Collaborative Video Re-indexing via Matrix Factorization

MING-FANG WENG and YUNG-YU CHUANG, National Taiwan University

Concept-based video indexing generates a matrix of scores predicting the possibilities of concepts occurring in video shots. Based on the idea of collaborative filtering, this paper presents unsupervised methods to refine the initial scores generated by concept classifiers by taking into account the concept-to-concept correlation and shot-to-shot similarity embedded within the score matrix. Given a noisy matrix, we refine the inaccurate scores via matrix factorization. This method is further improved by learning multiple local models and incorporating contextual-temporal structures. Experiments on the TRECVID 2006–2008 datasets demonstrate relative performance gains ranging from 13% to 52% without using any user annotations or external knowledge resources.

Categories and Subject Descriptors: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*

General Terms: Algorithms, Experimentation.

Additional Key Words and Phrases: Multimedia content analysis, semantic video indexing, concept detection, unsupervised learning, TRECVID

ACM Reference Format:

Weng, M.-F. and Chuang, Y.-Y. 2011. Collaborative Video Re-indexing via Matrix Factorization. ACM Trans. Multimedia Comput. Commun. Appl. 2, 3, Article 1 (May 2010), 0 pages.

DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The advancement of content acquisition devices and data storage technologies in recent years has resulted in rapid growth in the number of videos. In particular, with the popularity of Internet sharing platforms like YouTube has come an exponential number of publicly accessible videos. The resulting broad availability of videos has lead to a strong demand for effective and efficient access of videos [Lew et al. 2006]. *Query-by-concept*-based search addresses this issue by allowing users to find videos that are conceptually similar to the search query. The success of this paradigm however depends heavily on concept-based video annotation and indexing to identify whether the pre-defined concepts are present in a video shot. Unfortunately, because of the discrepancy between low-level feature descriptors and

This work was supported by the National Science Council of Taiwan, R.O.C., under NSC grants 99-2628-E-002-015, 099-2811-E-002-096, and 99-2622-E-002-026-CC2.

Author's address: M.-F. Weng and Y.-Y. Chuang, Department of Computer Science & Information Engineering, National Taiwan University, Taipei Taiwan 106; email: mfueng@cmlab.csie.ntu.edu.tw, cyy@csie.ntu.edu.tw.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

high-level semantic concepts, concept-based video indexing remains a critical obstacle to the success of query-by-concept [Snoek and Worring 2009].

A popular and standardized approach to detect the occurrence of concepts in shots is to employ machine learning techniques to train concept-specific detectors in a supervised manner [Snoek et al. 2006; Yanagawa et al. 2007; Jiang et al. 2007b]: these detectors are constructed by learning frequent feature patterns associated with the corresponding concepts. Recent research has shown that ensemble classifiers help to improve the accuracy of semantic concept detection [Jiang et al. 2008; Snoek et al. 2009]. Such approaches increase the diversity of classifiers. For example, one can sample a subset of annotated training examples, use a part of feature elements, or use different distance metrics in the learning phase. Given the individual classification results, simply combining them—e.g., taking the average of the classifier scores—substantially improves detection accuracy. However, while such approaches yield improved accuracy, many concepts are still hard to detect when not enough labeled examples are available, even when utilizing a number of diverse classifiers.

Another type of promising approach involves refining the scores of concept-specific detectors for better detection accuracy (re-indexing), often by exploring contextual correlation and temporal coherence [Naphade and Huang 2001; Naphade et al. 2002; Jiang et al. 2007a; Jiang et al. 2009]. By contextual correlation, we mean the co-occurrence between semantic concepts in a shot; temporal coherence relates to a single concept that occurs in multiple neighboring shots. As discussed in Liu et al.'s paper [2009], existing re-indexing methods which exploit contextual or temporal relations to refine the initial scores can be classified into three categories, according to the extra knowledge and resources involved. (1) Self-refining (unsupervised learning) methods use only initial scores to explore informative cues to refine indexing performance [Kennedy and Chang 2007; Yang et al. 2009]. (2) Example-based refining (supervised learning) methods discover relationships from user-provided examples and annotations to improve initial results [Liu et al. 2008; Weng and Chuang 2008]. (3) Crowd refining methods leverage external knowledge (e.g., WordNet and Wikipedia), heterogeneous resources (e.g., social media such as Flickr images and tags), or search engines (e.g., Google, Yahoo!, and Bing) for better performance [Aytar et al. 2008].

Generally speaking, in refining an initial result obtained from a set of independent concept detectors, example-based refining methods often yield greater performance gains than self-refining techniques if the discrepancy between training and test data distributions is small. However, supervised example-based methods rely heavily on expensive user annotation to acquire reliable knowledge for video re-indexing. From this perspective, self-refining methods are good because they do not require expensive manual annotation. Crowd refining methods, which utilize web-based or other easily accessible knowledge resources, share this advantage. However, these approaches suffer from potential cross-domain problems when the data distributions of the external sources do not match those of the target domain. Such domain gaps can seriously degrade performance [Jiang et al. 2008; Jiang et al. 2009]. Unsupervised self-refining methods do not have these problems because they acquire the knowledge directly from the input initial scores themselves. Furthermore, unsupervised methods are often more flexible, as they require only the initial scores as inputs.

This paper presents an unsupervised video re-indexing method which refines the detection scores generated by concept classifiers by exploiting structures embedded within the score matrix based on the idea of *collaborative filtering* [Su and Khoshgoftaar 2009]. Collaborative filtering has been used in many applications, two notable examples of which are recommender systems [Koren et al. 2009] and image de-noising [Ji et al. 2010]. In recommender systems, collaborative filtering utilizes user-user similarity and item-item correlation to predict missing preferences. Users with similar purchase patterns in the past will likely buy the same items in the future. Similarly, some items are often purchased together with other correlated items. Such patterns can often be discovered from the given sparse preference

matrix. We observe that the occurrence patterns of semantic concepts in videos closely match this characteristic of recommender systems. We treat video shots as users and concepts in the lexicon as items. There exists shot-to-shot similarity; shots with similar scores on a set of concepts are likely to behave similarly for another set of concepts. In addition, there is concept-to-concept correlation; many concepts are dependent to each other [Snoek and Worring 2009]. When a concept occurs, this often signifies the presence of other concepts: such correlation is also referred as a contextual or semantic relationship. Despite these similarities, the preference matrix is often sparse and accurate while the score matrix is dense and noisy.

Another notable application of collaborative filtering is image de-noising. In this setting, similar image patches are collected and collaboratively fitted with a linear model. The collaboratively learned linear model is then used to predict smooth patches without noise. Unlike recommender systems which adopt a single global model, image de-noising is usually performed by learning multiple local models, each for an individual group of similar patches. Local modeling is preferred, because modeling all the patches with a single linear model could lead to a compromised global structure, thus destroying salient local structures. Inspired by the concept of patch-based image de-noising, we divide the score matrix into several local blocks according to the observed temporal and contextual relationships, where a block corresponds to the concept presence profile of a clip (a sequence of adjacent shots) for a group of relevant concepts. We cluster similar blocks and fit a linear model for each cluster of similar blocks. This strategy enables us to incorporate not only contextual structures but also temporal ones, thus further improving the performance of semantic video indexing.

The main contributions of this paper are (1) the application of collaborative filtering to the video re-indexing problem and a demonstration of its effectiveness; and (2) the concept of clips, which incorporate temporal information into the collaborative filtering framework. In addition, this paper describes several ways to apply localized collaborative filtering, including selecting a subset of relevant concepts, grouping similar profiles, and prediction combination, to further improve the performance. As a result, the proposed collaborative video re-indexing method represents the first unsupervised approach that simultaneously utilizes both contextual and temporal information for video re-indexing. Our method is effective, improving the baselines 13.2%$\sim$51.7%, measured by mean inferred average precision, without using any external knowledge resources or user annotations.

2. RELATED WORK

The fundamental task of semantic video indexing can be formulated as a set of pattern recognition problems, in which various supervised learning methods, e.g., *support vector machines*, are used to build feature-based concept classifiers [Amir et al. 2005; Jiang et al. 2007b]. Although many techniques for feature extraction, feature fusion, and classifier combination have been proposed to improve detection accuracy [Snoek and Worring 2009], unfortunately, most concepts are still not easily detected even after utilizing a number of diverse classifiers in an ensemble classifier [Snoek et al. 2009]. A recent trend for concept detection research is to utilize the consolidation of kernel-based learning. These methods, however, only take into account patterns of low-level features associated with specific concepts. This is likely to yield suboptimal performance.

In recent years, much research has been focused on adding high-level relationships to the inference process by leveraging context. In their work, Jiang et al. [2007a] explore contextual relations by capturing co-occurrences between concepts using user-labeled annotations. Recently, Jiang et al. [2009] proposed *semantic diffusion* which gradually enhances the consistency of detection scores among concepts. The investigation into integrating contextual correlation and temporal coherence was pioneered by Naphade and Huang [2001]. Qi et al. [2007] proposed a *correlative multi-label* framework to simultaneously explore interactions between concepts and mappings between low-level features and

Table I. Categorization of relation learning approaches for semantic video re-indexing; g-CVR, c-CVR and l-CVR represent three variants of the proposed collaborative video re-indexing approach.

Learning sources \ Target relations	Contextual relation	Contextual-temporal relation
User-provided annotations	Naphade and Huang 2001; Naphade et al. 2002; Jiang et al. 2007a; Qi et al. 2007; 2008; Liu et al. 2008; Weng and Chuang 2008; Jiang et al. 2009.	Naphade and Huang 2001; Naphade et al. 2002; Liu et al. 2008; Qi et al. 2008; Weng and Chuang 2008.
Detector-generated predictions	Kennedy and Chang 2007; Yang et al. 2009; Jiang et al. 2009; **g-CVR; c-CVR.**	**l-CVR.**
External knowledge resources	Aytar et al. 2008.	N/A

single concepts. They further incorporated temporal information with a temporal kernel [Qi et al. 2008]. While these methods yielded good results in experiments with dozens of concepts, they have become impractical for scenarios with a greater number of concepts because of the inherent complexity in learning complete relationships.

To allow for tractable computation of relation modeling within a large-scale concept ontology, rather than explicitly learning a single globally optimal model for all concepts, more efficient alternatives have emerged that entail the implicitly learning of multiple local models for each single target concept respectively [Kennedy and Chang 2007; Liu et al. 2008; Yang et al. 2009]. For example, *multi-cue fusion* as proposed by Weng and Chuang [2008] is a data-driven approach to implicitly learning semantic and temporal relations from annotations for each concept. The relation learning process is separated from detector learning, greatly reducing its complexity. The learnt relationships are then used to refine the initial detection results. For implicit methods, the low computational cost involved in the learning stage makes the approach scalable to the number of concepts and more practical for large-scale semantic concept detection systems. In view of this, our focus here is to develop an implicit method, aiming to exploit benefits from the integration of contextual and temporal relations in an efficient and effective way.

As shown in Table I, these methods can be categorized according to their learning sources (user-provided annotations, detector-generated predictions, or external knowledge resources) and the target relations (contextual or contextual-temporal relations). For example, Kennedy and Chang [2007] proposed a reranking approach to exploit inter-concept relationships and adopted a classification-based method to learn concept-specific models using detector-generated predictions. Yang et al. [2009] shared a similar idea but proposed an ordinal reranking algorithm. The post-filtering method proposed by Liu et al. [2008] discovers and models both contextual and temporal relations from user-provided annotations. In general, exploiting knowledge from a manually annotated corpus often yields better performance than mining information from noisy and relatively unreliable scores [Jiang et al. 2009]. However, as mentioned in Section 1, unsupervised relation learning has the advantages of being free from the potential domain-shift problem and does not require expensive labeled training data. Furthermore, it is applicable in a wider range of applications, such as image tagging and video search [Kennedy and Chang 2007; Yang et al. 2009]. As seen in Table I, this paper fills the gap by proposing the first unsupervised method that simultaneously explores contextual and temporal relationships.

3. PROBLEM STATEMENT

We start by describing the video re-indexing problem and defining notation used in this paper. Assume that we have a well-defined lexicon $\mathcal{C}$ of m semantic concepts to be used to index videos, where $\mathcal{C}=\{c_1, c_2, \ldots, c_m\}$. Let $\mathcal{S}=\{s_1, s_2, \ldots, s_n\}$ be a collection of the n video shots to be indexed; without loss of generality, the indices are specified according to their temporal order, i.e., s_{t-1} is the shot previous to

s_t. Given a set of concept-specific detectors, e.g., supervised visual feature classifiers [Yanagawa et al. 2007; Jiang et al. 2007b] or unsupervised text-based relevance scoring functions [Adams et al. 2003], the output detection score $y_{t,i}$ indicates the possibility that concept c_i occurs in shot s_t. By concatenating the detection scores of all of the concepts that corresponding to shot s_t, we obtain an m-dimensional row vector $\mathbf{y}_t = [y_{t,1}, y_{t,2}, \ldots, y_{t,m}]$. This can be used to index s_t for concept-based applications. Thus, a conventional *video indexing* algorithm yields the matrix $\boldsymbol{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_n]$ as the vertical concatenation of those row vectors; this matrix expresses the predicted likelihood for the whole video set based on the lexicon.

Unfortunately, indexing results that consist of scores generated using concept detectors usually yield unsatisfactory performance since, in general, concept detectors only utilize cues within a single shot and a single concept, ignoring intrinsic dependencies among semantic concepts and among video shots [Weng and Chuang 2008; Snoek and Worring 2009]. Therefore, the goal of *video re-indexing* is to explore inter-concept and inter-shot cues beyond low-level features to yield more accurate indexing results. More specifically, given the initial score matrix $\boldsymbol{Y}$, video re-indexing is an attempt to find a refined score matrix $\overline{\boldsymbol{Y}}$ that yields better performance than $\boldsymbol{Y}$. As discussed in Section 1, while approaches that either mine relationships from manually labeled groundtruth or learn knowledge from external heterogeneous resources improve indexing quality, they are only practical for a limited set of scenarios and applications. In this paper, we focus on leveraging information within the initial scores to re-indexing video shots using the collaboratively refined ones.

4. LATENT FACTOR MODELS

Our collaborative video re-indexing method is based on the latent factor model that is widely used in recommender systems [Su and Khoshgoftaar 2009]. Latent factor models attempt to explain the scores by characterizing both concepts and shots on factors inferred from the patterns within the initial score matrix. The discovered factors may have intuitive meanings, or they could be completely uninterpretable. Matrix factorization is one of the most successful realizations of latent factor models [Koren et al. 2009]. Assuming a p-factor model, given the score matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times m}$, this approach finds two matrices $\boldsymbol{U} \in \mathbb{R}^{n \times p}$ and $\boldsymbol{V} \in \mathbb{R}^{m \times p}$ such that

$$\boldsymbol{Y} \approx \boldsymbol{U}\boldsymbol{V}^T. \tag{1}$$

This provides a low-rank approximation to the matrix $\boldsymbol{Y}$ since the parameter p is often smaller than the rank of $\boldsymbol{Y}$. It essentially maps both shots and concepts to a joint latent factor space in which we can directly measure their similarity. In this lower dimensional space, each shot s_t is represented by $\mathbf{u}_t \in \mathbb{R}^p$ (the t-th row of $\boldsymbol{U}$) and each concept c_i is modeled as $\mathbf{v}_i \in \mathbb{R}^p$ (the i-th row of $\boldsymbol{V}$). The refined score of shot s_t regarding concept c_i is just the inner product of their corresponding representations, namely, $\overline{y}_{t,i} = \mathbf{u}_t \mathbf{v}_i^T$. Such scores are refined because, ideally, the low-rank approximation removes noise while preserving the structures within the data.

To understand why the use of matrix factorization is helpful for the removal of inaccurate entries of the initial score matrix, we take the given $\boldsymbol{Y}$ as a noisy spatial-temporal image. The image is noisy because the detectors are not perfect. Fortunately, just like regular images, noise can be reduced to some degree by exploring the structures among detection scores. Because many video shots share similar content and concepts are often correlated, two types of structures exist within the matrix and can be used to refining scores. The first type of structure is *shot-to-shot similarity*: if the content of two video shots is very similar, they should receive similar scores for all concepts and have similar shot profiles. Therefore, if two shots have similar detection scores for many concepts, their scores for other concepts should not differ much. In other words, there is a dependence among rows within the score matrix. The second type of structure is *concept-to-concept correlation*: relevant concepts should exhibit similar

behavior for shots. For example, if the detection scores of *car* and *road* for most shots exhibit strong correlation, their scores should also show the similar correlation for the others. Similarly, the detection scores of *urban* and *studio* could complement each other. In other words, there is a dependence among columns within the score matrix. Therefore, the score matrix can be de-noised by finding a lower-rank approximation.

Instead of factorizing the matrix directly using singular value decomposition, to avoid overfitting, the factorization is formulated as a regularized problem:

$$\arg\min_{\boldsymbol{U},\boldsymbol{V}} \left\| \boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{Y} \right\|_F^2 + \frac{\lambda}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2 \right), \tag{2}$$

where $\lambda > 0$ is a regularization parameter and $\|\cdot\|_F$ is the Frobenius norm. The regularization term $\frac{\lambda}{2}\left(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2 \right)$ restricts the domains of the learned factorization $\boldsymbol{U}$ and $\boldsymbol{V}$ in order to avoid overfitting the initial scores. Another effect is that, by penalizing the magnitudes of $\boldsymbol{U}$ and $\boldsymbol{V}$, the regularizer tends to minimize the trace-norm, leading to a low-rank factorization [Rennie and Srebro 2005]. To speed up the process, a user-specified constant p, the estimated number of factors, is often given to constrain matrix sizes. Note that, in this case, the choice of p does not necessarily match the intrinsic dimensionality of the matrix. We seek to specify a p large enough to retain the intrinsic properties of the matrix but small enough to allow for more efficient computation.

The formulation of our low-rank approximation is unlike the robust principal component analysis approach [Candès et al. 2009], which is more appropriate when the sparse data samples are interpreted as outliers. In contrast, in our case, most of the detection scores are contaminated. To account for this, we turn to an approximation to noisy data in a least-squares sense. Rather than solving for Equation 2, we found that a balance of regularization coefficients for both matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ yields better performance. The actual objective function that we minimize is

$$J(\boldsymbol{U},\boldsymbol{V}) = \left\| \boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{Y} \right\|_F^2 + \lambda_u \|\boldsymbol{U}\|_F^2 + \lambda_v \|\boldsymbol{V}\|_F^2, \tag{3}$$

in which $\lambda_u = m\lambda/2p$ and $\lambda_v = n\lambda/2p$. Because of the product of unknowns $\boldsymbol{U}\boldsymbol{V}^T$, Equation 3 is not convex. Thus, we use the alternating least squares algorithm to solve it; by fixing one of the unknowns, the optimization becomes quadratic and can be solved optimally. We first randomly initialize two matrices, and then update $\boldsymbol{U}$ and $\boldsymbol{V}$ alternatively and iteratively until convergence using the conjugate gradient algorithm implemented by Rennie[1]. The partial derivatives of the objective function are

$$\frac{1}{2}\frac{\partial J}{\partial \boldsymbol{U}} = \left(\boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{Y} \right)\boldsymbol{V} + \lambda_u \boldsymbol{U}, \tag{4}$$

$$\frac{1}{2}\frac{\partial J}{\partial \boldsymbol{V}} = \left(\boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{Y} \right)^T \boldsymbol{U} + \lambda_v \boldsymbol{V}. \tag{5}$$

After finding the solution of $\boldsymbol{U}$ and $\boldsymbol{V}$, the refined score matrix $\overline{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{V}^T$ is calculated as the result of re-indexing.

Finally, in our experience we have found that it is critical to the performance of the approach to conduct a simple normalization on an input score matrix before performing matrix factorization. More specifically, we form a normalized matrix $\widetilde{\boldsymbol{Y}}$ consisting of zero-mean and unit-variance column vectors. That is, if μ_i and σ_i are the mean and standard deviation of the detection scores for concept c_i, then we have the normalized score $\widetilde{y}_{t,i} = \frac{1}{\sigma_i}(y_{t,i} - \mu_i)$. The main reason for this normalization is that for many concepts in the lexicon, the distributions of the positive and negative examples are extremely

[1]Downloaded from http://people.csail.mit.edu/jrennie/matlab/.

Algorithm 1 Global collaborative video re-indexing.
$\overline{\boldsymbol{Y}} = \textbf{g-CVR}(\boldsymbol{Y}, p, \lambda)$. Given score matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times m}$ and two parameters, p and λ, which are respectively the number of latent factors and the regularization coefficient, return refined score matrix $\overline{\boldsymbol{Y}} \in \mathbb{R}^{n \times m}$.

1: normalize $\boldsymbol{Y}$ to yield $\widetilde{\boldsymbol{Y}}$
2: $\lambda_u = \frac{m}{2p}\lambda$, $\lambda_v = \frac{n}{2p}\lambda$
3: generate random matrices $\boldsymbol{U} \in \mathbb{R}^{n \times p}$, $\boldsymbol{V} \in \mathbb{R}^{m \times p}$ as the initial guesses
4: solve the unconstrained minimization problem with the conjugate gradient method:
$(\boldsymbol{U}, \boldsymbol{V}) = \arg\min_{\boldsymbol{U},\boldsymbol{V}} \left\| \boldsymbol{U}\boldsymbol{V}^T - \widetilde{\boldsymbol{Y}} \right\|_F^2 + \lambda_u \|\boldsymbol{U}\|_F^2 + \lambda_v \|\boldsymbol{V}\|_F^2$
5: **return** $\overline{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{V}^T$

imbalanced. That is, the examples predicted to be negative are usually distributed in a much denser domain and are likely to dominate the ones with positive predictions. Furthermore, concept-specific detectors can be biased, i.e., detectors for some concepts consistently output higher scores. Thus, to balance the emphasis on all examples and all concepts in the optimization process, we normalize the score matrix so that the scores for each concept have similar ranges and statistical properties. We call the approach described in this section *global collaborative video re-indexing* (g-CVR) and summarize it in Algorithm 1.

5. LOCALIZED COLLABORATIVE VIDEO RE-INDEXING

Although the global method described in Section 4 works reasonably well, the use of a single global model for the entire score matrix restricts the effectiveness of the method, as compromises must be made to fit everything into a single linear model. In addition, temporal structures of videos are not explored at all because row (shot) order is ignored during matrix factorization. In this section we describe a novel algorithm to address these two issues. With this algorithm we take one step further the idea of collaborative video re-indexing by matrix factorization, yielding additional performance gains.

5.1 Overview

When a single linear model does not adequately represent the underlying structure, a more effective alternative is to assume the structure is locally linear and thus to model it with a set of linear models. This is the basic idea behind many nonlinear dimension reduction methods, including locally linear embedding [Roweis and Saul 2000]. Similarly, for video re-indexing, we would like to use the idea of multiple linear models for collaborative filtering.

There are many reasons why a single global linear model might be insufficient. First, it is less likely that all concepts are relevant. For example, the column vector for the occurrence of the concept *face* seems largely independent of the one for *animal*. In contrast, there is a strong dependence among the column vectors for *fire*, *smoke*, and *explosion* as well as those for *car*, *road*, and *urban*. Therefore, to better exploit concept-to-concept correlation, it would be more effective to pre-select a set of highly relevant concepts and to learn a factor model from this set instead of learning from all concepts. Second, while it is unrealistic to require that all video shots are similar in content, it is reasonable to expect to see strong shot-to-shot similarity among neighboring shots within a short time frame. Therefore, it is more effective to discover patterns from a handful of neighboring shots instead of mining them from the whole video set.

To exploit the structures within a set of neighboring shots for a set of relevant concepts, temporally, we define a clip as a set of neighboring shots in the video temporal domain; contextually, we define a concept group as a set of relevant concepts. Therefore, a submatrix of the score matrix $\boldsymbol{Y}$ corresponds

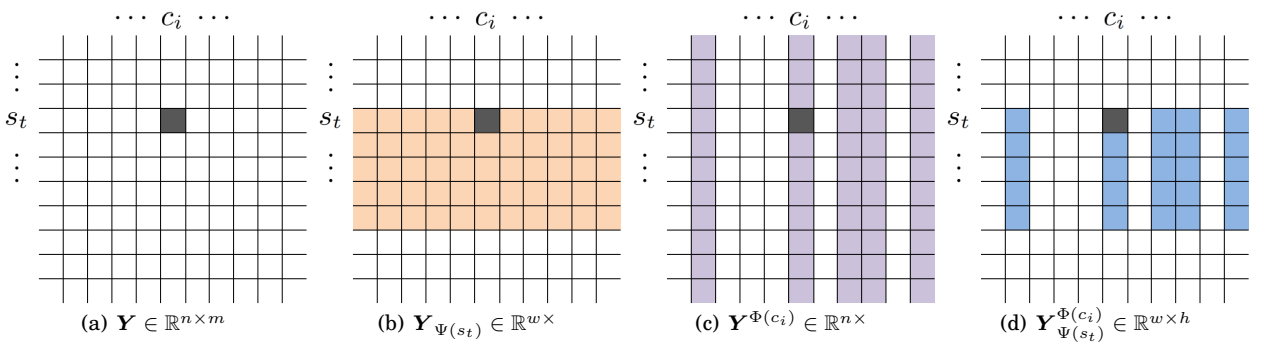


Fig. 1. To create a profile for a clip for a concept group, starting from shot s_t and centered at concept c_i, we define row and column projection functions Ψ and Φ of score matrix $\boldsymbol{Y}$ to select the scores for a span of shots and for a subset of concepts, respectively. (a) The initial score matrix. Each entry corresponds to the possibility of concept c_i occurring in shot s_t. (b) Submatrix $\boldsymbol{Y}_{\Psi(s_t)}$ corresponding to the profile for clip $\Psi(s_t)$ with respect to the whole lexicon. (c) Submatrix $\boldsymbol{Y}^{\Phi(c_i)}$ corresponding to the profile for video shots for the concept group $\Phi(c_i)$. (d) Submatrix $\boldsymbol{Y}^{\Phi(c_i)}_{\Psi(s_t)}$ corresponding to the clip profile of clip $\Psi(s_t)$ with respect to the concept group $\Phi(c_i)$.

to the detection scores of a clip regarding for a concept group. We use the notation $\boldsymbol{Y}^A_B \in \mathbb{R}^{|B|\times|A|}$ to represent a submatrix formed by selecting the rows specified in B and the columns specified in A from $\boldsymbol{Y}$, where A and B are two sets of indices representing a clip and a concept group respectively. The functions $\Psi(.)$ and $\Phi(.)$ are used to generate sets A and B. Given shot s_t, function $\Psi(s_t)$ returns the clip composed of s_t and the following $w-1$ shots; for concept c_i, function $\Phi(c_i)$ returns the concept group composed of c_i and its relevant concepts (relevance is defined in Section 5.2). Assume that the number of concepts relevant to c_i is $h-1$. Thus, as shown in Figure 1, $\boldsymbol{Y}_{\Psi(s_t)} \in \mathbb{R}^{w\times m}$ contains w rows corresponding to a clip and $\boldsymbol{Y}^{\Phi(c_i)} \in \mathbb{R}^{n\times h}$ selects h columns for a concept group. Finally, submatrix $\boldsymbol{Y}^{\Psi(s_t)}_{\Phi(c_i)} \in \mathbb{R}^{w\times h}$ represents a local profile for the clip consisting of s_t and its neighboring shots with respect to concept c_i and its relevant concepts.

For a small group of relevant concepts, we expect to discover stronger dependence among neighboring shots. Therefore, to fully exploit the structure embedded in the score matrix, we use a localized contextual-temporal model to jointly refine video indexing results. After extracting submatrix $\boldsymbol{Y}^{\Psi(s_t)}_{\Phi(c_i)}$, which represents the profile of clip $\Psi(s_t)$ with respect to concept group $\Phi(c_i)$, in order to reflect the temporal information, we unroll the submatrix into $(w\cdot h)$-dimensional vector $\mathbf{z}_{s_t,c_i} = \bigvee(\boldsymbol{Y}^{\Psi(s_t)}_{\Phi(c_i)})$, where $\bigvee(\boldsymbol{M})$ is the unrolling operator which concatenates all rows of matrix $\boldsymbol{M}$ into a long row vector. At this point, for a collection of n video shots, we have $n-w+1$ clips and their profile vectors for concept group $\Phi(c_i)$. Unrolling submatrices into profile vectors enables us to stack them into a matrix for factorization. When all of the clip profile vectors are stacked to form a matrix, two extra properties can be exploited. First, we can explore inter-concept correlation not only within a shot but also across several neighboring shots. For example, the concept *airplane takeoff* occurring on a shot is most likely correlated to concept *sky*, which appears in the shot following it. The dependency among column vectors in this matrix confirms the existence of this relationship. Second, we can further assume there are dependencies among row vectors in the matrix. This implies that if most of the components of two clip profiles are similar, the remaining components for these profiles should be assigned similar scores. This property, which we term *clip-to-clip similarity*, is different from shot-to-shot similarity, because it not

only relies on the context within a single shot but also takes into account cues of neighboring shots that are potentially more robust.

To better discover structures, we group these $n-w+1$ vectors of the dataset into K clusters of similar profiles. For each cluster $P^{(k)}$, the clip profile vectors of this cluster are stacked together to form matrix $\boldsymbol{Z} \in \mathbb{R}^{n_k \times w \cdot h}$, where n_k is the size of $P^{(k)}$. This matrix is then factorized using Equation 3 to obtain the refined matrix $\overline{\boldsymbol{Z}}$. As $\boldsymbol{Z}$ is formed of clips with similar profiles, the low-rank approximation $\overline{\boldsymbol{Z}}$ is likely sufficient to model the intrinsic structure of $\boldsymbol{Z}$. Each row of $\overline{\boldsymbol{Z}}$ represents a refined clip profile; we can replace the original corresponding clip profile in $\boldsymbol{Y}$ with this refined one. The process is repeated by picking one concept at a time, until all of the elements of the score matrix have been refined at least once. Some elements might be processed more than once, in which case we take the average of all of the refined scores for these elements.

We call this method *localized collaborative video re-indexing* (l-CVR). Algorithm 2 summarizes the l-CVR method and Figure 2 illustrates its process. Note that, in Algorithm 2, we define $\bigwedge(\mathbf{z}_{s_t,c_i})$ as an operator which inverses the unrolling operation by projecting elements of $\mathbf{z}_{s_t,c_i}$ back to their positions in the submatrix $\boldsymbol{Y}^{\Psi(s_t)}_{\Phi(c_i)}$. The details for the method are described in the following sections.

It is also worth mentioning that setting $w=1$ in l-CVR ignores temporal structures. In this case, the algorithm does not assume any temporal structures and explores only contextual relations. We call this *contextual collaborative video re-indexing* (c-CVR). Note that, although temporal information is ignored, as later shown in Section 6, c-CVR often yields better performance than g-CVR because it fits multiple linear models and takes into account only a small set of relevant concepts. This approach may be useful for applications such as image re-tagging [Chua et al. 2009] in which there is no temporal information; for such applications discovering contextual structures is still beneficial.

5.2 Contextual and Temporal Neighborhoods

An effective neighborhood system facilitates the discovery of matrix structure. For temporal neighborhoods, an intuitive solution is to use temporally neighboring shots, which are likely to contain similar contents. The function $\Psi(s_t)$ simply returns the set of w consecutive shots $\{s_t, s_{t+1}, \ldots, s_{t+w-1}\}$. Note that we include the shots following s_t but not the ones preceding it. Since the shot s_t is used only for indexing clips, it does not matter whether we put s_t at the front or the middle of the clip. Both placements result in similar sets of clips, the sole difference being the range of valid shot indices. For example, the clip set $\{\Psi(s_t) \mid 1 \le t \le n-w+1\}$ we used is equivalent to the clip set $\left\{\Psi'(s_t) \mid 1+\lfloor \frac{w}{2} \rfloor \le t \le n-\lfloor \frac{w-1}{2} \rfloor\right\}$ where $\Psi'(s_t) = \left\{s_{t-\lfloor \frac{w}{2} \rfloor}, s_{t-\lfloor \frac{w}{2} \rfloor+1}, \ldots, s_{t+\lfloor \frac{w-1}{2} \rfloor}\right\}$.

For contextual neighborhoods, for a given target concept c_i, we would like to discover a small number of concepts which are relevant to it. To this end, we calculate the Pearson product-moment correlation coefficient (PMCC) to measure the correlation between two concepts c_i and c_j, which is defined as

$$r(c_i, c_j) = \frac{\sum_{t=1}^{n} \widetilde{y}_{t,i}\, \widetilde{y}_{t,j}}{n-1}, \tag{6}$$

where $r(c_i, c_j)$ yields a value between $+1$ and -1. It should be noted that the detection scores used in Equation 6 have been normalized, as described in Section 4. In order to build a concept group for c_i, we select the $h-1$ concepts with the largest correction coefficients with c_i as the contextual neighbors. These concepts and the concept c_i form a concept group, namely $\Phi(c_i)$.

5.3 Grouping Similar Profiles

Due to the diversity of video content, the use of concept groups may not be sufficient to robustly discover contextual structures. For example, concept *car* could be judged relevant to either of the concept sets

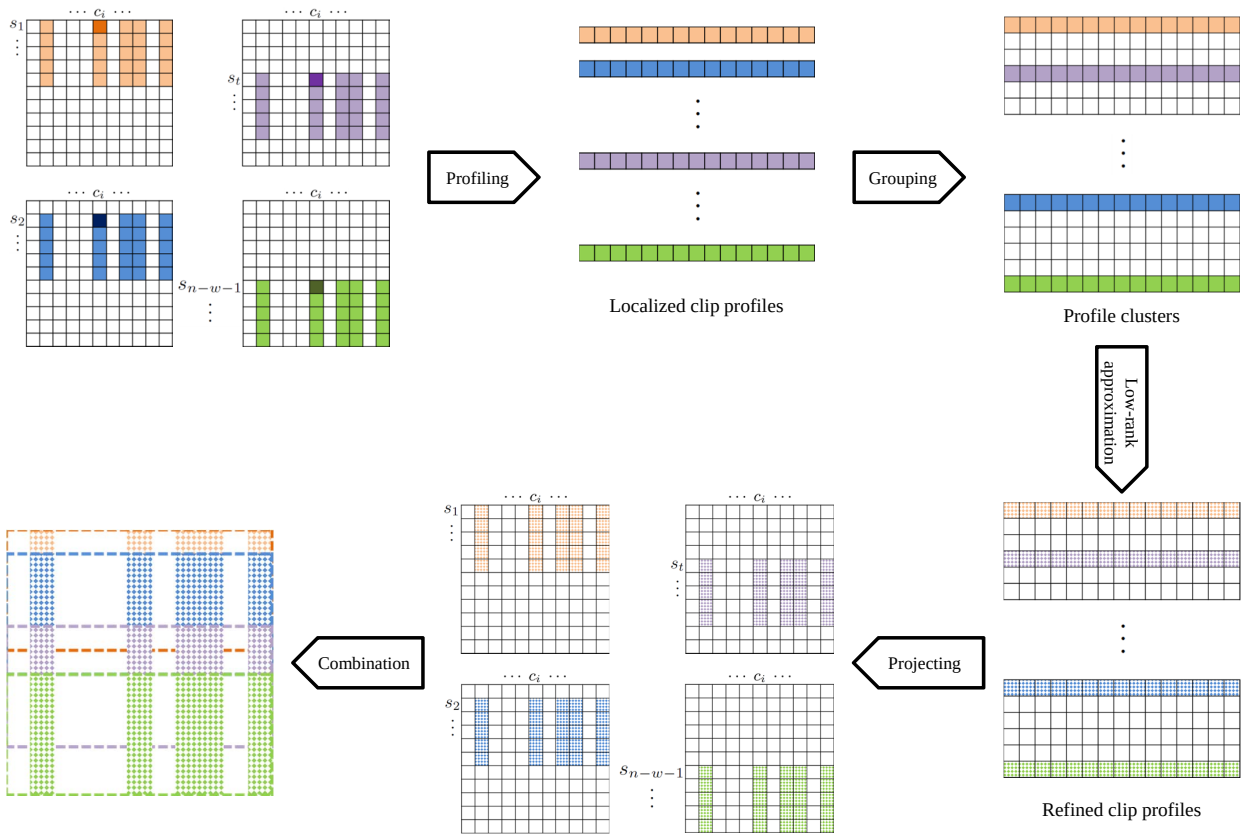


Fig. 2. Illustration of the localized collaborative re-indexing method. Submatrices for a concept group are collected and unrolled into localized clip profiles, which are grouped into clusters. The clip profiles of each cluster are stacked together to form a matrix. These matrices are factorized to yield low-rank approximations. The new scores are then projected back to their original positions in the score matrix. Finally, the refined scores are averaged to yield the final scores.

{*building*, *streets*, *traffic*} or {*trees*, *mountain*, *fields*}, depending on whether the video clip describes a cityscape or an urban scene. Therefore, to better discover an underlying linear model, we seek to perform collaborative filtering on clips with similar profiles. One way of achieving this is to select a pivot clip and then find its closest matchings. This process is repeated until no clip remains unprocessed. This way of find closest matches is however more time-consuming because we need to compare with all of the remaining clips. Instead, we use the k-means algorithm to group similar clip profiles together for better efficiency. The number of clusters K is determined by $K = \lceil \frac{n}{N_c} \rceil$, where N_c is the expected size of a cluster. In our current implementation, we empirically use $N_c = 2000$ for all experiments.

5.4 Optimization and Combination

As shown in Algorithm 2, after clustering, for each cluster $P^{(k)}$, we stack all clip profile vectors in $P^{(k)}$ to form matrix $\boldsymbol{Z} \in \mathbb{R}^{n_k \times (w \cdot h)}$ in which $n_k = \left| P^{(k)} \right|$. Matrix $\boldsymbol{Z}$ is then factorized using the method described in Section 4. We thus obtain the lower-rank approximated matrix $\overline{\boldsymbol{Z}}$, each row of which corresponds to a vector representing the refined clip profile $\bar{\mathbf{z}}_{s_t,c_i}$. Thus, we update the refined matrix by adding its rolled version $\bigwedge(\bar{\mathbf{z}}_{s_t,c_i})$ to the refined score matrix $\overline{\boldsymbol{Y}}$ as shown in line 19 of Algorithm 2. At the

Algorithm 2 Localized collaborative video re-indexing.
$\overline{\boldsymbol{Y}} = \textbf{l-CVR}(\boldsymbol{Y}, p, \lambda, w, h)$. Given score matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times m}$ and parameters p (number of latent factors), λ (regularization coefficient), w (number of shots in a clip), and h (number of relevant concepts in a concept group), return refined score matrix $\overline{\boldsymbol{Y}} \in \mathbb{R}^{n \times m}$.

1: $\overline{\boldsymbol{Y}} = \boldsymbol{0}$ ▷ initialization as zero matrix
2: $\boldsymbol{D} = \boldsymbol{0}$ ▷ $\boldsymbol{D}$ is the coverage matrix
3: normalize $\boldsymbol{Y}$ to yield $\widetilde{\boldsymbol{Y}}$
4: add all concepts into a queue
5: **while** (the queue is not empty) **do**
6: select a concept c_i from the queue
7: $P = \phi$ ▷ set of clip profiles
8: **for** $t = 1$ to $n - w + 1$ **do** ▷ scan over all clips
9: $\mathbf{z}_{s_t, c_i} = \bigvee \left(\widetilde{\boldsymbol{Y}}_{\Psi(s_t)}^{\Phi(c_i)} \right)$ ▷ unrolled clip profile vector
10: $P = P \cup \{\mathbf{z}_{s_t, c_i}\}$
11: **end for**
12: partition P into K clusters $P^{(1)}, \cdots, P^{(K)}$
13: **for** each cluster $P^{(k)}$ **do**
14: $\boldsymbol{Z}$ = the matrix formed by stacking vectors in $P^{(k)}$
15: compute λ_u and λ_v
16: $(\boldsymbol{U}, \boldsymbol{V}) = \arg\min_{\boldsymbol{U}, \boldsymbol{V}} \left\| \boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{Z} \right\|_F^2 + \lambda_u \left\| \boldsymbol{U} \right\|_F^2 + \lambda_v \left\| \boldsymbol{V} \right\|_F^2$
17: $\overline{\boldsymbol{Z}} = \boldsymbol{U}\boldsymbol{V}^T$
18: **for** each $\mathbf{z}_{s_t, c_i}$ in $P^{(k)}$ **do** ▷ refine each clip profile
19: $\overline{\boldsymbol{Y}}_{\Psi(s_t)}^{\Phi(c_i)} = \overline{\boldsymbol{Y}}_{\Psi(s_t)}^{\Phi(c_i)} + \bigwedge (\overline{\mathbf{z}}_{s_t, c_i})$ ▷ update scores
20: $\boldsymbol{D}_{\Psi(s_t)}^{\Phi(c_i)} = \boldsymbol{D}_{\Psi(s_t)}^{\Phi(c_i)} + \boldsymbol{1}$ ▷ update coverage
21: **end for**
22: **end for**
23: remove concepts in $\Phi(c_i)$ from the queue
24: **end while**
25: $\overline{\boldsymbol{Y}} = \overline{\boldsymbol{Y}} ./ \boldsymbol{D}$ ▷ take average
26: **return** $\overline{\boldsymbol{Y}}$

same time, coverage matrix $\boldsymbol{D}$ is updated to record how many times each element is covered (line 20). After all of the elements of $\overline{\boldsymbol{Y}}$ have been covered at least once, the final matrix is formed by taking averages, $\overline{\boldsymbol{Y}} = \overline{\boldsymbol{Y}} ./ \boldsymbol{D}$ (line 25); here, the operator $./$ represents the element-wise division of matrices, i.e., $\overline{y}_{i,j} = \overline{y}_{i,j} / d_{ij}$.

6. EXPERIMENTS AND RESULTS

In this section we present the detection performance of the proposed methods and compare this with other competitive approaches. We first describe the settings of the evaluation, including experimental datasets, the baseline detectors, and performance metrics in Section 6.1. Then, in Section 6.2 and Section 6.3 we present the results and the computational times of various methods respectively. Finally, in Section 6.4 we discuss the sensitivity of our collaborative video re-indexing methods to parameter settings.

Table II. The description of the TRECVID datasets used in our experiments. TV06, TV07, and TV08 denote the annual test collections from 2006 to 2008, respectively.

Dataset	TV06	TV07	TV08
Video domain	Broadcast News	Documentary	Documentary
Total number of videos	259	109	219
Length of videos (hours)	159	50	100
Total number of shots	79,484	18,142	35,766

6.1 Experimental Settings

To comprehensively evaluate the proposed approach, we conducted experiments on the TRECVID benchmarks [Smeaton et al. 2006]. TRECVID is an annual activity which encourages research in content-based video analysis and retrieval via an open, metrics-based approach. We performed the evaluations on the official test collections during 2006–2008. These three sets are denoted as TV06, TV07, and TV08, respectively. There are a few differences among these datasets. For example, TV06 is collected from multilingual news videos in American, Arabic, and Chinese broadcast channels, while TV07 and TV08 consist mainly of archival videos in Dutch. Furthermore, the sizes of the datasets are also different. The dataset details are summarized in Table II.

We used the publicly available detection scores of 374 concepts defined in the LSCOM ontology [Naphade et al. 2006; Kennedy and Hauptmann 2006], on TV06, TV07, and TV08 as the initial scores and the baselines. These scores were individually generated by VIREO-374 detectors which averagely combined the outputs of three feature-based classifiers using color, texture, and local keypoint features [Jiang et al. 2007b]. The primitive classifiers in VIREO-374 were trained on the TRECVID 2005 development set, along with the annotation data from the released Columbia374 [Naphade et al. 2006; Yanagawa et al. 2007]. The videos collected in this development set are from the same video domain as TV06; however, this domain is different from those of TV07 and TV08. Thus, to alleviate the domain-change problems on TV07 and TV08, based on TRECVID's collaborative annotation efforts, 36 of the 374 concept detectors in VIREO-374 were retrained using the TRECVID 2007 development data to ensure better detection scores for the corresponding concepts over TV07 and TV08. These scores are all publicly accessible in CU-VIREO374 [Jiang et al. 2008]. The baseline performance for TV06 and TV07 are on the very top (among top 10%) of the TRECVID campaigns. However, even after retraining on the TRECVID 2007 data, the accuracy of the public VIREO-374 benchmark on TV08 is still unsatisfactory. In order to determine how well our approach works on state-of-the-art detection results for TV08, we obtained the detection scores from one of the top performers in TRECVID 2008 [Jiang et al. 2010], which is not publicly available. Basically, most of these scores were generated based on the VIREO-374 detectors as well. However, for better performance, 19 detectors for the evaluated concepts were further retrained on the TRECVID 2008 development data using a similar method to VIREO-374. We denote this baseline as TV08$^+$.

Since it is very time-consuming to label groundtruth for a large set of semantic concepts on a huge test collection, in TRECVID, only a few dozen concepts are selected for evaluation each year. Following the TRECVID convention, we evaluated the performance of indexing results on the 20 officially selected concepts in each corresponding year[2], and used inferred average precision (infAP) [Yilmaz and Aslam 2006] and the average of multiple infAPs (mean infAP) to report the performance on individual concepts and overall performance. Note that, owing to the incomplete assessment of the results, infAP has become the official evaluation metric since TRECVID 2006 [Smeaton et al. 2006].

[2]We dropped the concept *Two_People* which is selected in TRECVID 2008 in performance evaluation because it is not defined in LSCOM and not reported in the VIREO-374 baseline.

6.2 Results

For comparisons, we have implemented several related approaches, including online ordinal reranking (OOR) [Yang et al. 2009], semantic diffusion (SD) [Jiang et al. 2009], and multi-cue fusion (MCF) [Weng and Chuang 2008]. In the OOR method, for each target concept, we selected as features the 25 most peripherally correlated concepts as measured by PMCC and performed reranking on the top 3,000 shots. Due to the sensitivity to parameters such as learning rate, convergence threshold, and fusion weight, for a fair comparison, we have explored a reasonable parameter space and report the overall best performance yielded by a unified setting[3]. In the SD method, we have slightly modified the authors' released package[4] so that the concept affinities are calculated according to the detection scores on each year's test data. Since the default parameters have been tested by the authors on the same datasets and they generally yield good performance, we report the performance using these default settings. The above two methods are unsupervised video re-indexing methods which explore contextual information and learn semantic relationships. To the best of our knowledge, as shown in Table I, there is no other approach that simultaneously utilizes contextual-temporal information in an unsupervised manner. Thus, as a reference, we have implemented the MCF method which explores the semantic and temporal relationships from manually annotated groundtruth[5].

The parameters in the proposed method include the clip length w, the concept group size h, the regularization coefficient λ, and the number of latent factors p. We empirically determined their proper values and discuss the parameter sensitivity for the proposed method in Section 6.4. The results listed here used $\{w=1, h=374, p=200\}$ for g-CVR, $\{w=1, h=25, p=20\}$ for c-CVR, $\{w=10, h=20, p=50\}$ for l-CVR, and $\log_2(\lambda)=2.5$ for all experiments.

Table III displays the overall performance gains over the VIREO-374 baselines on TV06, TV07, TV08, and TV08$^+$, when using our g-CVR, c-CVR, and l-CVR approaches, and comparisons with the OOR, SD, and MCF methods. When taking into account only contextual information, the proposed g-CVR method outperforms the other methods (OOR and SD) in most cases. The only exception is SD on TV08, but the difference is negligible. The localized CVR which uses only contextual relations (c-CVR) extends the performance gain yielded by g-CVR. Although g-CVR is slightly better than c-CVR on TV06, for the other three sets, c-CVR yields substantially higher accuracy. This shows the effectiveness of using localized profiles. The performance of c-CVR is quite similar to that for MCF using only contextual relation. Note that MCF is a supervised method for exploring the information from user-provided groundtruth while our method is unsupervised. Also, MCF learns relationships from the TRECVID 2005 development set and may thus exhibit the domain-shift problem: this might explain why MCF performs worse than c-CVR on TV08$^+$.

From Table III, we can observe that the overall improvement over the baselines yielded by l-CVR is significant (21.5%, 26.1%, 51.7%, and 13.2% for TV06, TV07, TV08, and TV08$^+$, respectively). Because the proposed l-CVR approach simultaneously exploits contextual and temporal information, it outperforms all methods that use only contextual information on all datasets—even the MCF method using only contextual information. This result demonstrates that video re-indexing greatly benefits from the use of temporal relations. From this point of view, our method is useful because it is the only unsupervised method which explores both temporal and contextual relations. When taking into account both contextual and temporal relationships, MCF performs better than l-CVR, but not by much. This is

[3]The overall best performance was obtained with $\eta=10^{-3}$, $\delta=10^{-4}$, and $\alpha=0.6$, which represent learning rate, convergence threshold, and fusion weight, respectively.

[4]http://vireo.cs.cityu.edu.hk/research/dasd/dasd.htm

[5]The annotations for a lexicon of 374 concepts on the TRECVID 2005 development set were obtained from Columbia374 [Naphade et al. 2006; Yanagawa et al. 2007].

Table III. Summary of performance gains over the VIREO-374 baselines on TV06, TV07, TV08 and TV08$^+$ when applying the proposed collaborative video re-indexing approaches (g-CVR, c-CVR, and l-CVR) and comparisons with online ordinal reranking (OOR) [Yang et al. 2009], semantic diffusion (SD) [Jiang et al. 2009], and multi-cue fusion (MCF) [Weng and Chuang 2008] approaches. Note that the MCF is a supervised method which utilizes the contextual and contextual-temporal relations from manually annotated groundtruth while others are unsupervised.

Dataset		TV06	TV07	TV08	TV08$^+$
# of evaluation concepts		20	20	19	19
Baseline (mean infAP)		0.1542	0.0984	0.0391	0.1334
Contextual relation	MCF*	16.7% (0.1800)	18.6% (0.1167)	42.1% (0.0556)	4.4% (0.1393)
	OOR	10.4% (0.1702)	9.8% (0.1080)	23.3% (0.0482)	4.0% (0.1388)
	SD	8.3% (0.1670)	9.6% (0.1078)	35.0% (0.0528)	4.0% (0.1387)
	g-CVR	15.3% (0.1778)	12.8% (0.1110)	33.5% (0.0522)	4.5% (0.1394)
	c-CVR	14.3% (0.1762)	16.0% (0.1141)	41.9% (0.0555)	6.3% (0.1419)
Contextual-temporal relation	MCF*	27.3% (0.1963)	33.7% (0.1316)	57.4% (0.0615)	20.5% (0.1607)
	l-CVR	**21.5% (0.1874)**	**26.1% (0.1241)**	**51.7% (0.0593)**	**13.2% (0.1510)**

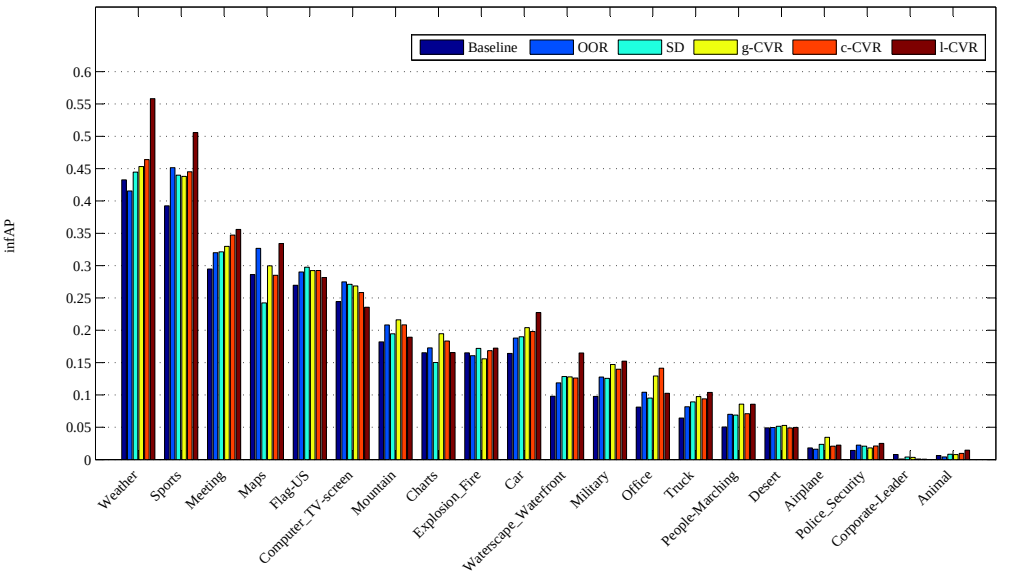
remarkable, considering that MCF uses expensive user annotations while our method is unsupervised. This again confirms the success of our approach in exploring contextual-temporal information as well as its effectiveness for video re-indexing.

We also noticed that the performance on TV08$^+$ does not benefit as greatly as those on the others. The main reason is that only 19 classifiers for the evaluated concepts on TV08$^+$ are retrained on new data. Scores of these 19 concepts are more accurate than the scores of the other concepts which are not evaluated but could support the evaluated concepts. Therefore, CVR approaches have to use the supporting concepts with less accurate scores to boost the performance of relatively more accurate evaluated concepts. This may partially explain why the relative performance gain is not as significant as others. We believe that performance will further improve given more accurate supporting concepts.

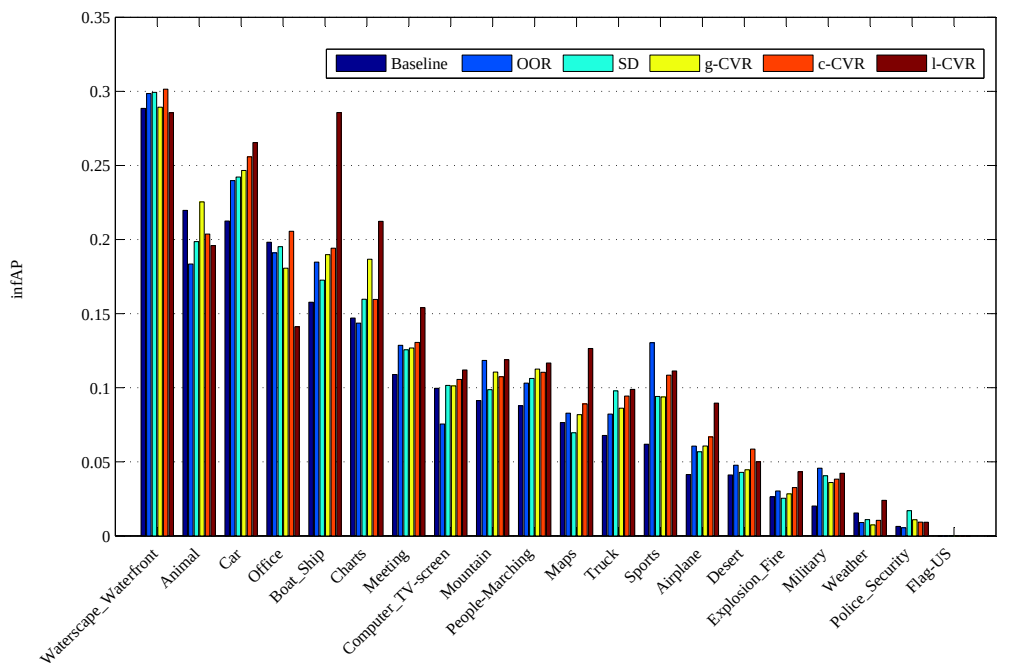
Figure 3 shows the performance of individual concepts on TV06, TV07, TV08 and TV08$^+$, for baseline, OOR, SD, and the g-CVR, c-CVR, and l-CVR approaches in terms of infAP. From Figure 3(a), for TV06, we note that some concepts are greatly improved by applying l-CVR, e.g., *Weather*, *Sports*, and *Car*, while the performance gains of a few concepts are not obvious, e.g., *Mountain*, *Desert*, and *Corporate-Leader*. Similar results are observed on TV07 and TV08; the performance of *Boat_Ship* and *Driver* is much improved but it fails on *Office* and *Hand*. There are a number of possible reasons. First, the concepts in the lexicon are not equally distributed in the semantic space. For instance, we found the semantic distribution around concept *Vehicle* is more dense. Therefore, concepts related to this concept, such as *Car* and *Truck*, may benefit from more cues for refinement and thus yield greater improvements. Second, the degree of noise varies from concept to concept as well as from dataset to dataset. To some extent, error propagation seems inevitable when noise population increases. Thus, in this situation, some concepts may suffer from performance degradation. Fortunately, empirical evidence seems to suggest that this case rarely occurs. As shown in Figure 3, CVR approaches consistently outperform the baselines.

6.3 Computation Time

We report the computational times of our algorithms on a PC with a 2.66 GHz Pentium Quad CPU and 8GB RAM for TV07 which contains 18,142 shots. The proposed g-CVR, c-CVR, and l-CVR methods took 134.1, 299.8, and 834.3 seconds, respectively. For unsupervised methods which explore only contextual relations, the proposed g-CVR and c-CVR methods are slower than OOR (101.5 seconds) and SD (8.8 sec-

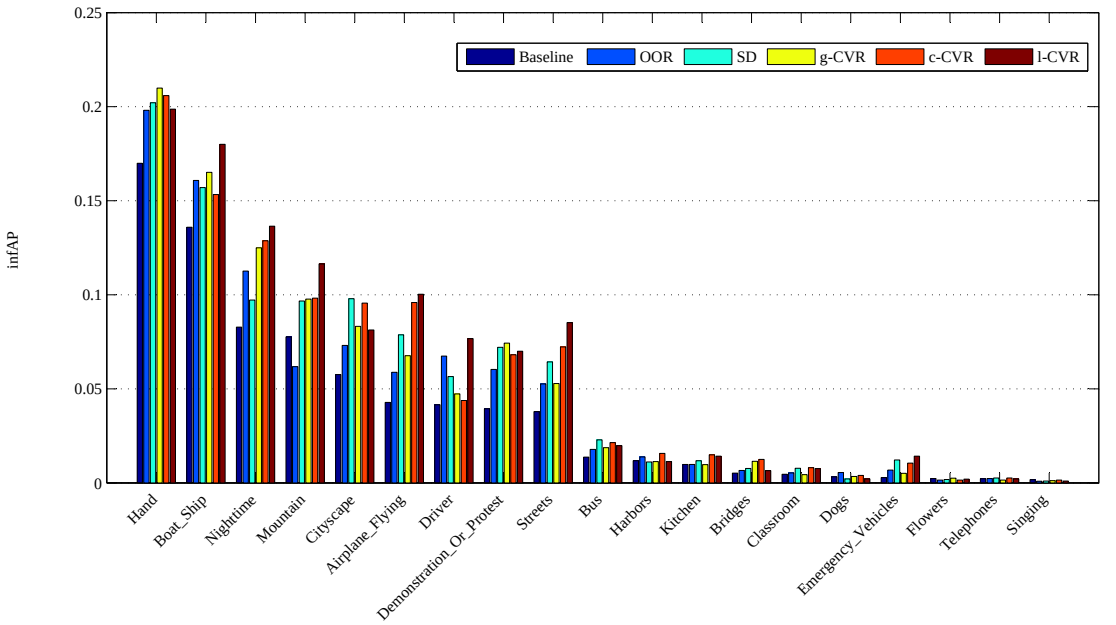


(a) Performance of individual concepts on TV06

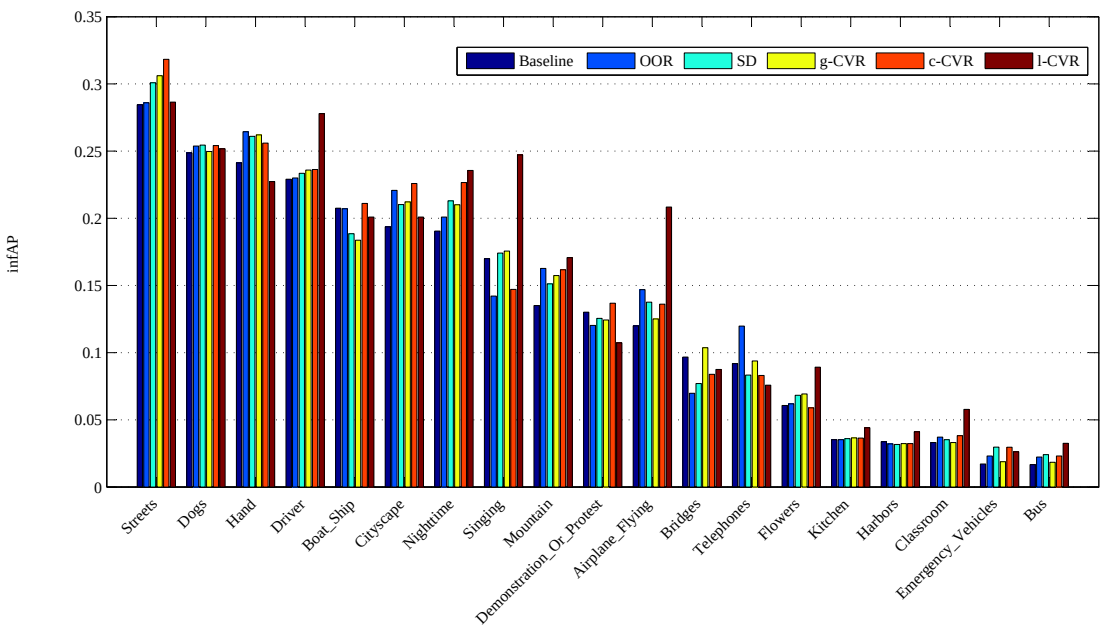


(b) Performance of individual concepts on TV07

Fig. 3. InfAP for the individual concepts in the official evaluation of the TRECVID 2006–2008 benchmarks, using the VIREO-374 baselines, online ordinal reranking (OOR) [Yang et al. 2009], semantic diffusion (SD) [Jiang et al. 2009], and the various versions of the proposed collaborative video re-indexing (g-CVR, c-CVR, and l-CVR) approaches.



(c) Performance of individual concepts on TV08



(d) Performance of individual concepts on TV08$^+$

Fig. 3. InfAP for the individual concepts in the official evaluation of the TRECVID 2006–2008 benchmarks, using the VIREO-374 baselines, online ordinal reranking (OOR) [Yang et al. 2009], semantic diffusion (SD) [Jiang et al. 2009], and the various versions of the proposed collaborative video re-indexing (g-CVR, c-CVR, and l-CVR) approaches (con't).

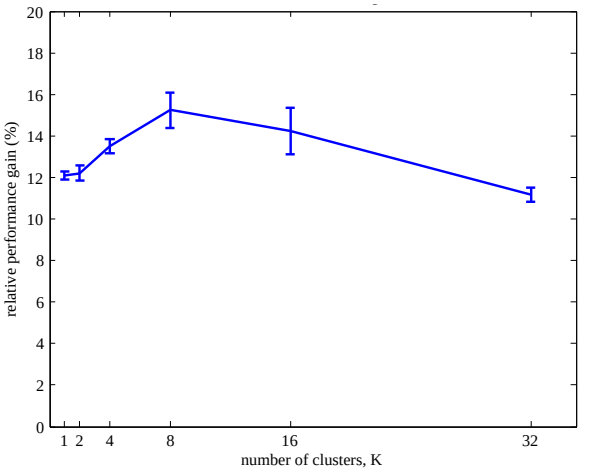


Fig. 4. The means and standard deviations of relative performance gains of the proposed c-CVR method on TV07 under various numbers of clusters. We can notice that the improvements for all experiments are consistently higher than 12% over the baseline and the variances are small. In addition, the results with the profile grouping usually offer more gains than the one without.

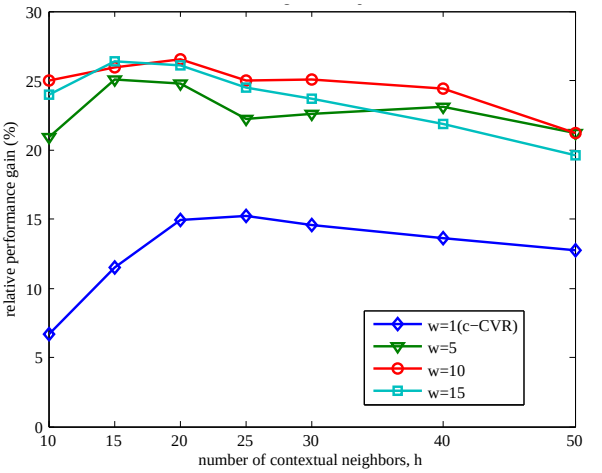


Fig. 5. The relative performance gains of the proposed CVR method on TV07 using various numbers of contextual and temporal neighbors. For exploring the contextual-temporal relation, our l-CVR yields more than 20% improvement over the baseline despite of the choices of w and h. Note that for adapting to various matrix sizes in this experiment, a unified threshold, i.e., $\xi = 80\%$, is used to dynamically determine the value of p.

onds) because matrix factorization is often more expensive. However, they offer better performance; and compared with classifier training and prediction, video re-indexing often only represents a small portion of time. Furthermore, the c-CVR and l-CVR methods can be significantly sped up by utilizing multi-core CPUs or modern GPUs, because matrix factorizations for clusters can be performed independently.

6.4 Discussion on Parameters

Like most unsupervised methods, our method uses several parameters: the number of clusters K, the temporal window size w, the number of relevant concepts h, the number of factors p, and the regularization coefficient λ. One can judge the usefulness of an unsupervised method from three perspectives: (1) parameter stability—is the performance sensitive to the parameter setting; (2) parameter generalization—can one use the same set of parameters for many other different collections; (3) strategy for selecting parameters—are there general principles or rules for choosing proper parameters. In the following, we evaluate these three issues of the proposed methods.

In the first study, we study the effect of the cluster numbers and validate the need of profile grouping by applying the c-CVR method on TV07. Because the results could depend on initial guesses, we repeated the experiment for each setting five times. Figure 4 reports the means and standard deviations of the results in terms of relative performance gains over the baseline for different K. First, we notice that, in all experiments, the gains are more than 12% and the variances are small. This shows that the low-rank approximation approach is effective and robust to the random initial guess and the numbers of clusters. In addition, the results with the profile clustering usually yield greater gains than those without clustering, except when there are too many clusters ($K = 32$). This indicates that while clustering generally helps, it is better to avoid creating too many tiny groups, the small shot count of which makes it difficult to do collaborative filtering. As a rule of thumb, good performance can be obtained if the average cluster size is around 2,000.

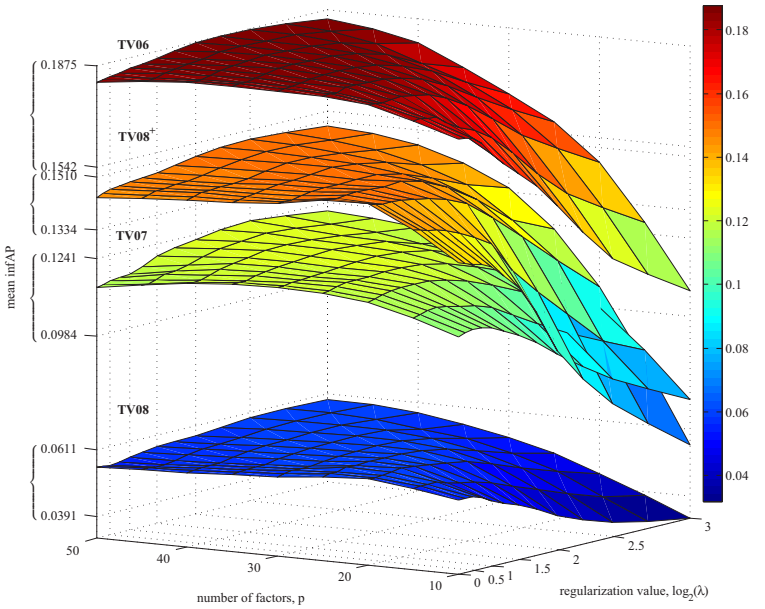


Fig. 6. Performance measures on TRECVID 2006–2008 test sets under various parameter settings, in terms of mean inferred average precision. The tick labels shown in the vertical axis are mean infAP of baselines and the maximum mean infAP achieved within this parameter space. For each dataset, the pair of baseline and improved mean infAP values are connected by "{". As an example, the mean infAP of TV08$^+$ baseline is 0.1334 and the maximal mean infAP for TV08$^+$ in the parameter space is 0.1510.

Next, we evaluate the effects of contextual and temporal neighborhoods, i.e., the parameters w and h. Since the choice of the number of latent factors p relies on the matrix dimensionality, to adapt to a variety of matrix sizes in this study, we dynamically set p as small as possible while keeping the energy content of the input matrix, i.e., the sum of all singular values, above a certain percentage level ξ. We set $\xi = 80\%$ in this experiment. Figure 5 shows the relative performance gains of our l-CVR method for various values of w and h. Note that l-CVR turns into c-CVR when $w = 1$. In this case, the performance gain increases along with the number of relevant concepts h since more contextual information is exploited. It saturates and degrades slightly when h becomes too large and fewer correlated concepts are included. Nevertheless, the gain is still more than 13% when a large h is selected. The number of relevant concepts can be determined by using PMCC as described in Section 5. When w increases, more temporal information can be taken into account, and performance generally improves. Similar to contextual neighbors, performance can degrade slightly when more remote neighbors are included. Nevertheless, l-CVR yields greater than 20% gains despite the choice of w and h. This shows that the algorithm is stable to different parameter settings. We used $w = 10$ and $h = 10$ for l-CVR in later experiments because smaller matrices are more efficient.

There are two more important parameters in our CVR method: the number of factors p and the regularization coefficient λ. Figure 6 illustrates the impact of these parameters on the proposed approach by displaying the overall performance on TV06, TV07, TV08, and TV08$^+$ when using different parameter settings in the parameter space defined by $10 \le p \le 50$ and $0 \le \log_2(\lambda) \le 3$. As one can see, most regions of this space provide similar and substantial performance gains, except for the region located at the corner with small p and large λ. Both tend to lead to much-lower-rank approximation. The bad performance of the very low-rank matrix approximation is due to serious intrinsic structure loss. For robustness, one can determine p by selecting the smallest number such that the energy content of refined data is still more than that of source data, e.g., above 80%. It is also worth noting that all of the datasets used in our experiments share similar relationships between parameter settings and performance gains. Thus we can use a single set of parameters for different datasets with different

characteristics. These findings attest to the generalization ability and performance stability of our collaborative video re-indexing method with respect to parameters.

7. CONCLUSIONS

In this paper, we have introduced an unsupervised method for video re-indexing based on collaborative filtering. The proposed method offers the following advantages: (1) It is an unsupervised approach that does not use expensive user annotations or potentially inaccurate external sources. (2) It takes into account both contextual and temporal cues in a unified way, and thus yields better performance than other unsupervised approaches. (3) The method is independent of the classifier type and can be applied to any classification results without re-training models. The decomposition of relation modeling and detector training makes the proposed method more scalable and easier to use.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions which are very valuable for improving this paper. We especially thank the Video Retrieval Group of City University of Hong Kong and Dr. Yu-Gang Jiang for providing us the useful detection scores. In addition, Aaron Heidel is appreciated for his editorial assistance.

REFERENCES

ADAMS, W. H., IYENGAR, G., LIN, C.-Y., NAPHADE, M. R., NETI, C., NOCK, H. J., AND SMITH, J. R. 2003. Semantic indexing of multimedia content using visual, audio, and text cues. *Eurasip Journal on Applied Signal Processing 2003,* 2, 170–185.

AMIR, A. ET AL. 2005. IBM research TRECVID-2005 video retrieval system. In *Online Proceedings of TRECVID Workshop*.

AYTAR, Y., SHAH, M., AND LUO, J. 2008. Utilizing semantic word similarity measures for video retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–8.

CANDÈS, E. J., LI, X., MA, Y., AND WRIGHT, J. 2009. Robust principal component analysis? Tech. rep., Stanford University.

CHUA, T.-S., TANG, J., HONG, R., LI, H., LUO, Z., AND ZHENG, Y.-T. 2009. NUS-WIDE: A real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*.

JI, H., LIU, C., SHEN, Z., AND XU, Y. 2010. Robust video denoising using low rank matrix completion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.

JIANG, W., CHANG, S.-F., AND LOUI, A. C. 2007. Context-based concept fusion with boosted conditional random fields. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. 949–952.

JIANG, Y.-G., NGO, C.-W., AND YANG, J. 2007. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*.

JIANG, Y.-G., WANG, J., CHANG, S.-F., AND NGO, C.-W. 2009. Domain adaptive semantic diffusion for large scale context-based video annotation. In *Proceedings of the IEEE International Conference On Computer Vision (ICCV)*.

JIANG, Y.-G., YANAGAWA, A., CHANG, S.-F., AND NGO, C.-W. 2008. CU-VIREO374: Fusing Columbia374 and VIREO374 for large scale semantic concept detection. Tech. rep., Columbia University.

JIANG, Y.-G., YANG, J., NGO, C.-W., AND HAUPTMANN, A. G. 2010. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Trans. Multimedia 12,* 1, 42–53.

KENNEDY, L. AND HAUPTMANN, A. 2006. LSCOM lexicon definitions and annotations version 1.0, DTO challenge workshop on large scale concept ontology for multimedia. Tech. rep., Columbia University.

KENNEDY, L. S. AND CHANG, S.-F. 2007. A reranking approach for context-based concept fusion in video indexing and retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*. 333–340.

KOREN, Y., BELL, R., AND VOLINSKY, C. 2009. Matrix factorization techniques for recommender systems. *Computer 42,* 8, 30–37.

LEW, M. S., SEBE, N., DJERABA, C., AND JAIN, R. 2006. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl. 2,* 1, 1–19.

LIU, K.-H., WENG, M.-F., TSENG, C.-Y., CHUANG, Y.-Y., AND CHEN, M.-S. 2008. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE Trans. Multimedia 10,* 2, 240–251.

LIU, Y., MEI, T., AND HUA, X.-S. 2009. CrowdReranking: Exploring multiple search engines for visual search reranking. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieva*. 500–507.

NAPHADE, M. R. AND HUANG, T. S. 2001. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Trans. Multimedia 3,* 1, 141–151.

NAPHADE, M. R., KOZINTSEV, I. V., AND HUANG, T. S. 2002. Factor graph framework for semantic video indexing. *IEEE Trans. Circuits Syst. Video Technol. 12,* 1, 40–52.

NAPHADE, M. R., SMITH, J. R., TEŠIĆ, J., CHANG, S.-F., HSU, W., KENNEDY, L., HAUPTMANN, A., AND CURTIS, J. 2006. Large-scale concept ontology for multimedia. *IEEE Multimedia 13,* 3, 86–91.

QI, G.-J., HUA, X.-S., RUI, Y., TANG, J., MEI, T., WANG, M., AND ZHANG, H.-J. 2008. Correlative multilabel video annotation with temporal kernels. *ACM Trans. Multimedia Comput. Commun. Appl. 5,* 1, 1–27.

QI, G.-J., HUA, X.-S., RUI, Y., TANG, J., MEI, T., AND ZHANG, H.-J. 2007. Correlative multi-label video annotation. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 17–26.

RENNIE, J. D. M. AND SREBRO, N. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*. 713–719.

ROWEIS, S. T. AND SAUL, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290,* 5500, 2323–2326.

SMEATON, A. F., OVER, P., AND KRAAIJ, W. 2006. Evaluation campaigns and TRECVid. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval (MIR)*. 321–330.

SNOEK, C. G. M. ET AL. 2009. The MediaMill TRECVID 2009 semantic video search engine. In *Online Proceedings of TRECVID Workshop*.

SNOEK, C. G. M. AND WORRING, M. 2009. Concept-based video retrieval. *Foundations and Trends in Information Retrieval 2,* 4, 215–322.

SNOEK, C. G. M., WORRING, M., VAN GEMERT, J. C., GEUSEBROEK, J.-M., AND SMEULDERS, A. W. M. 2006. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 421–430.

SU, X. AND KHOSHGOFTAAR, T. M. 2009. A survey of collaborative filtering techniques. *Adv. in Artif. Intell. 2009*, 1–19.

WENG, M.-F. AND CHUANG, Y.-Y. 2008. Multi-cue fusion for semantic video indexing. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 71–80.

YANAGAWA, A., CHANG, S.-F., KENNEDY, L., AND HSU, W. 2007. Columbia University's baseline detectors for 374 LSCOM semantic visual concepts. Tech. rep., Columbia University.

YANG, Y.-H., HSU, W. H., AND CHEN, H. H. 2009. Online reranking via ordinal informative concepts for context fusion in concept detection and video search. *IEEE Trans. Circuits Syst. Video Technol. 19,* 12, 1880–1890.

YILMAZ, E. AND ASLAM, J. A. 2006. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. 102–111.

Received September 2010; revised December 2010; accepted February 2011