrieval in the context of mobile visual search, where index compression and retrieval speed are critical. We propose two methods to efficiently represent manually captured videos of objects. Using large-scale experiments, we show that our approaches outperforms other representations based on FV aggregation.

Acknowledgements. L. Ballan is supported by an EU Marie Curie Fellowship (No. 623930).

5. REFERENCES

[1] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 61–76, 2011.

[2] Y. Wang, T. Mei, J. Wang, H. Li, and S. Li, "Jigsaw: Interactive mobile visual search with multimodal queries," in *Proc. ACM Multimedia*, 2011.

[3] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang, "Mobile product search with bag of hash bits and boundary reranking," in *Proc. CVPR*, 2012.

[4] D. Chen and B. Girod, "A hybrid mobile visual search system with compact global signatures," *IEEE Trans. on Multimedia*, vol. 17, no. 7, pp. 1019–1030, 2015.

[5] "Google goggles," https://play.google.com/store/apps/details?id=com.google.android.apps.unveil, 2010, Accessed: 2014-09-07.

[6] R. Arandjelović and A. Zisserman, "Smooth object retrieval using a bag of boundaries," in *Proc. ICCV*, 2011.

[7] D. Y Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung, "On visual similarity based 3d model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[8] A. Del Bimbo and P. Pala, "Content-based retrieval of 3d models," *ACM Trans. on Multimedia Computing, Comm. and Appl.*, vol. 2, no. 1, pp. 20–43, 2006.

[9] J.-L. Shih, C.-H. Lee, and J. T. Wang, "A new 3d model retrieval approach based on the elevation descriptor," *Pattern Recognition*, vol. 40, no. 1, pp. 283–295, 2007.

[10] Y. Gao and Q. Dai, "View-Based 3D Object Retrieval: Challenges and Approaches," *IEEE Multimedia*, vol. 21, no. 3, pp. 52–57, 2014.

[11] M. A. Savelonas, I. Pratikakis, and K. Sfikas, "An overview of partial 3D object retrieval methodologies," *Multimedia Tools and Applications*, vol. 74, no. 24, pp. 783–808, 2015.

[12] J. Xie, F. Zhu, G. Dai, and Y. Fang, "Progressive shape-distribution-encoder for 3d shape retrieval," in *Proc. ACM Multimedia*, 2015.

[13] I. Atmosukarto and L. G. Shapiro, "3d object retrieval using salient views," in *Proc. ACM MIR*, 2010.

[14] T. Furuya and R. Ohbuchi, "Diffusion-on-Manifold Aggregation of Local Features for Shape-based 3D Model Retrieval," in *Proc. ACM ICMR*, 2015.

[15] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Anal. and Machine Intellig.*, vol. 34, no. 9, pp. 1704–1716, 2012.

[16] J. Lin, L.-Y. Duan, T. Huang, and W. Gao, "Robust fisher codes for large scale image retrieval," in *Proc. ICASSP*, 2013.

[17] X. Lu, Z. Fang, T. Xu, H. Zhang, and H. Tuo, "Efficient image categorization with sparse fisher vector," in *Proc. ICASSP*, 2015.

[18] A. Araujo, J. Chaves, R. Angst, and B. Girod, "Temporal Aggregation for Large-Scale Query-by-Image Video Retrieval," in *Proc. ICIP*, 2015.

[19] S. Song, N.-M. Cheung, V. Chandrasekhar, B. Mandal, and J. Lin, "Egocentric activity recognition with multimodal fisher vector," in *Proc. ICASSP*, 2016.

[20] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," in *Proc. ECCV*, 2008.

[21] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *Proc. ICCV*, 2011.

[22] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. CVPR*, 2007.

[23] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. CVPR*, 2009.

[24] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *Proc. ECCV*, 2010.

[25] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," *arXiv:1602.02481*, 2016.

[26] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 835–846, 2006.

[27] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[28] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] D. Chen *et al.*, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Processing*, vol. 93, no. 8, pp. 2316–2327, 2013.

[30] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, "Compact descriptors for visual search," *IEEE Multimedia*, vol. 21, no. 3, pp. 30–40, 2014.