How Many Neighbours for Known-item Search?

Jakub Lokoč and Tomáš Souček

SIRET Research Group, Department of Software Engineering
Faculty of Mathematics and Physics, Charles University, Czech Republic
lokoc@ksi.mff.cuni.cz, tomas.soucek1@gmail.com

Abstract. In the ongoing multimedia age, search needs become more variable and challenging to aid. In the area of content-based similarity search, asking search engines for one or just a few nearest neighbours to a query does not have to be sufficient to accomplish a challenging search task. In this work, we investigate a task type where users search for one particular multimedia object in a large database. Complexity of the task is empirically demonstrated with a set of experiments and the need for a larger number of nearest neighbours is discussed. A baseline approach for finding a larger number of approximate nearest neighbours is tested, showing potential speed-up with respect to a naive sequential scan. Last but not least, an open efficiency challenge for metric access methods is discussed for datasets used in the experiments.

Keywords: Similarity search · Known-item search · Data indexing.

1 Introduction

Deep learning here, deep learning there, deep learning everywhere! Words that have come to mind of a multimedia retrieval researcher since 2012. Besides other retrieval challenges, similarity search [5, 29, 6] has also been significantly affected by the impressive deep learning paradigm [9]. The cornerstone of the general similarity search approach, similarity space (U, σ) consisting of a descriptor universe U and a similarity measure σ, started to be narrowed to "just" vector spaces with a cheap bin-to-bin similarity measure used during a deep model training process. In other words, a similarity of any multimedia data objects x, y is now often modeled with a cheap similarity function (usually linear time complexity) evaluated for their vector representations $v_x, v_y \in R^n$ obtained from a deep model[1]. Regardless of deep learning trends, there still exists a need for querying a large database for similar objects to a query object, assuming database objects are mapped to descriptors $S \subset U$. For a query q, applications usually require a set of most similar objects from a multimedia database. Assuming a popular approach to model similarity with a distance function δ, two popular similarity queries are $range(v_q, \theta) = \{v_o \in S | \delta(v_q, v_o) < \theta\}$, and k nearest neighbours query $kNN(v_q, S) = \{X \subset S : |X| = k, \forall v_x \in X, \forall v_y \in S - X : \delta(v_q, v_x) \leq \delta(v_q, v_y)\}$.

[1] In the following text, we follow the notation x, v_x to formally distinguish objects and their descriptors, where descriptors are means of object similarity evaluations.

The aforementioned similarity queries are useful for search needs initiated with a query object addressing the contents of multimedia objects[2]. Usually, users provide either a text query for a text-multimedia cross modal search approach [14, 22], or an example multimedia object. The ultimate problem is whether the provided query is good enough to ensure desired objects in the result set, i.e., in the set of the most similar objects to the query. The problem can be divided to two sub-problems – whether the user can provide a sufficient query object (or detailed text description), and whether the system implements a similarity model consistent with user expectation of similarity between two objects. In this paper, we further consider user need aspects [28], but we keep general formal specification of search needs. Let C be a subset of database objects representing some target class/topic. The search problem complexity differs if users want to find just an arbitrary item $x \in C$ (i.e., high precision is sufficient), or all items from C are required (i.e., high precision and also recall are necessary). Due to potentially high variability of objects in C, it is way more challenging to find all dataset instances of the class.

A special variant of the all-instance search task is for $|C| \to 1$, which corresponds to search need for a very narrow class of objects. In extreme case, only one multimedia object (e.g., image or shot) is required, which is referred to as *known-item search* (KIS). Although unique properties of a single searched object might seem as an advantage for the search engine, users often do not actively remember all the specific details for query formulation. On the other hand, there is an assumption that users can rely on (limited) passive knowledge of the known item when refining and browsing candidate result sets. The passive knowledge can include also a temporal context of the item in the case of video sequences. The complexity of a KIS task depends also on the number of similar dataset objects matching provided (potentially imperfect) query description. For example, searching for some specific scene of a surfing person would be way more easier if there are no other scenes of people surfing in the database. In case there exist near-duplicates (e.g., some small audio-visual transformations of the target object), multiple instances could be considered as the correct result. From this perspective, known-item search can be generalized from $|C| = 1$ to $|C| \geq 1$, but the set consists just of near-duplicate objects satisfying the need for one searched multimedia object. This is the main difference from an ad-hoc search task with a specific narrow search focus, where different objects can fulfill the specification.

In this paper, we argument that known-item search is often very challenging even with a state-of-the-art text-image search model (demonstrated in Section 3). In order to find a searched known item, an interactive search approach [27, 25] is therefore a preferred option as reported by respected evaluation campaigns [16, 10]. In the last decade, several interactive search systems were designed and tested [12, 13, 24, 11, 1, 19]. To deal with a known-item search task, users can either iteratively reformulate queries after unsuccessful inspection of top ranked items (kNN queries with low k), or, use advanced visualization [4], relevance

[2] We consider challenging content-based search cases, where users do not know unique structured attributes (e.g., filename or ID) of searched multimedia objects.

feedback [7] or other exploration methods when top ranked result set inspection fails. With a single query available, a substantial portion of the database has to be considered for inspection to guarantee a higher chance for success. Therefore, finding a larger set of (approximate) nearest objects to a query represents a suitable search step. At the same time, larger numbers of nearest neighbours represent a challenge for query processing methods.

2 Known-item Search

Imagine a large collection of funny videos, where a user wants to find one particular scene which made the user laugh for days. Definitely, the user might want to find this one particular scene again in the future, which would restrict the set of all funny scenes to one searched instance. Another example might be a memory of some experience, captured by a wearable camera to a personal lifelog database [10]. Again, the search need might focus just on the one specific memory. These examples illustrate that known-item search tasks are natural part of the set of possible search needs. The tasks are also well-suited for comparative evaluations [17] and benchmarking as the ground truth is determined by the one searched item (e.g., image or temporal segment), compared to partially unknown ground truth of more generally formulated Ad-hoc search tasks evaluated at TRECVID [3] (KIS tasks used to be evaluated at TRECVID in the past). We note that the discussed near-duplicates might be a missing part of ground truth for KIS tasks as well. Nevertheless, in an automatic evaluation of ranked lists the missing near-duplicates might achieve similar ranks as the available correct objects and also this approximation issue represents a consistent obstacle for all compared methods.

2.1 Problem formulation

Known-item search corresponds to a search scenario, where a user has just a mental picture of an existing multimedia object from a given database. Either the known object has been seen before, or a specific enough description (potentially including hand-drawn sketches) of the object was provided to the user. In the context of this paper, a generalized KIS task can be formulated as:

Definition 1. *Let DB be a multimedia collection, the task is to find one $t \in C_T \subset DB$, where C_T contains one known target object and its near-duplicates differing from the target object by a small audio-visual transformation, negligible for the search need (e.g., different encoding or minor image enhancement).*

For automatic evaluations analyzing ranking of database objects with respect to a query, the top ranked $t \in C_T$ is considered, optimistically assuming that users do not overlook a correct item in a displayed ranked result set. For search needs targeting just a part of a multimedia object (e.g., segment of a video), the definition can be modified by using an appropriate data representation unit.

2.2 Ranking model evaluation

In order to measure known-item search effectiveness of a model (U, δ), a set of pairs $B = \{[q_i, C_{T_i}]\}_{i=1}^{n}$ can be created for a multimedia database DB, where q_i represents a user defined query addressing selected $C_{T_i} \subset DB$ presented in some convenient form to the user in advance. We remind that near-duplicates might be missing in ground truth, which limits objects evaluated as correct. For each query q_i, all database objects $o \in DB$ are ranked with respect to $\delta(v_{q_i}, v_o)$ and the rank r_i of the top ranked object $t \in C_{T_i}$ is stored. Either the average of all ranks r_i can be computed, or an empirical cumulative graph detailing effectiveness for growing rank is reported using

$$F_B(r) = \frac{|\{r_i : r_i \in Ranks, r_i \leq r\}|}{|Ranks|},$$

where $Ranks$ represents all obtained top ranks r_i for all benchmark pairs $[q_i, C_{T_i}]$ and a tested model (U, δ). For example, see the cumulative graph in Figure 1 illustrating the percentage of findable known items when users browse a ranked list up to a rank r, provided that a correct item is not overlooked (which is generally not guaranteed [15]).

3 Experiments

This section presents an evaluation benchmark dataset and several experiments demonstrating challenges of effective and efficient known-item search.

3.1 Known-item search benchmark set

We analyze the performance of two respected text-image search approaches CLIP [22] and W2VV++ [14] (its BERT variant [15]) for a benchmark set comprising 327 pairs $[q_i, C_{T_i}]$, where all sets C_{T_i} are subsets of a 20K benchmark image dataset extracted from the V3C1 collection [23]. The search need (i.e., known item) was represented by one randomly selected image and no near-duplicates were considered during benchmark construction (i.e., $|C_{T_i}| = 1$). Free-form text descriptions (queries) for target images were provided by human annotators. Each annotator observed a target image for the whole annotation time (i.e., perfect memory was assumed). Although the size 20000 objects does not conform to the idea of big data, it might still represent for example a personal image database where known-item search can be expected.

Both CLIP and W2VV++ BERT text-image search approaches provide functions f_{visual}, f_{text} for joint image and text embedding to R^n (n=2048 for BERT, n=640 for CLIP). Using the functions, all database images (including known items) and text queries q_i were transformed to n-dimensional vectors. For ranking of the 20K images with respect to q_i, a similarity model based on $1 - \sigma_{cos}(f_{visual}(o), f_{text}(q_i))$ can be utilized to identify the rank of $t_i \in C_{T_i}$. For

all 327 pairs, Figure 1 shows the performance of both compared models, revealing more effective known-item search performance for the recently released CLIP model. Nevertheless, there were individual benchmark pairs (about 30%) where the CLIP model was outperformed by W2VV++ BERT.

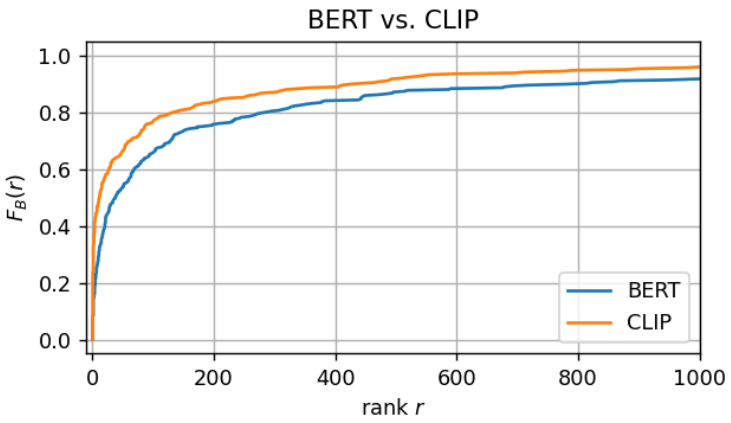

Fig. 1. Performance of text-image search models for the benchmark set, first 1000 ranks.

For both models, it is apparent that finding "just" one or ten nearest neighbours is not sufficient to solve all known-item search tasks even for the relatively small 20K dataset (though the performance of the new CLIP model is impressive!). With 100 nearest neighbours, more than 65% (75% for CLIP) of known items t_i searched by the query q_i would be directly findable in the result set. However, to provide a chance to solve 90% of all tasks by one query, hundreds of nearest neighbours are necessary for the 20K dataset. There also exist queries where even thousands of nearest neighbours are not enough. We emphasize that all the presented numbers are bound to the dataset size, for larger datasets the numbers of necessary nearest neighbours are significantly higher [15].

With the growing number of the nearest objects it becomes way more difficult to find the target with sequential result set browsing. Indeed, known-item search is a challenge that cannot be easily solved with just a single ranked list and scroll bar (at least yet). On the other hand, efficient construction of a larger candidate set is a promising first step that can be followed by a plethora of interactive search approaches. Assuming that the user cannot remember more details to extend/change the query, there are still options to inspect results for text query subsets, provide relevance feedback for displayed set of images, browse images in an exploratory structure, etc. However, these methods are beyond the scope of this paper.

3.2 Upper performance estimates for kNN browsing

Before we proceed to a large candidate set selection study in the next section, we investigate kNN based browsing with small candidate sets to solve KIS tasks for the small 20K dataset. To analyze the search strategy, we run simulations for the 327 benchmark pairs and the W2VV++ BERT model.

Each simulated search session started with a text query q_i. From the result, top ranked k items were selected as a display D_j out of which one item $q_j \in D_j$ was selected as a new query object for the next display presenting $\text{kNN}(v_{q_j}, S)$. This process was repeated until the target item t_i was found or the maximal limit of iterations was reached. The automatic selection (i.e., simulation of user interaction [8, 7]) of the new query considered two optimistic options based on $\text{kNN}(v_{t_i}, D'))$, where $t_i \in C_{T_i}$ is the searched target image and D' are descriptors of images on the current display. We consider an IDEAL user automatically selecting as the new query the most similar object from the display D to the target t_i. In addition, we consider also a randomized TOP user, where the new query object is selected randomly from $\text{kNN}(v_{t_i}, D'))$, $k = 8$. To prevent from cycles, once selected queries q_j were removed from the dataset in a given search session.

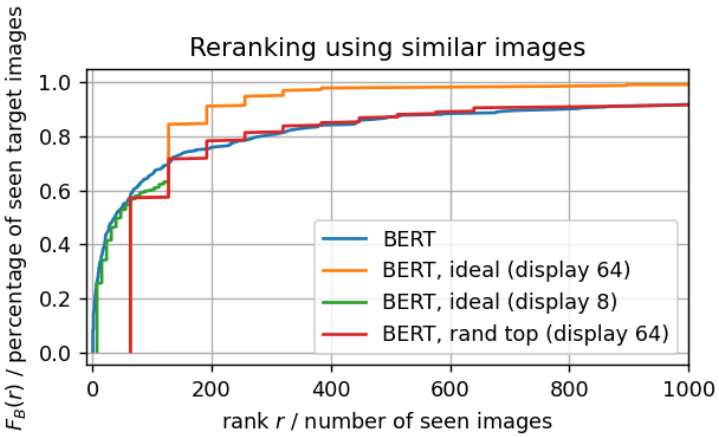

Fig. 2. Browsing simulations using 16 kNN displays, browsing performance ("stairs") related to performance of the W2VV++ BERT text search model (first 1000 ranks).

Figure 2 compares the W2VV++ BERT text search model fine-grained ranking (i.e., browsing the original ranked set) with iterative reformulations providing always 64 nearest objects for one selected query object (IDEAL or TOP) from the current display. For each iteration, the graph shows the increase of solved tasks for the whole display at once (therefore the staircase pattern). For the IDEAL user and $|D| = k = 64$ the kNN browsing would boost the performance

compared to the original ranked set. However, the IDEAL user is too optimistic, real users are not 100% consistent with the similarity model. Furthermore, for smaller display size $|D| = 8$, even the IDEAL user performance is worse than the original ranking. For the randomized TOP user and display size 64, the performance of kNN browsing has a similar performance effect as sequential search of the original ranked list. However, even the randomized TOP user is still rather optimistic, real users may select from a display also less similar items to $t_i \in C_{T_i}$.

To sum up the simulations, kNN based browsing with IDEAL selections and small $k = 8$ is not effective enough, while, for $k = 64$, such browsing would be a competitive strategy with respect to the original ranking. However, to the best of our knowledge selections by real users are usually not ideal which decreases recall gains by the kNN browsing strategy. The effect of less optimal selections is illustrated by the performance drop between IDEAL and TOP users in Figure 2. kNN browsing by the TOP user and $k = 64$ resulted in "just" similar effectiveness as sequential browsing of the original ranked list, where users do not have to select a good example query in each iteration. In other words, top ranked 1000 items for a text query could be browsed directly. For more effective browsing, advanced models based on relevance feedback were proposed [7], maintaining relevance scores for all objects in the database. In order to make the maintenance process more efficient, a larger candidate set can be selected for the models (e.g., 10% of top ranked items guaranteeing 90% of searched items).

3.3 A baseline study for efficient candidate set selection

In order to find top k nearest neighbours in a high-dimensional space efficiently, one popular option is dimension reduction. Figure 3 shows a comparison of dimension reduction techniques [21] for both models CLIP and W2VV++ BERT. We consider principal component analysis with data centering as a first step (PCA) and without centering using only Singular Value Decomposition (SVD). We compare effects of both approaches, provided that PCA might harm data by subtracting mean values to center (normalized) data vectors. The graph shows that reduction of the dimension to 128 does not affect the performance of the BERT variant regardless the reduction technique. However, the benefits of the CLIP model seem to vanish with the dimension reduction using SVD. Furthermore, PCA reduction to 128 dimensions (or even 256) significantly deteriorated the performance of the CLIP model which might be caused by specific properties of the text-image similarity space (see the next section).

Focusing just on the W2VV++ BERT model, Figure 4 presents ranking performance for decreasing number of dimensions selected after SVD. We may observe that up to 64 dimensions, the performance of the model does not deteriorate. In other words, 32 times smaller dataset of descriptors and faster computation can be achieved with a standard pre-processing technique. Furthermore, even lower dimensional versions are useful for approximate search in a filter and refine mode. For the data, 50% of the database can be filtered with the 16 dimensional version of descriptors and the remaining part can be refined

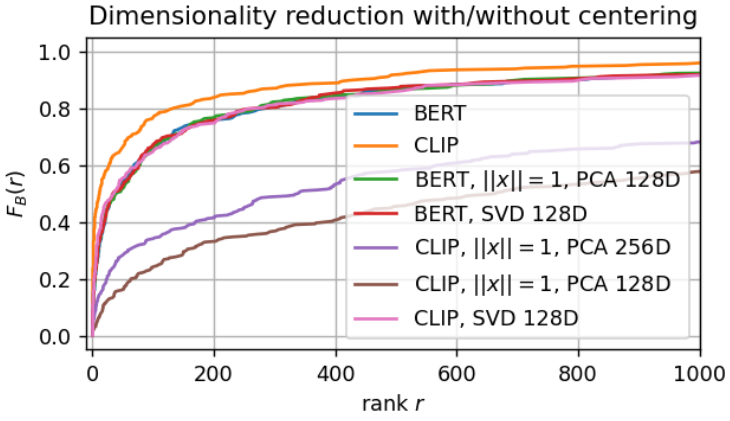

Fig. 3. Comparison of PCA and SVD for first 1000 ranks.

with the 64 dimensional version. This simple approach would reduce computation costs using a bin-to-bin measure like $-\sigma_{cos}(v_x, v_y)$ from $64 \cdot DBSize$ to $16 \cdot DBSize + 48 \cdot DBSize/2$ bin-to-bin operations. Please note that intermediate results for filtering can be re-used for refining and there emerge additional sorting costs for the refined half of the database. Allowing a small drop in recall, approximate 1000 most similar objects could be computed by refining just 10% of the 20K database filtered with 16 dimensional vectors, resulting in $16 \cdot DBSize + 4.8 \cdot DBSize$ bin-to-bin operations. At the same time, the approximate filtering approach still allows easy parallelization of the computation. We note that a bin-to-bin distance function for the first a dimensions of (normalized) data vectors can lower bound the distance for $b > a$ dimensions (e.g., for a similarity model based on squared Euclidean distance $\sum_{i=1}^{a} (v_{x_i} - v_{y_i})^2 \leq \sum_{i=1}^{b} (v_{x_i} - v_{y_i})^2$). Hence an optimal kNN query processing strategy [26] could be tested instead of a fixed hard filter of x% of the database.

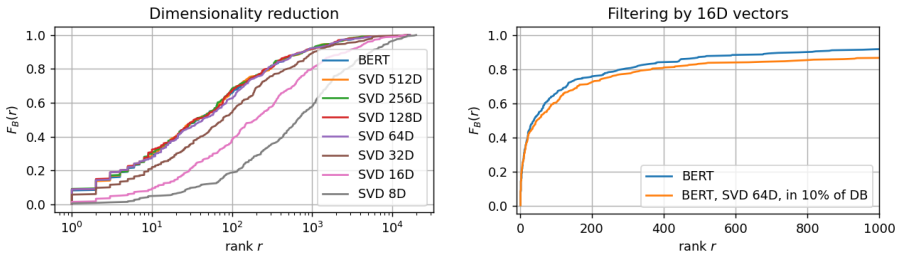

Fig. 4. Performance for decreasing dimensionality of descriptors after SVD. On the right, the effect of refining of 10% of database filtered with 16 dimensional vectors.

4 Is there a room for competitive metric indexing?

Using the single-space-for-all approach (e.g., (R^n, σ_{cos})) for various application domains reminds the motivation of the metric space approach providing one access method for different metric spaces. The question raised in this section is, whether general distance-based metric indexing [29, 6] can provide a competitive approach to methods presented in the previous section. Since metric indexing relies on lower bound estimation $LB(v_q, v_o) = |\delta(v_p, v_q) - \delta(v_p, v_o)|$ from precomputed distances between objects $v_o, v_p, v_q \in R^n$, we show distance distributions for the example models from Section 2. For normalized vectors, the cosine similarity is transformed to the Euclidean distance using $L_2(v_x, v_y) = \sqrt{2 \cdot (1 - \sigma_{cos}(v_x, v_y))}$. Figure 5 shows several L_2 distance distributions for CLIP and W2VV++ BERT models for the 20K benchmark dataset:

- Image-image variant shows the distance distribution histogram for all pairs of images in the 20K dataset.
- Text-image variant shows the distance distribution histogram for pairs between all text query vector representations and vectors of all images.
- Text-target variant shows the distance distribution histogram for pairs between all text query vector representations and their corresponding target item.
- Distance at rank 2000 variant shows the histogram of distances at rank 2000 from all 327 result sets for all benchmark queries.

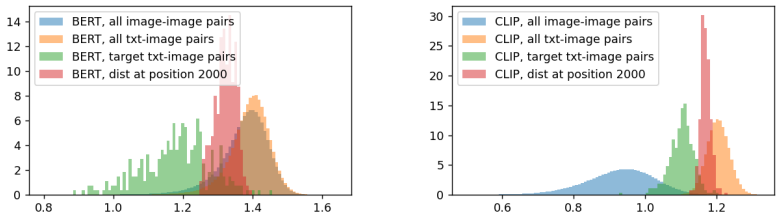

Fig. 5. Distance distribution histograms for CLIP and BERT, 20K benchmark dataset was used. All histograms are normalized, x axis is scaled and does not start at 0.

In the figure, all the selected distance distribution histograms appear in the right part of the possible spectrum, indicating high intrinsic dimensionality [6]. Whereas the W2VV++ BERT model has a similar distance distribution for image-image and text-image pairs, for the CLIP model the two histograms are significantly different. We hypothesise that this inconsistency is caused by different concepts used to design and train the CLIP model. Nevertheless, for both models the necessary distance from query to the searched item is high as well as distances between potentially indexed images. For a fixed high k, the distances at rank k are even higher. This questions filtering power of exact metric

filtering rules and leads to the need for approximate search methods. Although there have been proposed and empirically tested efficient approximate search approaches for metric spaces (e.g., pivot tables [18], permutation approaches [2], or M-Index [20]), the question is whether metric search methods could outperform (for the discussed KIS problem and high k) the simple sequential SVD based filtering approach for W2VV++ BERT (see the previous section) or could deal with specifics of the CLIP based similarity space. We leave this open question as well as all the descriptors of the benchmark dataset for the metric indexing community.

5 Conclusions

In this paper, we focused on the known-item search problem where a larger number of nearest neighbours may be necessary to achieve a high recall. After a brief introduction of the problem, experimental evaluations with two state-of-the-art text-image search models were presented. The difficulty of the task was demonstrated with a benchmark dataset comprising hundreds of query-target pairs. An analysis of browsing performance with simulated user actions provided additional motivation for larger candidate sets. A baseline model for high-dimensional vectors was studied and an open challenge for metric indexing community was provided in the form of a new benchmark dataset accessible at github repository https://github.com/soCzech/KIS-Neighbours.

Acknowledgments

This paper has been supported by Czech Science Foundation (GAČR) project 19-22071Y.

References

1. Amato, G., Bolettieri, P., Carrara, F., Debole, F., Falchi, F., Gennaro, C., Vadicamo, L., Vairo, C.: The visione video search system: Exploiting off-the-shelf text search engines for large-scale video retrieval. Journal of Imaging **7**(5) (2021). https://doi.org/10.3390/jimaging7050076, https://www.mdpi.com/2313-433X/7/5/76
2. Amato, G., Savino, P.: Approximate similarity search in metric spaces using inverted files. In: Proceedings of the 3rd International Conference on Scalable Information Systems. InfoScale '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL (2008)
3. Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Smeaton, A.F., Graham, Y., Kraaij, W., Quénot, G.: Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In: TRECVID 2019. NIST, USA (2019)
4. Barthel, K.U., Hezel, N.: Visually exploring millions of images using image maps and graphs. In: Big Data Analytics for Large-scale Multimedia Search. John Wiley and Sons Inc. (2018)

5. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Comput. Surv. **33**(3), 322–373 (Sep 2001). https://doi.org/10.1145/502807.502809, https://doi.org/10.1145/502807.502809
6. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. ACM Comput. Surv. **33**(3), 273–321 (Sep 2001). https://doi.org/10.1145/502807.502808, https://doi.org/10.1145/502807.502808
7. Cox, I., Miller, M., Omohundro, S., Yianilos, P.: Pichunter: Bayesian relevance feedback for image retrieval. In: Proceedings of 13th International Conference on Pattern Recognition. vol. 3, pp. 361–369 vol.3 (1996). https://doi.org/10.1109/ICPR.1996.546971
8. Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., Yianilos, P.N.: The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. IEEE transactions on image processing **9**(1), 20–37 (2000)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
10. Gurrin, C., Schoeffmann, K., Joho, H., Leibetseder, A., Zhou, L., Duane, A., Dang-Nguyen, D.T., Riegler, M., Piras, L., Tran, M.T., Lokoč, J., Hürst, W.: [invited papers] comparing approaches to interactive lifelog search at the lifelog search challenge (lsc2018). ITE Transactions on Media Technology and Applications **7**(2), 46–59 (2019). https://doi.org/10.3169/mta.7.46
11. Jónsson, B., Khan, O.S., Koelma, D.C., Rudinac, S., Worring, M., Zahálka, J.: Exquisitor at the video browser showdown 2020. In: Ro, Y.M., Cheng, W.H., Kim, J., Chu, W.T., Cui, P., Choi, J.W., Hu, M.C., De Neve, W. (eds.) MultiMedia Modeling. pp. 796–802. Springer International Publishing, Cham (2020)
12. Kratochvíl, M., Veselý, P., Mejzlík, F., Lokoč, J.: Som-hunter: Video browsing with relevance-to-som feedback loop. In: International Conference on Multimedia Modeling. pp. 790–795. Springer (2020)
13. Leibetseder, A., Münzer, B., Primus, J., Kletz, S., Schoeffmann, K.: divexplore 4.0: The itec deep interactive video exploration system at vbs2020. In: Ro, Y.M., Cheng, W.H., Kim, J., Chu, W.T., Cui, P., Choi, J.W., Hu, M.C., De Neve, W. (eds.) MultiMedia Modeling. pp. 753–759. Springer International Publishing, Cham (2020)
14. Li, X., Xu, C., Yang, G., Chen, Z., Dong, J.: W2VV++: fully deep learning for ad-hoc video search. In: Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019. pp. 1786–1794 (2019). https://doi.org/10.1145/3343031.3350906
15. Lokoč, J., Souček, T., Veselý, P., Mejzlík, F., Ji, J., Xu, C., Li, X.: A w2vv++ case study with automated and interactive text-to-video retrieval. In: Proceedings of the 28th ACM International Conference on Multimedia. MM '20, Association for Computing Machinery, New York, NY, USA (2020)
16. Lokoč, J., Bailer, W., Schoeffmann, K., Muenzer, B., Awad, G.: On influential trends in interactive video retrieval: Video browser showdown 2015 – 2017. IEEE Transactions on Multimedia **20**(12), 3361–3376 (2018)
17. Lokoč, J., Kovalčík, G., Münzer, B., Schöffmann, K., Bailer, W., Gasser, R., Vrochidis, S., Nguyen, P.A., Rujikietgumjorn, S., Barthel, K.U.: Interactive search or sequential browsing? a detailed analysis of the video browser showdown 2018. ACM Trans. Multimedia Comput. Commun. Appl. **15**(1), 29:1–29:18 (Feb 2019). https://doi.org/10.1145/3295663

18. Micó, M.L., Oncina, J., Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm (aesa) with linear preprocessing time and memory requirements. Pattern Recognition Letters **15**(1), 9–17 (1994). https://doi.org/https://doi.org/10.1016/0167-8655(94)90095-7, https://www.sciencedirect.com/science/article/pii/0167865594900957
19. Nguyen, P.A., Wu, J., Ngo, C.W., Francis, D., Huet, B.: Vireo @ video browser showdown 2020. In: Ro, Y.M., Cheng, W.H., Kim, J., Chu, W.T., Cui, P., Choi, J.W., Hu, M.C., De Neve, W. (eds.) MultiMedia Modeling. pp. 772–777. Springer International Publishing, Cham (2020)
20. Novák, D., Batko, M., Zezula, P.: Metric index: an efficient and scalable solution for precise and approximate similarity search. Information Systems **36** (2011). https://doi.org/http://dx.doi.org/10.1016/j.is.2010.10.002
21. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine **2**, 559–572 (1901)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. CoRR **abs/2103.00020** (2021), https://arxiv.org/abs/2103.00020
23. Rossetto, L., Schuldt, H., Awad, G., Butt, A.A.: V3C - A research video collection. In: MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part I. pp. 349–360 (2019). https://doi.org/10.1007/978-3-030-05710-7_29
24. Sauter, L., Amiri Parian, M., Gasser, R., Heller, S., Rossetto, L., Schuldt, H.: Combining boolean and multimedia retrieval in vitrivr for large-scale video search. In: Ro, Y.M., Cheng, W.H., Kim, J., Chu, W.T., Cui, P., Choi, J.W., Hu, M.C., De Neve, W. (eds.) MultiMedia Modeling. pp. 760–765. Springer International Publishing, Cham (2020)
25. Schoeffmann, K., Hudelist, M.A., Huber, J.: Video interaction tools: A survey of recent work. ACM Comput. Surv. **48**(1), 14:1–14:34 (Sep 2015)
26. Seidl, T., Kriegel, H.P.: Optimal multi-step k-nearest neighbor search. SIGMOD Rec. **27**(2), 154–165 (Jun 1998). https://doi.org/10.1145/276305.276319, https://doi.org/10.1145/276305.276319
27. Thomee, B., Lew, M.S.: Interactive search in image retrieval: a survey. International Journal of Multimedia Information Retrieval **1**(2), 71–86 (Jul 2012). https://doi.org/10.1007/s13735-012-0014-4
28. Worring, M., Sajda, P., Santini, S., Shamma, D.A., Smeaton, A.F., Yang, Q.: Where is the user in multimedia retrieval? IEEE MultiMedia **19**(4), 6–10 (2012)
29. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search - The Metric Space Approach, Adv. Database Syst., vol. 32. Kluwer (2006)