Near-Duplicate Keyframe Retrieval with Visual Keywords and Semantic Context

Xiao Wu
Department of Computer Science
City University of Hong Kong
83 Tat Chee Avenue, Hong Kong
wuxiao@cs.cityu.edu.hk

Wan-Lei Zhao
Department of Computer Science
City University of Hong Kong
83 Tat Chee Avenue, Hong Kong
wzhao2@cs.cityu.edu.hk

Chong-Wah Ngo
Department of Computer Science
City University of Hong Kong
83 Tat Chee Avenue, Hong Kong
cwngo@cs.cityu.edu.hk

ABSTRACT
Near-duplicate keyframes (NDK) play a unique role in large-scale video search, news topic detection and tracking. In this paper, we propose a novel NDK retrieval approach by exploring both visual and textual cues from the visual vocabulary and semantic context respectively. The vocabulary, which provides entries for visual keywords, is formed by the clustering of local keypoints. The semantic context is inferred from the speech transcript surrounding a keyframe. We experiment the usefulness of visual keywords and semantic context, separately and jointly, using cosine similarity and language models. By linearly fusing both modalities, performance improvement is reported compared with the techniques with keypoint matching. While matching suffers from expensive computation due to the need of online nearest neighbor search, our approach is effective and efficient enough for online video search.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering, Search process*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *Video analysis.*

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Similarity Measure, Image Retrieval, Language Model, Near-Duplicate Keyframe, Multiple Modalities, News Videos.

1. INTRODUCTION
Near-duplicate keyframes (NDK) are a set of similar keyframes but with certain variations induced by acquisition times, lighting conditions, and editing operations, which abundantly exist in real applications. Retrieval of near-duplicate keyframes [9, 13, 20] plays an important role in measuring video clip similarity, tracking video shots from multi-lingual sources, and threading news stories under the same topic [17].

Due to the large and diverse variations in near-duplicate keyframes, retrieving NDK remains difficult particularly for approaches based mainly on global features or signatures. Recently, keypoint matching sights promising performance for both NDK retrieval and detection [9, 13, 20]. Keypoints are local salient regions detected over images scales. Their descriptors (e.g., SIFT [10]) are mostly invariant to local transformations. Due to these properties, keypoints, in contrast to global features, can tolerate various geometric and photometric transformations. While keypoint matching has demonstrated to be effective for NDK identification, the matching process is naturally slow due to the large amount of keypoints and the high dimensionality of keypoint descriptors. Typically there are hundreds to thousands of keypoint available in one keyframe. Even with the multi-dimensional indexing structure [9], the matching (nearest neighbor search) is expected to be computationally intractable and not scalable to large video database.

Capitalizing on the merit of keypoints, we propose a visual keyword-based retrieval approach to eliminate the need of online keypoint matching as in [13]. The visual words are constructed by the offline quantization of keypoint descriptors. Under our representation, each keyframe is treated as a bag of visual words (BoW), analogous to the documents composed of text words. With BoW, we adopt the cosine similarity and language models respectively to evaluate the likelihood of near-duplicate between keyframes.

In addition to visual keywords, we also exploit textual keywords for NDK retrieval. Keyframes (images) are usually associated with a certain semantic context. For example, images embedded in web pages are commonly accompanied with their corresponding text contents, while keyframes extracted from news videos are correlated with the story contents in the audio track. In this paper, the *semantic context* of a keyframe refers to the story text transcripts extracted through speech recognition in the audio track. The semantic context basically provides a meaningful cue for NDK retrieval. If two keyframes are NDK, their semantic context is related to some extent. Figure 1 gives an example. The keyframes of the suspects in the "Arkansas school shooting" may be contained in other stories of this event or the related legal issues of shooting and adolescent education. But they have rare chance to be appeared in sports, commercials, or other events such as the "Indonesia chaos". That is, their semantic context is different. Therefore, to compare whether two keyframes are near-duplicates, in addition to the visual similarity on visual keywords, we also consider the semantic context similarity on text transcripts.

CIVR'07, July 9–11, 2007, Amsterdam, The Netherlands.

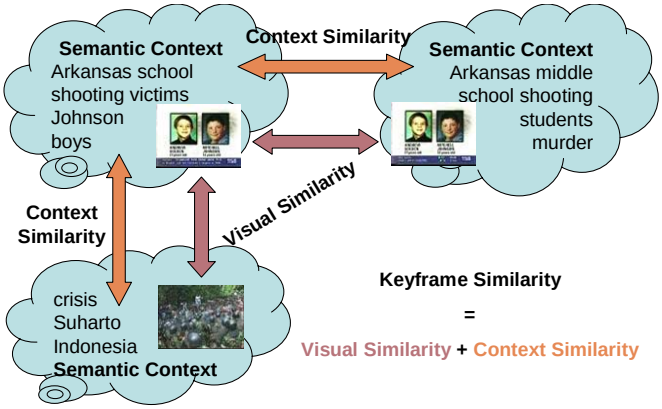

Figure 1. Near-duplicate keyframe retrieval with the visual and semantic similarity

To this end, we have two kinds of keywords associated with keyframes: one is visual keywords from the visual contents; the other is text keywords from the semantic context. They provide different viewpoints to describe the keyframes, which enlightens us the possibility of using vector space models and language models for both visual and text keywords to efficiently retrieve NDK. Furthermore, the fusion of visual information and semantic context is further explored to boost the performance. We demonstrate that the proposed approach not only greatly speeds up the retrieval efficiency, but is also competitive to and even better than the techniques with keypoint matching.

The rest of this paper is organized as follows. Section 2 describes the related work to NDK retrieval and detection. Sections 3 and 4 present the generation of visual keywords and the keyframe similarity measures respectively. Section 5 proposes the fusion of visual information and context semantic. Finally Section 6 presents our experimental results, and Section 7 concludes the major findings in this paper.

2. RELATED WORK

Near-duplicate keyframes are keyframes close to the exact duplicate of each other, but different in the capturing conditions, acquisition times, rendering conditions or editing operations [20]. There are a huge number of NDK existed in news stories and they provide critical cues for tasks like novelty/redundancy detection and topic threading. Recently, NDK were also exploited in [3] to boost the performance of interactive video search. Hsu et al. [6] tracked topics with visual duplicates and semantic concepts, and found that near-duplicates significantly improve the tracking performance. Zhai et al. [19] linked news stories by combining keyframe matching and textual correlation. In our previous work [17], we combined the text and NDK to thread the news topics.

The methods to detect the NDK were proposed in [5, 9, 13, 20, 23]. Most approaches utilized local keypoint features for matching and demonstrated the surprisingly good performance. Differing from global features, keypoints locate the local regions which are tolerant to geometric and photometric variations. The features (e.g. SIFT [10]) are further extracted to describe the spatial structure and local orientation from the region surrounding keypoints. One of the earlier studies [15] has indeed shown that keypoints are promising for object matching. In our recent work [13], the one-to-one symmetric (OOS) keypoint matching was proposed to evaluate the degree of near-duplicate between keyframes. To speed up OOS which is essentially a problem of nearest neighbor search, locality sensitive hashing (LSH) [9] and LIP-IS [23] were experimented. Nevertheless, even with the filtering mechanisms, the retrieval speed cannot be substantially improved due to various factors such as feature dimensionality and the amount of keypoints to be matched per keyframe. In [24], in order to enable large-scale video search with NDK, videos were first temporally partitioned into small groups and NDK pairs were then detected with keypoint matching within each group. NDK among groups were threaded with transitivity propagation [13] to allow efficient retrieval. However, some NDK pairs were missed due to the partition, while some were falsely propagated along the time dimension.

The idea of visual dictionary was initially proposed in [15], partly to convert the matching of keypoints analogous to the direct comparison of words. Under this framework, keypoints are offline clustered into visual keywords, and each keyframe is indexed with a vector of visual keywords. The comparison between keyframes could be performed by the classic *tf-idf* document vector space model. In addition to vector space models, language models were also introduced to information retrieval [14, 18], and has performed well empirically [1, 21]. Comparison of two keyframes can be converted to compare two distributions or vectors on visual keywords. However, visual keywords are different from traditional text words, and it is uncertain whether language models of visual keywords are similar to text language models and effective for NDK retrieval. To the best of our knowledge, little works have discussed the language models on visual keywords, and it is interesting and meaningful to explore it.

Generally, news videos provide richer information than images and text streams. Keyframes are usually within a certain semantic context, and NDK commonly appear in stories with the related context. The semantic context provides meaningful information for NDK retrieval. Contextual information has known to be useful for multimedia modeling and has been actively discussed from different viewpoints, ranging from the spatial, temporal and spatial contextual information [4], shape context [2], to pattern context [8]. Multi-modality fusion has also been intensively researched and applied to various tasks particularly the image and video retrieval. However, until now the fusion of visual keywords and semantic context for NDK retrieval, to the best of our knowledge, has not yet been explored.

3. VISUAL KEYWORDS GENERATION

Visual keywords (VK) are generated through the clustering of local keypoint features. Basically VK is a dictionary where each word represents a cluster of keypoints. The mapping between keyframe and dictionary is done by assigning each keypoint to a word which corresponds to its nearest cluster. Figure 2 illustrates the process of generating VK and depicting keyframes with the learned VK. Initially keypoints and their feature descriptors are extracted. The descriptors are clustered to learn a dictionary which in turn annotates each keyframe as a bag-of-words (BoW). A keyframe can be represented by a feature vector containing the frequency of each visual keyword. Statistically, keyframes can also be represented as smoothed probability distributions over the visual keywords. Therefore, the degree of near-duplicate between

keyframes can be evaluated with vector space models and language models.

Currently, there are a couple of keypoint detectors and descriptors available [11, 12]. The detectors basically locate stable keypoints (and their support regions) which are invariant to kinds of transformations introduced by geometric and photometric changes. Popular detectors include Harris-Affine [12], Hessian-Affine [12], and Difference of Gaussian (DoG) [10]. The descriptors of keypoints are invariant to certain transformations that exist in different images. SIFT (Scale-Invariant Feature Transform) has shown to be one of the best descriptors for keypoints [11], which is a 128-dimensional feature vector that captures the spatial structure and the local orientation distribution of a patch surrounding keypoints. PCA-SIFT, proposed in [9], is a compact version of SIFT with principal component analysis. In consideration of both efficiency and effectiveness, we adopted Hessian-Affine [12] as the keypoint detector, and SIFT [10] as the descriptor.

4. VISUAL SIMILARITY

With BoW representation, we exploit vector space model (through cosine similarity) and language models to measure the similarity of keyframes.

4.1 Cosine Similarity

The *cosine similarity* metric is a popular vector space model in information retrieval. The cosine of the angle between the visual keyword vectors of two keyframes determines the similarity score. It is a pairwise measure, defined by

$$S_V(K_i \mid K_j) = \frac{\sum_{k=1}^{m} v_k(K_i) v_k(K_j)}{\sqrt{\sum_{k=1}^{m} v_k(K_i)^2 \sum_{k=1}^{m} v_k(K_j)^2}} \quad (1)$$

where $v_k(K_i)$ is the weight for visual keyword v_k in keyframe K_i. The weighting function used in our experiments is specified by the following formula [1]:

$$v_k(K_i) = \frac{tf(v_k, K_i)}{tf(v_k, K_i) + 0.5 + (1.5 * \frac{len(K_i)}{adl})} \cdot \frac{\log \frac{n+0.5}{dv_k}}{\log(n+1.0)} \quad (2)$$

It is a *tf-idf* function in which $tf(v_k\ K_i)$ is the term frequency of visual keyword v_k in keyframe K_i , adl is the average number of visual keywords in a keyframe, dv_k is the document frequency of the visual keyword, $len(K_i)$ is the number of visual keywords in the keyframe, and n is the number of keyframes in the corpus.

4.2 Language Models on Visual Keywords

In addition to the vector space model, we also explore the language models on visual keywords to compare the similarity between keyframes. A *language model* on BoW is a probability distribution that captures the statistical regularities of visual keywords. We assume that a keyframe K_i is generated by a unigram visual keywords distribution θ. In the language model, a multinomial model $p(v_k|\theta_i)$ over visual keyword v_k is estimated for each keyframe K_i in the visual collection C. Given two language models of keyframes built on visual keywords, language modeling refers to the problem of estimating the likelihood that two keyframes could have been generated by the same language

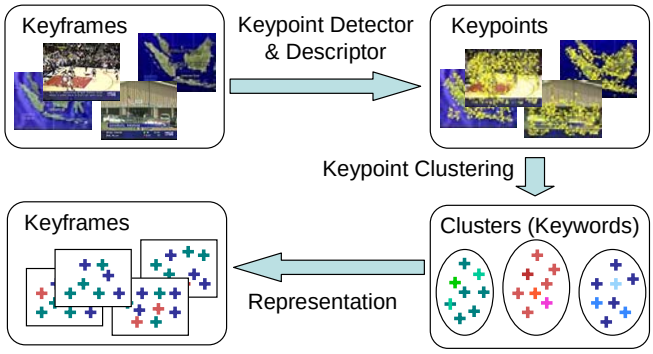

Figure 2. Keyframe representation

model. The similarity between two keyframes is measured by the *Kullback-Leibler (KL) divergence* between two language models.

4.2.1 Symmetric KL Divergence

A distribution similarity measure, *KL* divergence (or relative entropy), is commonly used to measure the similarity between two distributions. However, *KL* divergence is asymmetric measure, i.e. *KL(A,B)* is not equal to *KL(B,A)*. For two NDK, their similarity should be symmetric [13]. So we use the symmetric *KL* divergence to measure the similarity between two keyframes. This property makes the measure stable. The similarity measure (i.e. symmetric *KL*) is defined as:

$$S_V(K_i \mid K_j) = (-KL(\theta_i, \theta_j) - KL(\theta_j, \theta_i))/2 \quad (3)$$

$$= (-\sum_{v_k} p(v_k \mid \theta_i) \log(\frac{p(v_k \mid \theta_i)}{p(v_k \mid \theta_j)}) - \sum_{v_k} p(v_k \mid \theta_j) \log(\frac{p(v_k \mid \theta_j)}{p(v_k \mid \theta_i)}))/2$$

where θ_i is the language model for keyframe K_i, which is a multinomial distribution. Here $p(v_k|\theta_i)$ is the probability of visual keyword v_k occurring in keyframe K_i, similarly for $p(v_k|\theta_j)$. The higher the similarity is, the more near-duplicate two keyframes are.

The simplest way to estimate $p(v_k|\theta_i)$ is the *Maximum Likelihood Estimation (MLE)*, simply given by relative counts:

$$p(v_k \mid \theta_i) = \frac{tf(v_k, K_i)}{\sum_{v_k} tf(v_k, K_i)} \quad (4)$$

where $tf(v_k, K_i)$ is the term frequency of visual keyword v_k in keyframe K_i. However, the problem of MLE is that it will generate a zero probability if a visual keyword never occurs in the keyframe K_i, which will cause $KL(\theta_i, \theta_j)=\infty$.

Smoothing techniques are used to assign a non-zero probability of the unseen keywords and improve the accuracy of feature probability estimation. Prior research on text information retrieval [18, 21] shows that different smoothing techniques highly affect the performance. For language models, we mainly use *Bayesian* smoothing with *Dirichlet* priors and *Shrinkage*. Furthermore, a *Mixture Model* is also experimented.

4.2.2 Dirichlet Smoothing

This smoothing technique uses the conjugate prior for multinomial distribution, which is the *Dirichlet* distribution. It automatically adjusts the amount of reliance on the visual keywords according to the total number of visual keywords. For a *Dirichlet* distribution with parameters:

19980509_ABC_12	19980511_ABC_14	19980509_CNN_10
		
United States allies delivered strong warning date **president Suharto Indonesia** reform **government** fix **economy** end growing unrest **country Suharto** warning student writers **protests** face crackdown troops agencies marked key **Indonesia nation** students growing boulder **demonstration day** police confined campus dole streets coordinate brazenly **demanding** overthrow **support** recede execution **president** part years question corruption folly recently **public support** prison **economic crisis** caused **pain** anger **support** student **demands** readying ordinary **Indonesians** longer **afraid** speak regressive blessing disguise borrow **nation** caught Evans broken cottages easing **government international** easy **demonstrations army** private wrong **military** openly showed **sympathy** students soldiers suffering urgent **crisis** stop **president Suharto** issuing tough warning **army** crush **violent protests** threaten **nation** stability students intimidating end peaceful **protests national** symbol hundred young **people** hauled combat troops shouted journalists world kind joy **news** court	**Indonesian president Suharto Indonesians painful** sacrifices survive **economic crisis** rapidly plunging **Indonesia** closer chaos **nation** dictators conference Egypt week home world fourth populous **country** boil high priority **United States** baby sees market fears capital Jakarta risky **days** opponents **president Suharto** anti **government protests** brought university exploded **violence** soaring prices middle class longer **afraid** criticize regime **armed** forces part camp Dave chilled **sympathy public** desire democracy months stand powerful local organizations **demanded support** resignation exclusive interview Rice **United States** m. f. pressure **Suharto** withholding **international** aid create suffering America moral **support** agents tough clear messages **Suharto** doubts **day** hold differs human rights tests kill **demonstrations** widespread popular **president Suharto** risk **military** back fairly included step submitted **people** problem revolution network revolution landed person jail worse word revolution inspire folk yeah t. **news** Jakarta	Italy authorities hundred **people** killed dozens missing torrential rains unleashed rivers mud debris yesterday emergency crews rescued man buried neck mind **days** doctors condition

Figure 3. Three keyframes with a green border have high visual similarity on visual keywords. The semantic context to which the first two keyframes belong is similar, while the third is not.

$$(\mu p(v_1 \mid C), \mu p(v_2 \mid C), ..., \mu p(v_n \mid C))$$

the posterior distribution using *Bayesian* analysis is:

$$p_\mu(v_k \mid \theta_i) = \frac{tf(v_k, K_i) + \mu p(v_k \mid C)}{\sum_{v_k} tf(v_k, K_i) + \mu} \quad (5)$$

$p(v_k|C)$ is the collection language model and μ is a parameter to adjust the degree of smoothing. In our experiments, the collection model is built on all keyframes in the corpus.

4.2.3 Shrinkage Smoothing

Shrinkage smoothing is a special case of the *Jelinek-Mercer* smoothing method, which involves a linear interpolation of the maximum likelihood model with *n*-gram model [18]. Based on the assumption that a keyframe is generated by sampling from two different language models: a keyframe model and a collection model, the language model of a keyframe is determined by:

$$p(v_k \mid \theta_K) = (1-\lambda) p(v_k \mid \theta_{ML_K}) + \lambda p(v_k \mid \theta_{ML_C}) \quad (6)$$

using coefficients λ to control the influence of each model. θ_{ML_K} and θ_{ML_C} are the maximum likelihood language model of the keyframe and collection respectively.

4.2.4 Mixture Model

A mixture model [21] is based on the assumption that keywords occurred more frequently in a keyframe than in the collection should have a higher probability in the keyframe model. Therefore, the approach is to deduce the maximum likelihood keyframe model. Each visual keyword in the keyframe is generated by the keyframe and collection language models with probability *(1-λ)*, and λ respectively.

$$p(v_k \mid \theta_{ML_K}) = (1-\lambda) p(v_k \mid \theta_K) + \lambda p(v_k \mid \theta_{ML_C}) \quad (7)$$

To note, although equations of shrinkage smoothing and mixture model look similar, the model acquired and used to calculate *KL* divergence is different. Shrinkage smoothing increases the probability of keywords that occur frequently in the collection if they occur less frequently in keyframe, while mixture model decreases the probability of these features [21]. Similar to [1], the language model θ_K that maximizes the likelihood of the observed keyframe, given fixed parameters, was computed using the technique described in [22].

5. SEMANTIC CONTEXT

In the previous section, we discuss the cosine similarity and language models built on visual keywords to measure the visual similarity. However, visual dictionary based approaches are actually an approximation of many-to-many matching. Compared to the one-to-one keypoint matching between two keyframes [13], the assignment of keywords to keypoints lacks distinctiveness. Practically the performance of keyword assignment cannot exceed that of keypoint matching. In this section, we further explore another useful source: semantic context, and then propose the fusion of visual and contextual similarity for NDK retrieval.

Keyframes are usually accompanied with their semantic context. Near-duplicate keyframes frequently appear in stories having related context, while have rare chance to emerge in unrelated context. The pure visual methods may overlook the interactions between visual and contextual information. Although the similar contextual information does not have a say about the identity of near-duplicates, the contextual contents potentially determine the appearance of NDK. Visual and contextual information complement each other, so a robust and reasonable approach should combine both visual and contextual contents to determine the degree of similarity among keyframes while exploiting the significance of these contents.

Figure 3 shows three keyframes with a green border and their corresponding semantic context. The semantic context refers to the story text transcripts extracted through speech recognition in the audio track. The three keyframes have high visual similarity due to the sharing of similar visual keyword distribution. The

assignment of visual words to keypoints is indeed affected by the performance of clustering, which is always far from perfect practically. As a consequence, these keyframes are claimed similar in our experiment even though the third keyframe is semantically different. This ambiguity can be alleviated by investigating the semantic context. For instance, the first two stories have similar semantic context and the common words appeared in both stories are bolded in blue. We can see that words such as "Indonesia", "Suharto", and "protests" are frequently appeared in both stories. A judgment can be obtained that these two stories are on the same theme that a chaos happened in Indonesia. In addition to the visual similarity, the context similarity reinforces the confidence that the two keyframes are near-duplicates. On the contrary, although the third keyframe also has high visual similarity, the semantic context is totally different from previous ones. Its context is mainly about a torrent in Italy, and the crucial words such as "Indonesia", "crisis" never present in it. So it has less probability to be near-duplicates with the other two keyframes. In conclusion, if two keyframes have high visual similarity, and meanwhile their semantic similarity is also high, then they can be declared as near-duplicates. Otherwise, if their visual similarity is high but the context similarity is low, or vice verse, they have less chance to be near-duplicates.

Motivated by these observations, we propose the fusion of visual and contextual similarity to evaluate the degree of near-duplicate. We use linear fusion to combine the similarity scores from visual and contextual contents, denoted as V and T respectively. Linear fusion model has been shown to be one of the most effective approaches to fuse textual and visual modalities in video retrieval. Given two keyframes K_i and K_j, the similarity measure is defined as:

$$S(K_i, K_j) = S_V(K_i, K_j) + \alpha S_T(T_i, T_j) \qquad (8)$$

where $S_V(K_i, K_j)$ is the visual similarity between K_i and K_j, while $S_T(K_i, K_j)$ is the contextual similarity. The visual and contextual similarity measures can be either cosine similarity or any language model mentioned in Section 4. Their similarity scores are normalized in the range of [0, 1] for fusion. The factor α is used to control the influence of context similarity. The linear factor is determined empirically, but the value should be relatively small to avoid the context similarity dominating the final score. High contextual similarity alone is not a strong indication for NDK, but a reliable complement for visual similarity.

6. EXPERIMENTS

6.1 Dataset and Performance Metric

We use the data set given by [20] for evaluation, which is a subset of TRECVID 2004 video corpus [16]. The data set consists of 600 keyframes with 150 NDK pairs (i.e. 300 NDK). We randomly select part of keyframes (150 keyframes) from the data set to build a visual vocabulary with 3500 individual visual keywords based on [15]. For visual vocabulary generation, keypoints were extracted with Hessian Affine [12] and described by SIFT [10]. Traditional *k*-means algorithm was employed to group keypoints (77,706) into 3500 clusters, in which each cluster represents a visual keyword.

We use the story and shot boundaries specified by TRECVID for experiments. The textual features are a list of words extracted from speech transcripts by an automatic speech recognition system (ASR) at LIMSI [7]. Totally, there are 398 stories, in which 6 stories have no corresponding text transcripts. After data preprocessing (such as word stemming and stop-word removal), there are 5,551 unique words. The context similarity is computed based on the stories that the keyframes belong to.

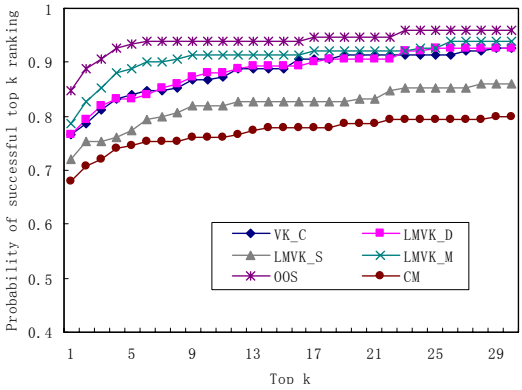

Figure 4. Performance comparison

We use all NDK pairs (150) as queries for NDK retrieval in the experiments. The retrieval performance is evaluated with the probability of the successful top-*k* retrieval, defined as:

$$R(k) = \frac{Q_c}{Q_a}$$

where Q_c is the number of queries that find its duplicate in the top k list, and Q_a is the total number of queries. The ranking is based on the similarity score.

To get the general performance evaluation, we also calculate the average probability of the top-10 retrieval.

$$AP = (\sum_{k=1}^{10} R(k))/10$$

6.2 Performance on Visual Keywords

We evaluate the performance of language models (LMVK) and cosine similarity (VK_C) on visual keywords. The performance is compared with the OOS keypoint matching technique [13] and block-based color moment (CM). We treat OOS as the upper limit performance of keypoint based methods, and CM as the baseline when global features are used. OOS performs one-to-one symmetric matching among keypoints, and rank keyframes according to the cardinality of keypoints being matched. For CM, each keyframe is depicted with the first three color moments (i.e. mean, standard deviation, and skewness) extracted in *Lab* color space over 5×5 grid partitions. For language models, we test Dirichlet smoothing (LMVK_D), Shrinkage smoothing (LMVK_S) and Mixture Model (LMVK_M).

The comparison is summarized in Figure 4. In the experiment, OOS achieves the best performance because the one-to-one symmetric matching scheme precisely locates the nearest neighbor of keypoints and eliminates the false matches, which guarantees stable and unique matches among keypoints. CM, on the other hand, performs poorly since the global information is not enough to capture the variations such as lighting, viewpoint and editing changes. Visual keyword (VK) consistently outperforms CM, while approaching the performance of OOS

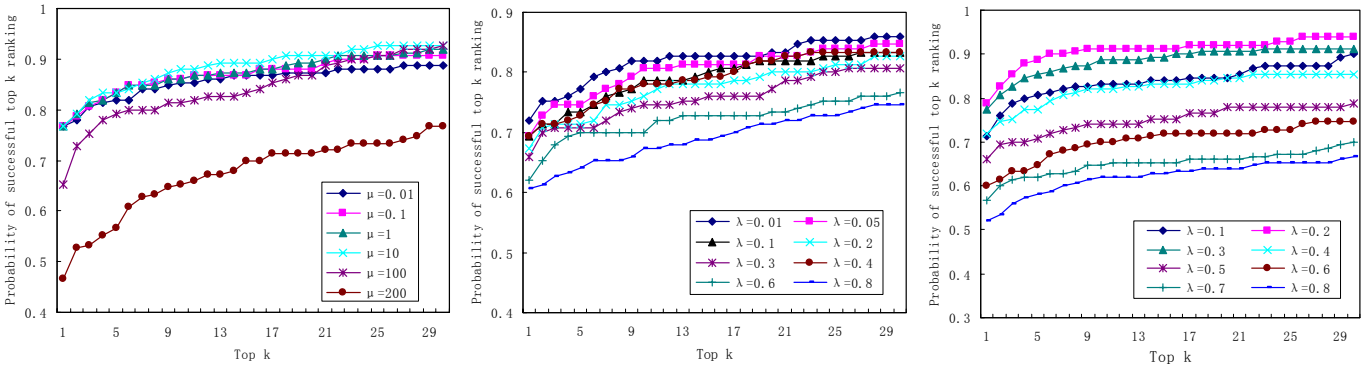

Figure 5. Performance of language models (LMVK) with various smoothing techniques (a) Dirichlet smoothing (LMVK_D) (b) Shrinkage smoothing (LMVK_S) (c) Mixture model (LMVK_M)

depending on the similarity measure being used. Compared with OOS, VK is less competing. This is not surprise since VK is the outcome of keypoint clustering. Owing to the fact that a visual word represents a group of similar keypoints, VK based approaches are indeed the approximate version of offline many-to-many (M2M) matching. As studied in [23], M2M is less tolerant to noise compared to one-to-one matching such as OOS in NDK identification. Among the various measures on VK, LMVK demonstrates promising performance, but depends on smoothing techniques. Cosine similarity (VK_C) is a robust measure. Although other factors may affect the performance of VK (e.g. vocabulary size, "polysemy" and "synonymy"), VK based methods have shown the potential to effectively measure the similarity of keyframes.

Language models have been shown to be sensitive to smoothing methods and their parameter settings in text retrieval [18]. Through empirical studies, we have similar conclusion for visual keywords. Figure 5 shows the performance of smoothing techniques with different parameter settings. For Dirichlet smoothing (Figure 5(a)), the smoothing performance is keyframe dependent, which accurately adjusts the probability for unseen visual keywords. The relative weighting of visual keywords is emphasized when the parameter μ is small, so it has good performance. As μ becomes large, the weighting of visual keywords has less impact and is mainly dominated by the collection probability. The performance drops in this case. For Shrinkage smoothing and Mixture Model (Figure 5 (b) and (c)), the parameter (λ) is same for all keyframes, which is keyframe independent. When λ is high, the probability of visual keywords is mainly determined by background corpus model, which cannot provide an accurate estimation. So the performance is relatively poor. When λ is small, it emphasizes more on the relative visual term weighing. The probability of visual keywords is controlled more by the keyframe model, and less by corpus model. Therefore, the performance improves. Mixture Model increases the keyword probability in the keyframe model for keywords that occurred frequently in a keyframe than in the background. It achieves more accurate keyword probability estimation for each keyframe, whose performance approaches OOS. The accuracy of top-1 retrieval is around 0.8 and top-5 reaches 0.9.

6.3 Fusion of Visual and Context Similarity

In this section, we experiment various combinations of visual and contextual cues under four different similarity measures. In total,

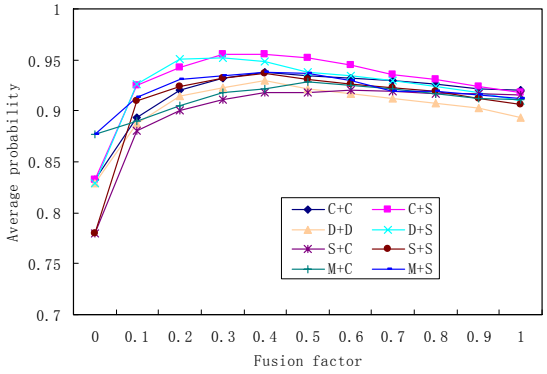

Figure 6. Effects of the fusion factor

there are 16 (4×4) combinations, considering the cosine similarity and 3 language models on visual and context. These combinations are further compared against 8 different visual-only and text-only performances. To distinguish fusion results, we use the notation Measure_V+Measure_T, indicating the similarity measure for visual (V) and context (T) respectively. We experiment: Cosine similarity (C), Dirichlet smoothing (D), Shrinkage smoothing (S) and Mixture model (M). For example, M+D denotes that this method uses a Mixture Model to measure the visual similarity and Dirichlet smoothing to compare the context similarity, and then fuses them together to obtain the final results. For each combination, we implement different fusion settings and report only the best result.

First, we study the effect of the fusion factor α in Equation (8). For simplicity, 8 combinations are randomly selected to show in Figure 6. The general trend is the performance increases at first and then drops. When α is zero, semantic context has no effect and the measures rely only on visual similarity. When the fusion factor increases, the context similarity takes part. We can see an obvious performance improvement as the factor increases from 0 to 0.1, which proves that the context similarity provides a meaningful complement for the visual contents. For two keyframes having high visual similarity, if their context similarity is also high, they have a strong confidence to be NDK. When their context is unrelated, the possibility to be near-duplicates is low. The context similarity gives an indication of being NDK for keyframes having high visual similarity. As the fusion factor increases, the context similarity begins to dominate and the performance drops. Obviously having high context similarity

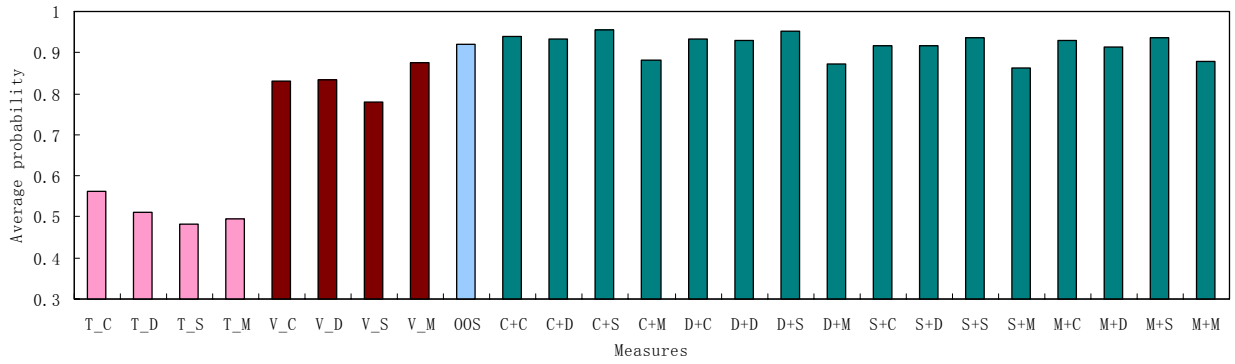

Figure 7. Performance comparison of different measures

(T – Context similarity, V – Visual similarity, C – Cosine, D – Dirichlet smoothing, S – Shrinkage smoothing, M – Mixture model)

(**Measures on context similarity**, **Measures on visual similarity**, **OOS**, **Combination of visual and contextual similarity**)

alone cannot confirm the identity of NDK. We can arrive at the conclusion only if they also have high visual similarity.

Figure 7 summarizes the AP performance of multiple and single modality approaches, in which the measures on context similarity, visual similarity, and their combinations are labeled in pink, brown, and green, respectively. We also compare them with the one-to-one symmetric matching (OOS). It is obvious that measures on context similarity have poor performance simply because stories may include multiple keyframes which share the same context. Although NDK are potentially existed within these keyframes, all keyframes with similar context will be falsely treated as NDK. On the other hand, measures on visual similarity have superior performance than methods on context. Visual keywords based approaches are effective to capture the complex variations among NDK. However, due to the noise and effects of keypoint clustering, it might result in Non-NDK having high visual similarity, especially for keyframes with cluttered contents. With the assistance from context information, such kind of false alarms can be effectively eliminated. We can see that the fusion of the visual and context cues improves the performance of individual measures (almost) across all the combinations of similarity measures. Their performances approach and even outperform the best OOS (Figure 7). Visual keywords and text context evaluate the likelihood of near-duplicate from the visual and semantic viewpoints respectively, and their combination reinforces the confidence of NDK and weakens the weight of Non-NDK. Figure 8 shows four pairs of Non-NDK keyframes having high visual similarity on visual keywords. Due to the complex scene content, keypoints are falsely grouped, resulting in high visual similarity. However, their context is totally different. For example, the "Arkansas school shooting" has nothing overlap with the "health" topic, and thus can be easily filtered by noticing the difference in semantic context. To certain extent, the visual and context similarity complement each other and lead to performance improvement.

6.4 Speed Efficiency

Table 1 shows the total retrieval time for 150 NDK queries for each method. These experiments were tested on a Pentium-4 machine with 3G Hz CPU and 512M main memory in Windows-XP environment. The clustering of keypoints was performed offline, which is processed at the indexing stage, so the time is not included. The time for comparing semantic context and fusing visual and context similarity is rather quick. All computation is finished within one second except Dirichlet smoothing, which is less than 10 seconds. So we do not show the speed efficiency in this table.

Figure 8. Non-NDK pairs having high visual similarity on visual keywords, but different semantic context

Table 1. Speed Efficiency

Methods	OOS	VK_C	LMVK			CM
			D	S	M	
Time	6.49h	16"	6'15"	1'02"	1'32"	3'26"

As seen in Table 1, VK based approaches are generally fast. For instance, LMVK with Shrinkage smoothing can answer about 150 queries per minute, even faster than CM. OOS, although capable of guaranteeing stable and unique matches among keypoints, is extremely expensive due to the large pool of keypoints. In the experiments, we use 128-dimension SIFT, instead of 36-dimension PCA-SIFT in [9], so as to show the best possible performance with keypoint matching. If PCA-SIFT and LIP-IS index structure [23] are used, the speed is slightly above 30 minutes which is still considered high for online search. Visual keywords based approaches are much faster than the one-to-one matching. Compared to the exhaustive keypoint matching, their computation is performed among visual keywords, which greatly accelerates the process. For the measures on visual keywords except Dirichlet smoothing, only visual keywords appeared in either keyframes are needed to calculate the weight or probability, leading to fast computation. Dirichlet smoothing calculates the probability of all visual keywords, which results in slower speed than the other language models.

7. CONCLUSION

In this paper, we propose a novel NDK retrieval approach by exploiting visual keywords and semantic context to meet the

requirement of online retrieval. Cosine similarity and language models are studied and experimented to explore the usefulness of visual keywords. Furthermore, the visual and context similarity are linearly fused to fully exploit the advantages of both modalities. Experiments on a subset of TRECVID 2004 show that:

- Bag-of-words representation is both effective and efficient for NDK retrieval. Both cosine similarity and language models show reasonably good performance on visual keywords.
- There is no obvious winner between cosine similarity and language models on visual keywords. Cosine similarity appears robust and no parameter setting is involved. As in text retrieval [18], we find that language models are sensitive to smoothing techniques and their parameters.
- Mixture model can accurately estimate the probability of visual keywords, which demonstrates the best performances among all measures. The retrieval precision indeed approaches the techniques with keypoint matching. Meanwhile, the speed is even faster than the baseline retrieval with color moment.
- Semantic context is a useful cue for NDK retrieval. By complementing context to visual words, the performance can exceed the techniques with keypoint matching, while enjoying the merit of speed efficiency.
- Using both visual keywords and semantic context, the online and accurate retrieval of NDK pairs become feasible. This also enlightens the efficient ways of mining NDK in large video database for online large-scale video search.

While encouraging, there are several issues worth further studying. First, visual keywords are not identical to text words. The factors such as vocabulary size and corpus diversity may determine the generation of visual keywords, and eventually affect the performance of NDK retrieval. While we experiment text-based language models for visual keywords, a valid question is whether we need "pure" language models which can naturally take the outcome of clustering (e.g., size and density of a cluster) into consideration. Furthermore, keypoints are detected from the gray level images and the color information is missing. Color has been shown to be essential in many retrieval tasks. In the future, we will explore the possibility of fusing color information as another complement.

8. ACKNOWLEDGMENTS

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 118905) and a grant from City University of Hong Kong (Project No. 7002112).

9. REFERENCES

[1] J. Allan, C. Wade, and A. Bolivar. Retrieval and Novelty Detection at the Sentence Level. *ACM SIGIR'03*.

[2] J. Amores, N. Sebe, P. Radeva, T. Gevers, and A. Smeulders. Boosting Contextual Information in Content-Based Image Retrieval. *ACM MIR'04*, 2004

[3] S.-F. Chang and et al. Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction. *TRECVID 2005*, Washington DC, 2005.

[4] M. Davis, S. King, N. Good, and R. Sarvas. From Context to Content: Leveraging Context to Infer Media Metadata. *ACM MM'04*, 2004.

[5] P. Duygulu, J.-Y. Pan and D. A. Forsyth. Towards Auto-Documentary: Tracking the Evolution of News Stories. *ACM MM'04*, USA, Oct. 2004, pp. 820-827.

[6] W. H. Hsu and S.-F. Chang. Topic Tracking across Broadcast News Videos with Visual Duplicates and Semantic Concepts. *ICIP'06*, Atlanta, GA, October 2006.

[7] J. L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. *Speech Communication*, 2002.

[8] J. Li. A Mutual Semantic Endorsement Approach to Image Retrieval and Context Provision. *ACM MIR'05*, 2005.

[9] Y. Ke, R. Suthankar, and L. Huston. Efficient Near-Duplicate Detection and Sub-Image Retrieval. *ACM MM'04*.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91-110, 2004.

[11] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *CVPR'03*, pp. 257-263.

[12] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *IJCV*, 60 (2004), pp. 63-86.

[13] C.-W. Ngo, W.-L. Zhao and Y.-G. Jiang. Fast Tracking of Near-Duplicate Keyframes in Broadcast Domain with Transitivity Propagation. *ACM MM'06*, pp. 845-854, 2006.

[14] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. *ACM SIGIR'98*.

[15] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. *ICCV'03*.

[16] TRECVID, http://www-nlpir.nist.gov/projects/trecvid/.

[17] X. Wu, C.-W. Ngo and Q. Li. Threading and Autodocumenting News Videos. *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 59-68, March 2006.

[18] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *ACM SIGIR'01*, USA, pp. 334-342, Sep. 2001.

[19] Y. Zhai and M. Shah. Tracking News Stories across Different Sources. *ACM MM'05*, Singapore, Nov. 2005.

[20] D.-Q. Zhang and S.-F. Chang. Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning. *ACM MM'04*, pp. 877-884, 2004.

[21] Y. Zhang, J. Callan, and T. Minka. Novelty and Redundancy Detection in Adaptive Filtering. *ACM SIGIR'02*, 2002.

[22] Y. Zhang, W. Xu, and J. Callan. Exact Maximum Likelihood Estimation for Word Mixtures. *Text Learning Workshop at ICML'02*.

[23] W.-L. Zhao, C.-W. Ngo, H.-K. Tan and X. Wu. Near-Duplicate Keyframe Identification with Interest Point Matching and Pattern Learning. *IEEE Trans. on MM*, 2007.

[24] Y. Zheng, S.-Y. Neo, T.-S. Chua and Q. Tian. Fast Near-Duplicate Keyframe Detection in Large-Scale Video Corpus for Video Search. *IWAIT'07*.