

Mental Visual Indexing: Towards Fast Video Browsing

Richang Hong[†], Jun He[†], Hanwang Zhang[‡] and Tat-Seng Chua[‡]

[†]School of Computer and Information, Hefei University of Technology

[‡]School of Computing, National University of Singapore

{hongrc.hfut, hj.hfut.mail, hanwangzhang}@gmail.com, dcscts@nus.edu.sg

ABSTRACT

Video browsing describes an interactive process where users want to find a target shot in a long video. Therefore, it is crucial for a video browsing system to be fast and accurate with minimum user effort. In sharp contrast to traditional Relevance Feedback (RF), we propose a novel paradigm for fast video browsing dubbed Mental Visual Indexing (MVI). At each interactive round, the user only needs to select one of the displayed shots that is most visually similar to her mental target and then the user's choice will further tailor the search to the target. The search model update given a user feedback only requires vector inner products, which makes MVI highly responsive. MVI is underpinned by a sequence model in terms of Recurrent Neural Network (RNN), which is trained by automatically generated shot sequences from a rigorous Bayesian framework, which simulates user feedback process. Experimental results on three 3-hour movies conducted by real users demonstrate the effectiveness of the proposed approach.

Keywords

Mental Search, Video Browsing, RNN, Temporal Model, Query intent

1. INTRODUCTION

A common scenario is that we have seen a movie long time ago and one day we are eager to seek an interesting shot that just suddenly comes to our mind. Then, we might have to scroll the progress bar in a video player that forward/rewind to find our target shot, which is a video browsing issue in the temporal space. When the movie is long, *e.g.* over 2 hours, the procedure is sometimes frustrating since it is very time-consuming and tricky, especially on small-screen mobile devices. We argue that the conventional interactive search by scrolling in most commercial players is less intelligent since it neglects the inherent visual information of videos.

In fact, many video shots are close to each other in the feature space or visually similar although they are far apart

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '16, October 15–19, 2016, Amsterdam, The Netherlands.

© 2016 ACM. ISBN 978-1-4503-3603-1/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2964284.2967296>

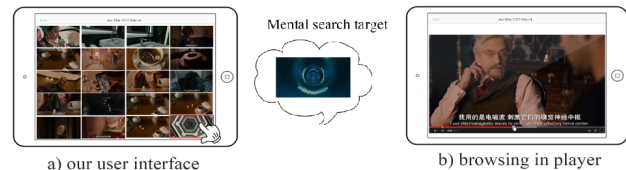


Figure 1: Two different modes of interaction: a) user selects one shot according to the visual similarity, see the one marked in the red box; b) user scrolls the progress bar to find the target shot.

in the temporal space. To exploit visual features may accelerate the video browsing process. Be aware that exploiting visual features of videos has been extensively studied in multimedia community [7, 8, 18, 19, 20, 22], *e.g.* TRECVID Instance Search task (INS) has been run for five years since 2010 to address the need of finding more video segments of a certain person, object, or place in a video collection, given one or more visual examples of the specific item [11]. Video Browsing Showdown (VBS) is another event which focuses on video analysis and retrieval. To be concrete, VBS is a live video browsing competition where researchers evaluate and demonstrate the efficiency of their video browsing tools. The VBS participants use their own systems to perform an interactive search in the specified video files taken from a common data set and try to find the desired segment as fast as possible [14].

Relevance Feedback (RF) is a feature of some information retrieval systems [13] and is widely used in different kinds of works [1, 2, 5, 9, 10]. The intuitive idea behind RF is to involve users in the retrieval process so as to improve the final result [23]. Typically, users are asked to label the top candidates returned by the search model as “relevant” or “irrelevant”. The feedbacks are then used to refine the search model. Through iterative feedback and model refinement, relevance feedback attempts to capture the information needs of users and improve search results gradually [21]. The main drawback is that RF suffers from a lot of user efforts which prevent a system from being easy-to-use. In a retrieval system, for instance, a user is usually asked to refine her query by telling the system whether a human face detector is needed or not, or by pinpointing and submitting several toy examples. Moreover severely, computation cost of RF is high. So, it would not support many users doing RF simultaneously. Since RF involves transmitting more data,

it leads to a more expensive Internet traffic than traditional retrieval methods.

To this end, we propose an spectacularly fast and effective method for video browsing. In the interactive process, the only user effort required is to click one shot which most visually relates to the target, out of 20 displayed shots. Every time when asked to make a selection, the user is involved in a new search session. It is obvious that a shot which is closely related to traffic will never be chosen if the user wants the one that is about a banquet. Since the user selection reflects her query intent, we can assist the user to find the target as quickly as possible if we succeed in capturing the query intent conveyed in the user selections. In other words, it's possible to learn about the target from user feedbacks. Thus, we propose a fast and traffic-economic video browsing paradigm called Mental Visual Indexing (MVI). The core of MVI is a temporal model, specifically a RNN model referenced as char-RNN [6]. The model receives the user feedback and takes the corresponding shot ID as input, indexing the user's query intent in its hidden variables. The output of the model is a prediction which indicates the probability of one shot being the user's search target. The model can quickly response since it requires no heavy real-time similarity comparison. To train the model, we first select every shot in the shot repository as the mental target and generate 40 pseudo user feedback sequences using the statistical model detailed in Section 2.3. A pseudo user feedback sequence is a sentence which is composed of at least two shot IDs, starting and ending with the target shot ID. Then we feed all the sequences to our char-RNN model. Experiments show that our char-RNN based framework is accurate, fast, and user-friendly.

Our contributions are summarized as follows:

- We propose a very fast and traffic-economic video browsing paradigm called Mental Visual Indexing (MVI). It only requires user click from mental.
- MVI is built on a deep temporal model: char-RNN, to index the long-term visual relationships among click-through user feedbacks. It is automatically trained by sampling machine-generated visual feedback steps. Once trained, it can quickly calculate the next candidate video shots.
- We develop a video browsing interface. Experiments on real-world movies show that user can find a target in 3-5 steps.

2. THE PROPOSED METHOD

2.1 System Overview

Given a long video and a mental target shot, our goal is to assist the user to find the target via minimum interactions. To achieve this goal, we proposed a framework which consists of two main components: a statistical framework and a Character-Level Language Model, *i.e.* the char-RNN model. The statistical framework is employed to generate pseudo user feedbacks or click histories based on visual similarity. These feedbacks accumulate into a large training data set for char-RNN network training. The char-RNN model is employed to analyze the interrelationships between video shots. Once well trained, the model is able to successfully infer user's query intent during the procedure of video

browsing. See Figure 2 for a general understanding about our framework.

Our system is divided into two parts: off-line part and on-line part. In the off-line part, we cut a video into shots which are represented by their VGG19 [15] features. Then, we feed all the Shots-Target pairs to the statistical framework which results in a bunch of pseudo user feedback sequences. Each Shots-Target pair here is composed of all the shots and a randomly selected target. A sequence indicates that all the video shots contained in the sequence have some sort of inner connection in the visual feature space. We trained the char-RNN model on the sequences. In the on-line part, we initially show the user some shots, then the user selects the one that is most similar to her mental target. The system receives the user's choice and feeds the corresponding shot ID to the well trained RNN model. The model, then, predicts the next candidate shots for display. The process lasts until the user meets her target.

2.2 Mental Visual Indexing

Suppose there are N shots segmented from a long video, denoted as $\mathcal{V} = \{1\dots i\dots N\}$ for simplicity. Let $\mathcal{M} \subset \mathcal{V}$ denotes the possible subset of the long video shots, *i.e.*, it can be considered as candidates of users' target mental shots. Generally, \mathcal{M} is unknown or even imaginary to the system. We assume that if a member of \mathcal{M} is displayed, the user can easily find the target by using this shot as a starting point for precise localization in the long video.

We denote the user feedback history as $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{c_t = k, k \in \mathcal{D}_t\}$, of which \mathcal{D}_t is the shots displayed during search session t and \mathcal{H}_{t-1} stands for the previous history. The essence of MVI lies in its ability to learn about the user's query intent from \mathcal{H}_t gradually in the process of user interaction with the system. Eq (1) is the update formula for query intent modeling, where s_{t-1} is the intent guessed after the previous interaction, c_t donates the user's feedback in the current round and s_t stands for the updated intention.

$$s_t = f(c_t, s_{t-1}). \quad (1)$$

Inspired by [16], we employed the char-RNN [6] model to automatically learn about the inherent query information as the search session lasts. Originally, the model is designed to take character sequence as input and predict the next character. In this work, we extend the model to take a shot ID as input and output the IDs of the candidate shots for next display. The model is trained with a dictionary file which is a text file with unique shot ID per line and a text file containing all the pseudo user feedback sequences. Each sequence is a machine-generated sentence with shot IDs as the words. All the sequences are generated by randomly selecting one shot as the target and feeding all the shots and the pseudo target as a Shots-Target pair to our statistical framework. The statistical framework then builds the sequence by continuously comparing visual similarity until the target is found (see Figure 2).

When trained on the sequences, the RNN model is reliable to grasp the inner connections among video shots because only relevant shots appear in the same sequence. Therefore, the model guarantees the output shots are always very related with the input one and captures user intent by updating its hidden variables, *i.e.* s_t . The model is very fast in that the heavy computation is done in the off-line part

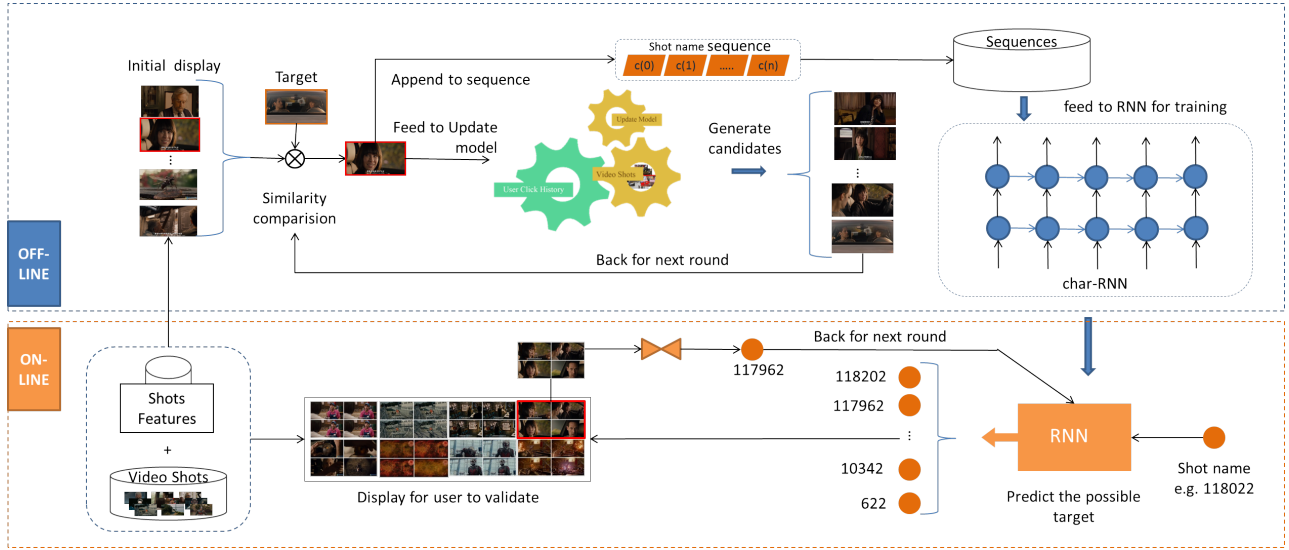


Figure 2: The framework is divided into off-line part and on-line part: 1) In off-line part, we generate training by simulating user feedback and train the char-RNN model; 2) In on-line part, the model takes the user feedback as input and predicts the candidates for next display.

and in the on-line part, the only required computation is one step of vector inner product.

2.3 Statistical Framework for Training

The statistical framework is a variant of the one proposed in [2] which is inspired by [1, 17]. Given the Shots-Target pair, the framework is employed to generate the pseudo user feedback sequences in the off-line part which form the training data for our char-RNN model. To be concrete, imagining this is a real user feedback procedure, we maintain N -independent Bayesian system $p_t(i)$ for each video shot i . In particular, $p_t(i)$ is the estimated probability of shot i belonging to the target set \mathcal{M} :

$$p_t(i) = P(i \in \mathcal{M} | \mathcal{H}_t). \quad (2)$$

Since we have no prior knowledge about \mathcal{M} , we take $p_0(i) = 0.5$ as the starting state.

By applying Bayes rule, we can rewrite the definition in Eq (2) as:

$$p_t(i) = \frac{P(c_t = k | \mathcal{H}_{t-1}, i \in \mathcal{M}, k \in \mathcal{D}_t) P(i \in \mathcal{M} | \mathcal{H}_{t-1}, k \in \mathcal{D}_t)}{P(c_t = k | \mathcal{H}_{t-1}, k \in \mathcal{D}_t)}. \quad (3)$$

In order to further simplify the above model for computability, we should note several statistical sufficiency and assumptions. First, $c_t = k$ is a sufficient statistic for $k \in \mathcal{D}_t$ since it is obvious that once the user clicks on k , k must be in \mathcal{D}_t . Second, the fact $i \in \mathcal{M}$ is unrelated to the display set \mathcal{D}_t . Third, once given the display \mathcal{D}_t , the history \mathcal{H}_{t-1} is no longer informative to the user click $c_t = k$. Based on these statistical properties, we have the update model for $p_t(i)$ as:

$$\begin{aligned} p_t(i) &= \frac{P(c_t = k | i \in \mathcal{M}, k \in \mathcal{D}_t) p_{t-1}(i)}{P(c_t = k | k \in \mathcal{D}_t)} \\ &= \frac{P_+(k|i, D) p_{t-1}(i)}{P_+(k|i, D) p_{t-1}(i) + P_-(k|i, D) (1 - p_{t-1}(i))} \end{aligned} \quad (4)$$

where

$$\begin{cases} P(c_t = k | i \in \mathcal{M}, k \in \mathcal{D}_t) = P_+(k|i, D), \\ P(c_t = k | i \notin \mathcal{M}, k \in \mathcal{D}_t) = P_-(k|i, D). \end{cases} \quad (5)$$

Therefore, all we need to update $p_t(i)$ is to model the user feedback $P_+(k|i, D)$ and $P_-(k|i, D)$. We assume that the user always selects the shot that most matches her mental target from the shots displayed, the click probability can be modeled as:

$$\begin{cases} P_+(k|i, D) = \frac{s(i, k)}{\sum_{j \in \mathcal{D}_t} s(j, k)}, \\ P_-(k|i, D) = \frac{d(i, k)}{\sum_{j \in \mathcal{D}_t} d(j, k)}. \end{cases} \quad (6)$$

where $d(\cdot, \cdot)$ and $s(\cdot, \cdot)$ denote the distance and similarity in the feature space, respectively. The intuition behind the above definitions is that if i is the target shot, the probability is enhanced by the similarity between i and k ; if i is not the target, the probability is depressed by the distance between i and k .

Now the problem is how to choose \mathcal{D}_{t+1} given the user feedback and \mathcal{D}_t . The simplest approach may be to select several shots that are most likely to belong to \mathcal{M} . Unfortunately, it's not efficient in the real user interactive procedure. For example, usually two very similar shots are probably either both in \mathcal{M} or both not in \mathcal{M} , which may lead to the unexpected situations where both of the shots are displayed in the next round or none of them are displayed. Both of the situations will prevent the user from getting more information about the shot repository. To avoid this case, we select the most informative subset $\mathcal{D}_{t+1} \subset \mathcal{V}$ by minimizing $\min_{\mathcal{D} \subset \mathcal{V}} \text{Entropy}(z_{t+1} | \mathcal{H}_t)$. Intuitively, the selection attempts to make an optimal Voronoi partition for the video shots. The detailed calculation of the probability z_{t+1} is given in [1].

The situation is a little different when it comes to our user feedback simulation. A simple machine does not have an imaginary target and, of course, does not have the ability to tell visual similarity intelligently as we humans do. The workaround is to select the one that is closest to the target in the feature space as the feedback. We keep recording the

feedback until the predetermined target is found and treat each record as a sequence.

3. EXPERIMENTS

3.1 Experiment Settings

Experiments were conducted on 3 videos: Ant-Man, Life of PI and a product launch video from Microsoft. We first produced three sets with about 8200, 7780 and 12500 frames respectively by extracting one frame every 20 frames. Then, Caffe toolbox [4] and the model discussed in [15] were employed to generate a high level representation (VGG19) for each frame. The 4096-dimension VGG19 feature we used in our experiments is the output of full-connection layer “fc7” before RELU and dropout layer.

Video was cut into shots by comparing frame-level features. For each frame, we compared its feature with the previous one to see how similar they are. After that consecutive frames with high similarity were restructured into one shot by averaging frame-level features. In order to give users a clearer picture of the shot, an informative thumbnail was generated. For computability, the frame-level features as well as the shot features were normalized. Unlike our baseline, feature dimension reduction is not required in that no copious real-time feature comparisons are needed. Even in the baseline, only when high precision was not strictly demanded did we apply PCA-whitening [3, 12] to reduce the dimension into 512 [2].

We chose to set $rnn_size = 128$, $num_layers = 1$, $seq_len = 20$, $learning_rate = 0.01$ and the other training parameters to their default values after comparing the performance under different model settings. The most important parameter is seq_len which specifies the length of each stream and, as well, limits the back-propagation of the gradients. For instance, if $seq_len = 10$, then the gradients will never back-propagate more than 10 time steps, and the model might not find dependencies longer than this length in number of characters. Because the mean sequence length over all training sequences is about 10, we set $seq_len = 20$. Although it is recommended to use num_layers of either 2/3, we found a better performance could be achieved when we set $num_layers = 1$ [6].

3.2 Experiment Results

In this section we first evaluated the efficiency of our approach, then we compared our approach with the one proposed in [2], finally we conducted a user study to identify the effectiveness of our approach. All the experiments were based on the 10 randomly selected targets (see Figure 3).

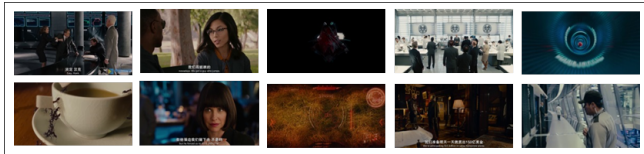


Figure 3: The 10 randomly selected target shots

To have a basic understanding about the efficiency of our framework, we selected 4 target shots from Figure 3 and conducted a general validation. Table 1 lists the result. It





				iterations found
4	3	7	4	
yes	yes	yes	yes	

Table 1: Efficiency validation of our framework

shows that our approach is effective and, in average, we can find a target in 4 iterations.

The next set of experiment was performed on each of the shots in Figure 3. Both the baseline method and our approach were evaluated. The result shows that, by average, our approach can locate the target in 30 seconds with 96% accuracy within 4 iterations which is much better than the baseline (70s, 8 iterations and 88% accuracy).

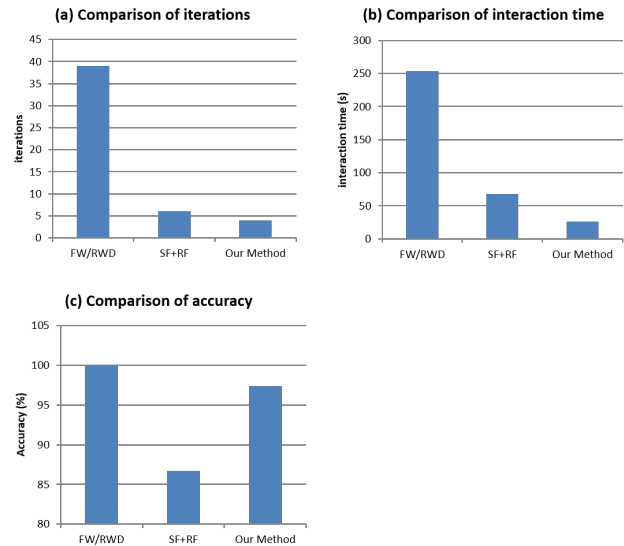


Figure 4: User study results: a) mean iterations comparison; b) mean interaction time comparison; c) mean retrieval accuracy comparison.

Eight students were involved in our user study, with five of them are enrolled in computer vision studies and the others are enrolled in natural language processing (NLP) studies, which means about half of the student are novices. Three tasks have been conducted: 1) find all the targets in any common video player; 2) find all the target shots in the system proposed in [2]; 3) find all the target shots in our system. The general comparison of three tasks is shown in Figure 4. The result shows that our approach can measurably improve system performance.

4. CONCLUSIONS

In the paper, we extended the char-RNN temporal model to video browsing task by a new interactive search paradigm named Mental Visual Indexing (MVI). The intuition behind MVI is to learn the inherent connections between video shots automatically and capture user query intent gradually by analyzing the user feedbacks. When trained on a large set of simulated user feedbacks, the proposed model responses quickly and accurately to the user query. Our results show

that the char-RNN based model improves system performance by shortening system response time, decreasing interactive iterations and increasing prediction accuracy.

5. ACKNOWLEDGMENTS

This work was supported in part by the Anhui Fund for Distinguished Young Scholars under Grant 1508085J04, in part by the National High-Tech Development Program of China under Grant 2014AAA015104, in part by the National Natural Science Foundation of China under Grant 61472116.

We are grateful to all study participants for their contributions and to NExT for the supply of lab facilities. NExT research is supported by Nation Research Foundation, Prime Minister’s Office, Singapore under its IRC@SG Funding Initiative.

6. REFERENCES

- [1] M. Ferecatu and D. Geman. A statistical framework for image category search from a mental picture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):1087–1101, 2009.
- [2] J. He, X. Shang, H. Zhang, and T.-S. Chua. Mental visual browsing. In *MultiMedia Modeling*, pages 424–428. Springer, 2016.
- [3] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In *Computer Vision–ECCV 2012*, pages 774–787. Springer, 2012.
- [4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [5] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *Proceedings of International Conference on Multimedia Retrieval*, page 297. ACM, 2014.
- [6] Karpathy. Multi-layer recurrent neural networks (lstm, gru, rnn) for character-level language models in torch. <https://github.com/karpathy/char-rnn>. Accessed April 4, 2016.
- [7] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. Hauptmann. Complex event detection via multi-source video attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2627–2633, 2013.
- [8] V. Mezaris, A. Dimou, and I. Kompatsiaris. Local invariant feature tracks for high-level video feature extraction. In *11th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2010, Desenzano del Garda, Italy, April 12-14, 2010*, pages 1–4, 2010.
- [9] I. Mironica, B. Ionescu, J. Uijlings, and N. Sebe. Fisher kernel based relevance feedback for multimodal video retrieval. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 65–72. ACM, 2013.
- [10] A. Nordin. Improving an information retrieval system by using machine learning to improve user relevance feedback. 2016.
- [11] P. Over, J. Fiscus, G. Sanders, D. Joy, M. Michel, G. Awad, A. Smeaton, W. Kraaij, and G. Quénot. Trecvid 2014—an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, page 52, 2014.
- [12] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.
- [13] J. J. Rocchio. Relevance feedback in information retrieval. 1971.
- [14] K. Schoeffmann, D. Ahlström, W. Bailer, C. Cobârzan, F. Hopfgartner, K. McGuinness, C. Gurrin, C. Frisson, D.-D. Le, M. Del Fabro, et al. The video browser showdown: a live evaluation of interactive video search tools. *International Journal of Multimedia Information Retrieval*, 3(2):113–127, 2014.
- [15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [16] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.
- [17] M. Wang, Y. Gao, K. Lu, and Y. Rui. View-based discriminative probabilistic modeling for 3d object retrieval and recognition. *IEEE Transactions on Image Processing*, 22(4):1395–1407, 2013.
- [18] Z. Zha, M. Wang, Y. Zheng, Y. Yang, R. Hong, and T. Chua. Interactive video indexing with statistical active learning. *IEEE Trans. Multimedia*, 14(1):17–27, 2012.
- [19] H. Zhang, X. Shang, W. Yang, H. Xu, H. Luan, and T.-S. Chua. Online collaborative learning for open-vocabulary visual classifiers. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
- [20] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua. Discrete collaborative filtering. In *Proc. of SIGIR*, volume 16, 2016.
- [21] H. Zhang, Z.-J. Zha, S. Yan, J. Bian, and T.-S. Chua. Attribute feedback. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 79–88. ACM, 2012.
- [22] X. Zhang, Y. Yang, Y. Zhang, H. Luan, J. Li, H. Zhang, and T. Chua. Enhancing video event recognition using automatically constructed semantic-visual knowledge base. *IEEE Trans. Multimedia*, 17(9):1562–1575, 2015.
- [23] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.