Interactive Multimodal Learning on 100 Million Images

Jan Zahálka[1], Stevan Rudinac[1], Björn Þór Jónsson[1,2],
Dennis C. Koelma[1], and Marcel Worring[1]
[1]Informatics Institute, University of Amsterdam, The Netherlands
[2]School of Computer Science, Reykjavik University, Iceland
{j.zahalka, s.rudinac, d.c.koelma, m.worring}@uva.nl, bjorn@ru.is

ABSTRACT

This paper presents Blackthorn, an efficient interactive multimodal learning approach facilitating analysis of multimedia collections of 100 million items on a single high-end workstation. This is achieved by efficient data compression and optimizations to the interactive learning process. The compressed ι-I64 data representation costs tens of bytes per item yet preserves most of the visual and textual semantic information. The optimized interactive learning model scores the ι-I64-compressed data directly, greatly reducing the computational requirements. The experiments show that Blackthorn is up to 105x faster than the conventional relevance feedback baseline. Blackthorn is shown to vastly outperform the baseline with respect to recall over time. Blackthorn reaches up to 92% of the precision achieved by the baseline, validating the efficacy of the ι-I64 representation. On the YFCC100M dataset, Blackthorn performes one complete interaction round in 0.7 seconds. Blackthorn thus opens multimedia collections comprising 100 million items to learning-based analysis in fully interactive time.

Keywords

Interactive multimodal learning; multimedia analytics; data compression; YFCC100M

1. INTRODUCTION

Multimedia collections are becoming the central information resource for a growing number of domains, such as physics, forensics, and marketing. This increases the need for methods for fast and insightful analysis of the data. Often, the analysts need to progress from the dataset to insight in hours or few days at most. For example, a journalist discovering stories about the Brussels terrorist attack in the ensuing flood of social media content cannot wait weeks before publishing the story; she needs at least preliminary insights right away. An essential ingredient of insight gain in multimedia analytics in any knowledge domain is *interaction* [17, 28]. Currently, multimedia collections can easily reach millions of items, and even larger datasets are common. Behemoths such as the Yahoo Flickr Creative Commons 100M (YFCC100M) dataset [21] were not so long ago considered inconceivable challenges, yet to truly advance large-scale multimedia analytics, techniques for analyzing such large datasets are needed. How can interactivity in multimedia analytics be assured facing the rapid growth of collection size?

Interactive learning techniques, such as relevance feedback or active learning, are a good fit for multimedia analytics, as they elicit training data labels from the user directly and train the model based on them [9]. This paradigm aligns well with insight, the goal of multimedia analytics. Insight is iteratively built by the analyst through interaction with all or most of the data in the collection and application of the analyst's domain knowledge [17, 28]. Since interactive learning trains its model online based on the user's interactions, the analysis is timely. How to enable interactive and multimodal learning to support large-scale multimedia analytics using modest computational resources (standard high-end workstation with 64 GB RAM and 16-core CPU) is the research question addressed in this work.

In this paper we introduce Blackthorn, an efficient interactive multimodal learning approach for collections of 100 million multimedia items.[1] The architecture of Blackthorn is depicted in Figure 1. Blackthorn brings three main contributions: a multimedia content compression method allowing to fit billions of items into 64 GB of RAM; optimizations to the interactive learning process; and opening up the possibility of analytics on a collection of 100M multimedia items in interactive time (under 1 second per interaction round). To the best of our knowledge, no work has yet addressed fully interactive-learning-based exploration of an image dataset with 100 million items on a single workstation.

2. RELATED WORK

In response to the rapid growth of digital collections, a number of approaches facilitating more efficient search and exploration have been proposed. Interactive learning (e.g., relevance feedback) has been proven invaluable for improving multimedia information retrieval [9], and benchmarks such as the Video Browser Showdown were introduced [19]. However, the collections considered in these benchmarks are much smaller than the one considered in this work. In addition, most of current approaches employ entire computa-

[1]As this paper focuses on *efficiency* of interactive learning, the name is inspired by the efficiency of the blackthorn shrub: as fire wood, it yields much heat with little smoke.

ICMR'16, June 06-09, 2016, New York, NY, USA

DOI: http://dx.doi.org/10.1145/2911996.2912062

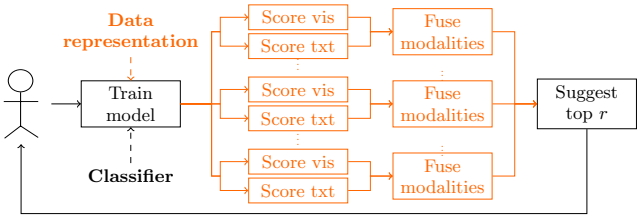

Figure 1: Interactive learning with Blackthorn. The components of interactive learning innovated by Blackthorn are marked orange.

tional clusters to facilitate analysis of the data in interactive time [6, 12]. The algorithmic efficiency component often boils down to reducing the number of items for which the similarity is computed, reducing the dimensionality of data representation, or simplifying the similarity computation.

Early work demonstrated that exact k-nn search techniques suffer from the curse of dimensionality [2], paving the way for approximate methods. Several recent works from the multimedia and computer vision communities rely on embedding visual features in binary space to compress the data and reduce the distance computation time [4, 8, 11, 15, 16, 24, 29]. LSH is a well-known scalar-based indexing technique [1]. The NV-tree is a scalar-based method, which has been shown to significantly outperform LSH while requiring about 6 bytes per descriptor [14]. Product quantization is a family of vector-based quantization schemes producing compact code-words by indexing low-dimensional subspaces independently [10, 13, 26].

While these approaches for scaling up image retrieval were proven effective in k-nn search, their applicability in analytic tasks remains limited for at least two reasons. First, binary hashes are not suited for classification, which is an essential component of most analytic platforms. Moreover, in many analytic tasks search and exploration are alternately performed, which requires updating, summarizing and repartitioning of the collection based on user interactions [28]. Such operations require preservation of the original vectors or a significant portion of information contained in them. Novel approaches are needed for building such representations for efficient deployment in analytic tasks.

3. DATA REPRESENTATION

In the interactive learning setting, it is desirable that the user understands the data representation. Hence, we focus on representations carrying semantic meaning. However, in case the collection comprises 100 million items, the memory requirements of such representations are prohibitive. Indeed, assuming 1000 visual concepts, 100 text topics, and 8 bytes (B) as the size of one floating-point number, the semantic representation of a 100M collection requires roughly 880 GB of RAM. We address this issue by introducing a data compression method that greatly reduces the representation size with only a modest information loss.

Semantic representations such as visual concept scores output by a convolutional neural network [20] or LDA topics in the text domain [3] are *sparse*, allowing the design of an efficient representation following the established sparse representation practices [25]. Thus, in each modality, most of the information contained in a feature vector with n_f features can be preserved by encoding only the top t_f features in compact bit fields, since $t_f \ll n_f$. For the k-th feature, $1 \leqslant k \leqslant t_f$, we need to encode the feature ID (denoted i_k) and the respective score (denoted s_k). The bit cost for encoding the ID (denoted b_i) is $\lceil \log_2 n_f \rceil$. The bit cost for encoding the score (denoted b_s) is $\lceil p \cdot \log_2 10 \rceil$, where p denotes the decimal precision. Further, we encode the number of features actually encoded for item I (denoted e_I), as due to sparsity, e_I could be smaller than t_f. The bit cost of this encoding (denoted b_e) is $\lceil \log_2 f \rceil$. Obtaining the bit field compressing the feature IDs of item I in each modality (denoted $\mathcal{I}$) is defined by Equation 1. Obtaining the bit field compressing the scores, $\mathcal{S}$, is defined in Equation 2. Here, $\vee$ denotes the binary OR operation, $\ll$ the left bit shift operation, and $[\cdot]$ the rounding function.

$$\mathcal{I} = e_I \vee \bigvee_{k=0}^{e_I-1} i_k \ll (k \cdot b_i + b_e) \quad (1)$$

$$\mathcal{S} = e_I \vee \bigvee_{k=0}^{e_I-1} [10^p \cdot s_k] \ll (k \cdot b_s + b_e) \quad (2)$$

The memory bit requirements for $\mathcal{I}_I$ and $\mathcal{S}_I$, $b_\mathcal{I}$ and $b_\mathcal{S}$, are then equal to $e_I \cdot b_i + b_e$ and $e_I \cdot b_s + b_e$, respectively. $\mathcal{I}$ and $\mathcal{S}$ can be decompressed to obtain the encoded features, setting the features outside the top t_f to 0. Let $\wedge$ denote the bitwise AND operation and $\gg$ the right arithmetic bit shift operation. Equation 3 describes the decompression of the number of encoded features (e_I), Equation 4 describes the decompression of the k-th feature ID, and Equation 5 describes the decompression of the respective feature score.

$$e_I = \mathcal{I} \wedge \left(2^{b_e} - 1\right) \quad (3)$$

$$\delta(\mathcal{I}, k) = (\mathcal{I} \gg (k \cdot b_i + b_e)) \wedge \left(2^{b_i} - 1\right) \quad (4)$$

$$\delta(\mathcal{S}, k) = \frac{(\mathcal{S} \gg (k \cdot b_s + b_e)) \wedge \left(2^{b_s} - 1\right)}{10^p} \quad (5)$$

Since most current machines are 64-bit, it is desirable to choose t_f and p so that $\mathcal{I}$ and $\mathcal{S}$ align with the 64-bit architecture. Given a semantic representation with less than 1024 relevant visual concepts and text topics and $p = 3$, we can encode $\mathcal{I}$ and $\mathcal{S}$ each in ι 64-bit integers, keeping the top 6ι features. This setting enables a trade-off between minimizing memory and computational requirements (low ι) on the one hand and information loss on the other (high ι).

The resulting memory-efficient **ι-integer (ι-I64) representation** only requires $4\iota + 1$ 64-bit integers per item: 2ι for $\mathcal{I}$ and $\mathcal{S}$ per modality, as well as an additional integer for the item ID. With $\iota = 1$, the representation is 220x smaller than the uncompressed representation and allows fitting 1.5 billion items into 60 GB of memory. This makes ι-I64 representation suitable for large-scale interactive multimodal learning.

4. INTERACTIVE LEARNING

Each interactive multimodal learning session consists of a number of *interaction rounds*, each with 3 steps. Firstly, a classifier is trained for each modality on the user-provided examples. Secondly, the unlabelled items in the collection are scored. Thirdly, the rankings per modality are aggregated and the top r results are returned to the user. A user

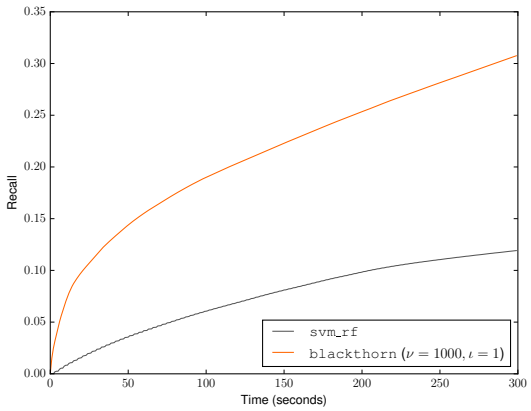

Figure 2: Recall over time (YFCC100M-California).

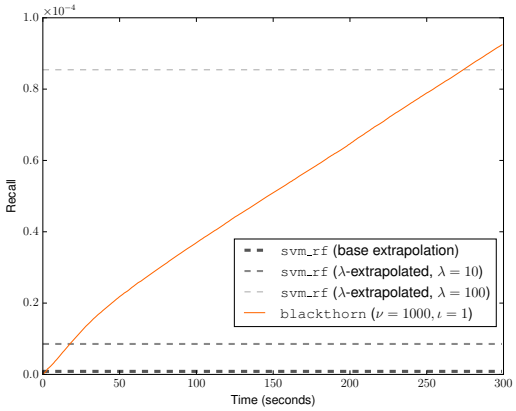

Figure 3: Recall over time (YFCC100M). The baseline values are extrapolated from YFCC100M-California, with λ further multiplying the extrapolated performance.

Table 1: Precision and time per interaction round (YFCC100M-California).

Algorithm	Precision	Time
`svm_rf`	0.48	3.16 s
`blackthorn` ($\nu = 1000$, $\iota = 1$)	0.44 (92%)	0.07 s (45x faster)
`blackthorn` ($\nu = 200$, $\iota = 1$)	0.35 (73%)	0.03 s (105x faster)

typically provides at most tens of examples per interaction round, regardless of the collection size. However, the time cost of steps 2 and 3 increases dramatically as the collection grows larger. We have alleviated this issue somewhat by data compression and an efficient parallel implementation in C, but this does not solve the issue. In this section, we address Blackthorn's algorithmic optimizations.

Blackthorn utilizes a linear SVM classifier. To improve the classifier's efficiency, we design a custom SVM scoring function that operates directly in ι-I64-compressed space. The SVM scoring function σ performs a vector multiplication of the model's weight vector $\mathbf{w}$ with the feature vector $\mathbf{x}$, adding bias b to the result. This costs $2n_f + 1$ mathematical operations (multiplications and additions) per item and modality. In the ι-I64 representation, only the 6ι encoded features carry any value. Hence, we can obtain the SVM score by multiplying only the decompressed feature values by the corresponding values of the model's weight vector $\mathbf{w}$:

$$\sigma_c(I) = \sum_{j=0}^{\iota-1} \sum_{k=0}^{e_I-1} \left(w_{\delta(\mathcal{I}, 6j+k)} \cdot \delta(\mathcal{S}, 6j+k) \right) + b \qquad (6)$$

This computation costs $42\iota + 1$ mathematical operations per item and modality. Since $\iota \ll n_f$, this amounts to a considerable speedup: for example, in a setting with $\iota = 1$ and $n_f = 1000$ (for instance, 1000 visual concepts), the number of mathematical operations is reduced 46.5x.

Late modality fusion is another aspect requiring attention with respect to algorithmic complexity. With a straightforward implementation, this requires $O(N \log N)$ time, which is higher than the $O(N)$ complexity of scoring. To reduce fusion complexity to $O(N)$, we follow the protocol of top-d rank aggregation [7] in order to obtain the top r relevant results. For each modality, the top-scoring ν ($r \leqslant \nu \ll n$) results in that modality are nominated to the final ranking in $O(N)$ time. The multimodal ranking is obtained by performing rank aggregations on the nominated items ($O(\nu \log \nu)$ time). Because $\nu \ll N$, modality fusion is essentially computed in linear time. Optimized modality fusion in combination with efficient scoring ensures that all stages of interactive learning complete in $O(N)$ time.

5. EXPERIMENTAL SETUP

To gauge Blackthorn's analytic performance and scalability, we report three evaluation measures: recall over time, precision, and time per interaction round. We compare our approach (`blackthorn`) with the standard linear-SVM-based relevance feedback approach (`svm_rf`) [9]. Both algorithms use 1000 ILSVRC ImageNet concepts [18] extracted by GoogLeNet [20] as visual features and 100 LDA topics [3] extracted using Gensim [23] from the tags, title, and description of each image. Both `blackthorn` and `svm_rf` are implemented in C and use the VLFeat SVM library [22]. Two datasets are used in the experiments. The first one is the YFCC100M dataset itself, which is too big to be handled by `svm_rf`. To allow for direct comparison of `blackthorn` and `svm_rf`, we use a custom second dataset, YFCC100M-California, which contains all items in YFCC100M tagged with the *california* tag. This dataset has 1,221,608 items.

To evaluate YFCC100M in an interactive setting, we perform a task inspired by the MediaEval Placing task [5]. The evaluation task is to retrieve items pertaining to a large city based *solely* on visual and text content. Note that this is a very difficult task. However, it is suitable the sake of providing a comparison between approaches. This task is performed by artificial users (*actors*), each corresponding to one city in the top 50 cities with the highest number of items. The geo information is discarded for evaluation purposes. Each actor conducts 50 sessions, initializing each with 100 uniform random samples from its city as positives and 200 uniform random samples from the collection as negatives. In each interaction round, 25 items are suggested.

6. RESULTS ON YFCC100M-CALIFORNIA

The precision and time per interaction round are reported in Table 1. The development of recall in the time period of

Figure 4: Example items involved in the first interaction round performed on the YFCC100M dataset by the actor interested in items from Prague. None of the suggestions are regarded as relevant for our evaluation task, despite the semantically similar content.

5 minutes is depicted in Figure 2. For all three evaluation measures, we compare `svm_rf` with the best-scoring ν, ι configuration of `blackthorn`. We have tested `blackthorn` with $\nu \in \{100, 200, \ldots, 1000\}$ and $\iota \in \{1, 2, 3\}$. Varying ν has some impact on precision, with $\nu = 100$ reaching roughly 75% of the precison of $\nu = 1000$. Increasing ν impacts the time per interaction round: $\nu = 1000$ costs roughly twice the time of the $\nu = 100$. Regarding recall over time, low ν tends to outperform the higher values due to the lower time per interaction round. Regarding ι, we observe that increasing its value brings very minimal precision and recall gain, but has a negative impact on the time per interaction round. We believe that the low precision and recall gain is caused by insufficient decimal precision of the compressed values. We aim to investigate this further. Overall, we recommend choosing $\iota = 1$ in all cases, higher ν for maximum precision, and lower ν for maximum recall over time and speed.

The experiments clearly show that computational speed and efficiency are `blackthorn`'s forte. One interaction round of `blackthorn` on YFCC100M-California takes 0.05 ± 0.01 seconds on average across all evaluated ν values, and the fastest configuration is 105x faster than `svm_rf`. The information loss incurred by the ι-I64 representation turns out to be affordable. The highest-precision configuration of `blackthorn` keeps 92% of the average precision with a speed-up of 45. The recall over time plot (Figure 2) clearly shows that `blackthorn` dramatically surpasses `svm_rf`.

7. RESULTS ON YFCC100M

Our approach succeeded in the test of interactivity on the large dataset: on average across all ν values, `blackthorn` takes 0.69 ± 0.01 seconds per interaction round. Figure 3 shows the development of recall over time on YFCC100M. Given the dataset's prohibitive size, `svm_rf` is infeasible. In order to compare the approaches, we extrapolate `svm_rf`'s results on YFCC100M-California. Since `svm_rf`'s computational cost is $O(N)$, the 3.16 second performance on 1.2M becomes roughly 4.3 minutes. Hence, only the first interaction round needs to be considered. The base extrapolation is twice the recall value of `blackthorn`'s recall in the first interaction round, following the trend observed on YFCC100M-California. In addition, we report two λ-extrapolated `svm_rf` baselines, where $\lambda \in \{10, 100\}$ further multiplies the base extrapolation. We believe this more than compensates for any potential unverifiable factors influencing the true recall scalability. The results clearly show `blackthorn`'s efficiency with respect to recall gain over time: it is able to surpass all three interpolations in the first 5 minutes of the analysis.

The precision per interaction round on the YFCC100M dataset, averaged over all ν values, is 0.003 ± 0.0001. Note that the low values are influenced by two important factors. Firstly, the size of the dataset brings an increase in noise and lower discriminative power of feature vectors. Secondly, the evaluation task is extremely difficult. Indeed, there are cases where the algorithm's suggestions are evaluated as irrelevant, despite their semantic similarity to the provided positives. This is demonstrated in Figure 4. In a less strict setting, these suggestions could be easily deemed relevant. Despite the low values of precision, `blackthorn` does provide relevant suggestions even on the large dataset.

8. CONCLUSION

In this paper, we have presented Blackthorn, an efficient interactive multimodal learning framework which enables full interactive-learning-based analysis of a collection with 100M multimedia items. The data compression method of Blackthorn, ι-I64, is shown to reduce the size of multimodal features by a factor of 220 *and* preserve most of the information contained in the original features. Blackthorn yields a massive 105x speed-up in comparison to the classic relevance feedback paradigm. The experiments further show that Blackthorn is suitable for the analysis of the entire YFCC100M dataset. It is able to learn on the fly from the user-provided training samples, and one interaction round on the entire 100M collection takes less than a second. Its high efficiency and low resource cost would also support multi-category exploration with proactive suggestions by the system. The analysis can be performed on a single standard high-end workstation with 64 GB RAM and 16 CPU cores. In order to foster further research on YFCC100M, the Blackthorn software package has been made available as an open-source tool [27]. In conclusion, Blackthorn is a step forward towards fully harnessing the wealth of information contained in large-scale multimedia collections.

9. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[2] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is 'nearest neighbor' meaningful? In *ICDT*, 1999.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[4] S. Bondugula, V. Manjunatha, L. S. Davis, and D. Doermann. Shoe: Sibling hashing with output embeddings. In *ACM MM*, pages 823–826, 2015.

[5] J. Choi, C. Hauff, O. V. Laere, and B. Thomee. The Placing task at MediaEval 2015. In *MediaEval*, 2015.

[6] O. de Rooij and M. Worring. Active bucket categorization for high recall video retrieval. *IEEE TMM*, 15(4):898–907, June 2013.

[7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.

[8] A. Gordo, F. Perronnin, Y. Gong, and S. Lazebnik. Asymmetric distances for binary embeddings. *IEEE TPAMI*, 36(1):33–47, 2014.

[9] T. Huang, C. Dagli, S. Rajaram, E. Chang, M. Mandel, G. E. Poliner, and D. Ellis. Active learning for interactive multimedia retrieval. *Proc. IEEE*, 96(4):648–667, 2008.

[10] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE TPAMI*, 33(1), 2011.

[11] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE TPAMI*, 34(9):1704–1716, 2012.

[12] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. G. Hauptmann. Fast and accurate content-based semantic search in 100M internet videos. In *ACM MM*, pages 49–58, 2015.

[13] Y. S. Kalantidis and Y. Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *IEEE CVPR*, 2014.

[14] H. Lejsek, B. T. Jónsson, and L. Amsaleg. NV-Tree: nearest neighbors at the billion scale. In *ICMR*, 2011.

[15] P. Li, M. Wang, J. Cheng, C. Xu, and H. Lu. Spectral hashing with semantically consistent graph for image indexing. *IEEE TMM*, 15(1):141–152, 2013.

[16] M. Norouzi, A. Punjani, and D. Fleet. Fast search in hamming space with multi-index hashing. In *CVPR*, pages 3108–3115, 2012.

[17] C. North. Towards measuring visualization insight. *IEEE TCGA*, 26(3):6–9, 2006.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[19] K. Schoeffmann. A user-centric media retrieval competition: The video browser showdown 2012-2014. *IEEE MM*, 21(4):8–13, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

[21] B. Thomee, B. Elizalde, D. A. Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016.

[22] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[23] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *LREC*, pages 45–50, 2010.

[24] J. Wang, H. T. Shen, S. Yan, N. Yu, S. Li, and J. Wang. Optimized distances for binary code ranking. In *ACM MM*, pages 517–526, 2014.

[25] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[26] E. S. Xioufis, S. Papadopoulos, Y. Kompatsiaris, G. Tsoumakas, and I. P. Vlahavas. A comprehensive study over VLAD and product quantization in large-scale image retrieval. *IEEE TMM*, 16(6), 2014.

[27] J. Zahálka. Blackthorn. http://staff.fnwi.uva.nl/j.zahalka/blackthorn.html, 2016.

[28] J. Zahálka and M. Worring. Towards interactive, intelligent, and integrated multimedia analytics. In *IEEE VAST*, pages 3–12, 2014.

[29] L. Zhang, Y. Zhang, J. Tang, X. Gu, J. Li, and Q. Tian. Topology preserving hashing for similarity search. In *ACM MM*, pages 123–132, 2013.