# EFFICIENT NEAR-DUPLICATE KEYFRAME RETRIEVAL WITH VISUAL LANGUAGE MODELS

*Xiao Wu, Wan-Lei Zhao and Chong-Wah Ngo*

Department of Computer Science, City University of Hong Kong

## ABSTRACT

Near-duplicate keyframe retrieval is a critical task for video similarity measure, video threading and tracking. In this paper, instead of using expensive point-to-point matching on keypoints, we investigate the visual language models built on visual keywords to speed up the near-duplicate keyframe retrieval. The main idea is to estimate a visual language model on visual keywords for each keyframe and compare keyframes by the likelihood of their visual language models. Experiments on a subset of TRECVID-2004 video corpus show that visual language models built on visual keywords demonstrate promising performance for near-duplicate keyframe retrieval, which greatly speed up the retrieval speed although sacrifice a little performance compared to expensive point-to-point matching.

## 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 [2, 5, 11] plays an important role in measuring video clip similarity, tracking video shots of multi-lingual sources, and threading news stories under the same topic.

Recently, approaches based on keypoints have demonstrated promising performance on object matching [7] and near-duplicate keyframe detection [2]. Keypoints are salient regions detected over image scales and their descriptors of keypoints are invariant to certain transformations that exist in different images. In [14], keyframes were first partitioned into small groups and then performed keypoint matching. Our previous work [5] based on one-to-one symmetric matching (OOS) also showed good results on NDK detection. However, due to the large number of keypoints in one keyframe (may over a thousand), matching keypoints between two keyframes is computationally expensive and makes on-line retrieval infeasible.

To tackle this problem, a visual vocabulary is constructed in [7] to offline quantize the keypoints by clustering keypoints into clusters. Similar to the traditional document composed of text words, a keyframe can be regarded as a set of visual keywords. Techniques employed on text retrieval can be applied to keyframe retrieval. Comparison of two keyframes can be converted to compare two distributions or vectors built on visual keywords.

Language modeling methods have been successfully applied to speech recognition, machine translation, and natural language processing, which attract a lot of research attentions due to its foundation in statistical theory. More recently, the language modeling framework has been introduced to information retrieval [6], and has performed well empirically [1, 12]. 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. However, visual keywords are different from traditional text words, and it is uncertain whether language models built on visual keywords are similar to text language models and effective for near-duplicate keyframe retrieval.

In this paper, we explore visual language models built on visual keywords to speed up the near-duplicate keyframe retrieval, instead of exhaustive point-to-point matching. The basic idea is to estimate a language model for each keyframe and then compare keyframes by the likelihood of their language models.

## 2. VISUAL KEYWORDS

Keypoints are salient regions detected over image scales. Currently, there are a couple of keypoint detectors and descriptors [4]. The detectors basically locate stable keypoints (and their support regions) which are invariant to certain variations introduced by geometric and photometric changes. And the descriptors of keypoints are invariant to certain transformations that exist in different images. In this paper, we adopt Hessian-Affine [4] as the keypoint detector, and SIFT [3] as the descriptor. SIFT (Scale-Invariant Feature Transform) has shown to be one of the best descriptors for keypoints, which is a 128-dimensional feature vector that captures the spatial structure and the local orientation distribution of a patch surrounding keypoints.

Clustering algorithm is then applied on these keypoints to group keypoints into clusters, which constructs a visual vocabulary. All keypoints in a cluster correspond to one visual keyword. Similar to traditional documents that are

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).

treated as a bag of words, we can regard that each keyframe is composed of a bag of visual keywords. Statistically, keyframes can be represented as smoothed probability distributions over the visual keywords. The techniques used in the textual features can be as well applied to the keyframes.

## 3. VISUAL LANGUAGE MODEL

A *visual language model* is a probability distribution that captures the statistical regularities of visual keywords. Given two visual 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 visual language model. The similarity between two keyframes is measured by the *Kullback-Leibler (KL)* divergence between two visual language models.

In the visual language model approach, a keyframe is represented by a unigram visual keywords distribution $\theta$. We assume that a keyframe $K_i$ is generated by a probabilistic model based on keywords. In the visual 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 $C$.

### 3.1. Symmetric KL Divergence Measure

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 near-duplicate keyframes, their similarity should be symmetric. So we propose the symmetric *KL* divergence to measure the similarity between two keyframes. This symmetric property is emphasized so that if keyframe *A* matches to *B*, then *B* is also near-duplicate with *A*. This property makes the measure stable. The similarity measure (i.e. symmetric KL) is defined as:

$$S(K_i \mid K_j) = (-KL(\theta_i,\theta_j) - KL(\theta_j,\theta_i))/2$$

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

where $\theta_i$ is the visual language model for keyframe $K_i$, which is a multinomial distribution. Here $p(v_k|\theta_i)$ is the probability of visual term $v_k$ occurring in keyframe $K_i$, similarly for $p(v_k|\theta_j)$. The *KL* divergence can be regarded as a distance between distributions. 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:

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

where $tf(v_k, K_i)$ is the term frequency of visual keyword $v_k$ in keyframe $K_i$. However, the problem of maximum likelihood estimation is that it will generate a zero probability if a 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. The general form of a smoothed model is as follows:

$$p(v_k \mid \theta_i) = \begin{cases} p_s(v_k \mid \theta_i) & \text{if visual keyword } v_k \text{ is seen} \\ \alpha_k p(v_k \mid C) & \text{otherwise} \end{cases}$$

where $p_s(v_k|\theta_i)$ is the smoothed probability of a visual keyword seen in the keyframe $K_i$, $p(v_k|C)$ is the collection language model, and $\alpha_k$ is a coefficient using for controlling the probability of unseen visual keywords. The sum of all probabilities is equal to one. In our experiments, the collection model is built on all the keyframes in the corpus.

Prior research on textual information retrieval [10] shows that different smoothing techniques highly affect the performance. For visual language model, we mainly use Bayesian smoothing with *Dirichlet* priors and *Shrinkage*. Furthermore, *Mixture model* is also experimented.

### 3.2. 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 the visual keywords. For a *Dirichlet* distribution with parameters:

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

the posterior distribution using Bayesian analysis is:

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

$$\alpha_k = \frac{\mu}{\sum_{v_k} tf(v_k, K_i) + \mu}$$

$p(v_k|C)$ is the collection visual language model and $\mu$ is a parameter to adjust the degree of smoothing.

### 3.3. Shrinkage Smoothing

*Shrinkage* smoothing is a special case of *Jelinek-Mercer* smoothing method, which involves a linear interpolation of the maximum likelihood model with *n*-gram model [10]. Based on the assumption that a keyframe is generated by sampling from two different visual language models: a keyframe visual model and a collection visual model, the visual language model of a keyframe is determined by:

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

using coefficients $\lambda$ ($\alpha_k=\lambda$) to control the influence of each visual model.

$\theta_{MLK}$ is the maximum likelihood visual language model of the keyframe and $\theta_{MLC}$ is the maximum likelihood visual language model of the collection.

### 3.4. Mixture Model

A keyframe is assumed to be generated by the mixture of two different visual language models: a keyframe-specific

visual model $\theta_K$, and a visual model for the collection $\theta_C$. Mixture model [12] is based on the opposite assumption that keywords occurred frequently in a keyframe than in the collection should have higher probability in the keyframe model. Therefore, the approach is to deduce the maximum likelihood keyframe visual model. Each visual keyword in the keyframe is generated by the two visual language models with probability *(1-λ)*, and $\lambda$ respectively.

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

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 [12]. Similar to [1], the visual language model $\theta_K$ that maximizes the likelihood of the observed keyframe, given fixed parameters, was computed using the technique described in Zhang et al. [13].

## 4. EXPERIMENTS

### 4.1. Data Set and Performance Metric

We use the data set given by [11] for evaluation, which is a subset of TRECVID-2004 corpus [8]. 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 method in [7]. Keypoints are extracted with Hessian Affine [4] and described by SIFT [3]. Traditional *k*-means algorithm is employed to group keypoints (77,706) into 3500 clusters, in which each cluster represents a visual keyword.

We use all NDK (300 keyframes) as queries for NDK retrieval in the experiments. The retrieval performance is evaluated with the probability of the successful top-*k* retrieval, defined as:

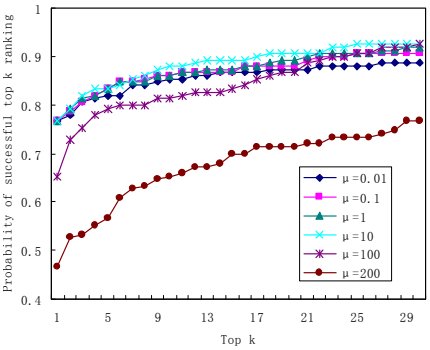
$$R(k) = \frac{Qc}{Qa}$$

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

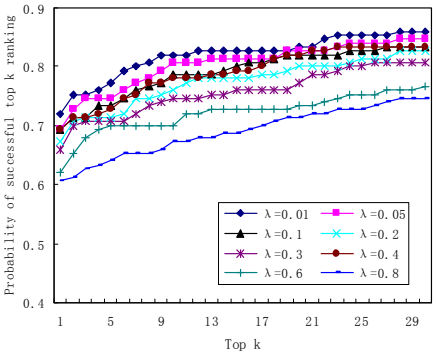
### 4.2. Effect of Visual Language Models

Traditional language models on text are sensitive to the smoothing methods and the parameter settings [10]. Visual language models have the similar phenomenon. Figure 1 shows the performance of smoothing techniques with different parameters.

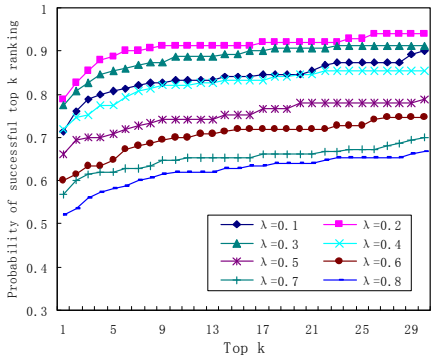
For Dirichlet smoothing (Figure 1(a)), the smoothing performance is keyframe dependent. The relative weighting of visual keywords is emphasized when the parameter $\mu$ is small. It achieves good performance when $\mu$ is small. As $\mu$ becomes large, the coefficient $\alpha_k$ increases. The weighting of visual keywords has less impact and is mainly dominated by the collection probability. The performance drops in this case.




(a) LMVK_Dirichlet




(b) LMVK_Shrinkage




(c) LMVK_Mixture_model

**Figure 1. Performance of smoothing**

For Shrinkage smoothing and Mixture model (Figure 1 (b) and (c)), the parameter ($\lambda$) is same for all keyframes, which is keyframe independent. When $\lambda$ 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 $\lambda$ is small, it emphasizes more on the relative visual term weighing. The probability of visual keywords is controlled more by the keyframe visual model, and less by corpus model. Therefore, the performance improves.

### 4.3. Overall Performance Comparison

To study the overall performance of visual language models built on visual keywords (LMVK), we compare it with one-to-one symmetric matching (OOS), and block-based color moment (CM). OOS matching was proposed by our previous work [5] to guarantee the reliable matching among keypoints. The ranking is based on the cardinality of keypoints being matched. For CM, each keyframe is depicted with the first three color moments extracted in *Lab* color space over 5 × 5 grid partitions. For visual language models, we test Dirichlet smoothing (LMVK_D), Shrinkage smoothing (LMVK_S) and Mixture model (LMVK_M), and the best result for each method is selected for comparison.

The performance comparison is shown in Figure 2. OOS symmetric matching guarantees stable and unique matches among keypoints, which achieves the best performance. But it is expensive. Visual language models demonstrate promising performance. Mixture model accurately estimates the keyword probability for each keyframe, whose performance approaches OOS. Dirichlet smoothing is more effective than Shrinkage smoothing. Although other factors may affect the performance of visual language models (e.g. vocabulary size, "polysemy and synonymy"), they have shown the potential to estimate the probability of visual keywords and measure their similarity under complicated variations. Because of various variations of NDK (e.g. lighting condition, editing, viewpoint), CM does not perform well.

### 4.4. Speed Efficiency

**Table 1. Speed Efficiency**

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

Table 1 shows the total retrieval time for 150 NDK queries for each method. These experiments are tested on a Pentium-4 machine with 3G Hz CPU and 512M main memory in Windows-XP environment.

OOS is extremely expensive due to the large amount of keypoints available for matching between every keyframe pair. In the experiments, we use 128-dimension SIFT, instead of 36-dimension PCA-SIFT as in [5], so as to show the best possible performance with keypoint matching. If PCA-SIFT and LIP-IS index structure [5] are used, the speed is slightly above 30 min which is still considered high for online search. The approaches with visual language models are much faster than the one-to-one matching. Compared to exhaustive keypoint matching, their computation is performed among visual keywords, which greatly accelerates the process. Dirichlet smoothing calculates the probability of all visual keywords instead of keywords just appeared in both keyframes, which results in slower speed than the other language models.

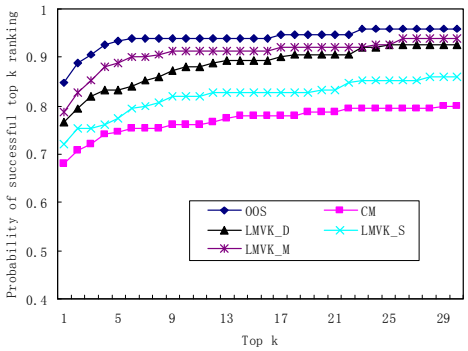



**Figure 2. Performance Comparison**

## 5. CONCLUSION

In this paper, we present the visual language models built on top of visual keywords to retrieve near-duplicate keyframes, which shows promising performance and accelerates the retrieval speed significantly. However, visual keywords are not identical to text words. They have unique properties that are worth exploring further. Furthermore, keyframes are usually self-contained within a certain context (e.g. news stories, web pages). The possibility of having NDK is rather high when their contexts are similar (e.g. discussing the same event) [9]. We will integrate the text features to improve the retrieval accuracy and speed in the future.

## 6. REFERENCES


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

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

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints", Int. Journal on Computer Vision, 60(2):91-110, 2004.

[4] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector", ECCV, 2002.

[5] 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.

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

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

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

[9] X. Wu, W-L. Zhao, and C-W. Ngo. "Near-Duplicate Retrieval with Visual Keywords and Semantic Context", ACM CIVR'07.

[10] 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.

[11] D.-Q. Zhang, and S.-F. Chan, "Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning", ACM MM'04, pages 877-884, 2004.

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

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

[14] 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.