The Use of Temporal, Semantic and Visual Partitioning Model for Efficient Near-Duplicate Keyframe Detection in Large Scale News Corpus

Yan-Tao Zheng[1], Shi-Yong Neo[1], Tat-Seng Chua[1], Qi Tian[2]
[1]National University of Singapore, 3 Science Dr, Singapore 117543
[2]Institute for Infocomm Research (I^2R), 21 Heng Mui Keng Terrace, Singapore 119613
{yantaozheng, neoshiyo, chuats}@comp.nus.edu.sg, tian@i2r.a-star.edu.sg

ABSTRACT

Near-duplicate keyframes (NDKs) are important visual cues to link news stories from different TV channel, time, language, etc. However, the quadratic complexity required for NDK detection renders it intractable in large-scale news video corpus. To address this issue, we propose a temporal, semantic and visual partitioning model to divide the corpus into small overlapping partitions by exploiting domain knowledge and corpus characteristics. This enables us to efficiently detect NDKs in each partition separately and then link them together across partitions. We divide the corpus temporally into sequential partitions and semantically into news story genre groups; and within each partition, we visually group potential NDKs by using asymmetric hierarchical k-means clustering on our proposed semi-global image features. In each visual group, we detect NDK pairs by exploiting our proposed SIFT-based fast keypoint matching scheme based on local color information of keypoints. Finally, the detected NDK groups in each partition are linked up via transitivity propagation of NDKs shared by different partitions. The testing on TRECV 06 corpus with 62k keyframes shows that our proposed approach could result in multifold increase in speed as compared to the best reported approach and complete the NDK detection in a manageable time with satisfactory accuracy.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]

Keywords

Near-duplicate keyframe detection, corpus partitioning

1. INTRODUCTION

Near-duplicate keyframes (NDKs) denote a group of keyframes that depict the same or duplicate scene in the whole or part of the image but with slightly varying visual appearance [17]. The reason for visual difference is due to geometric, photometric and scale changes caused by the variance of camera shooting angle, lighting condition, camera sensor or video editing process. Figure 1 shows some examples of NDKs from TRECV 06 corpus [16].

The task of detecting NDKs in video corpus is important as it helps to link relevant news stories across different TV news channel, language and time. This type of visual cues is especially useful in news video retrieval, topic detection and tracking and story threading. In the interactive [1] and automated [2] search system, NDK detection results have been exploited and reported to be effective to boost search performance. Moreover, Hsu et al. [4] also found that the NDK detection results can improve the topic tracking performance significantly.

Recently, researchers have formulated the NDK detection as a keypoint-based image matching problem [8] [14], which has been found effective because of its robustness to scale, illumination changes and partial occlusions, etc. However, one of the major challenges of keypoint-based NDK detection is the speed efficiency, as the keypoint matching is a slow process especially when we need to identify NDK pairs in a search space of $N \times N$ potential pairs, where N is the number of keyframes. This speed issue is especially critical for large-scale news video corpus such as the TRECV 04/05 and TRECV 06 with 65k and 62k live scene keyframes respectively. It is natural to address the speed issue by indexing the image descriptors of the image corpus [8]. This approach can speed up NDK detection by accelerating the database accessing speed. However, it does not resolve the intrinsic quadratic complexity issue, where each keyframe is attempted to match against all the rest of keyframes in the corpus. Consequently, the computational cost still increases quadratically as the number of keyframes increases.

It is observed that as compared to the number of outlier non-NDKs, the number of NDKs is fairly small; and the visual appearance of its non-NDKs can be fairly disparate to the keyframe. Moreover, the popular keypoint-based image matching method, like Scale Invariant Feature Transform (SIFT) [9] or PCA-SIFT [7], is an expensive computational process, due to the huge number of keypoints in each keyframe (around 300 - 1k keypoints per keyframe). Therefore, it is preferable to group potential NDKs into small partitions of size $N^{'}$ ($N^{'} << N$) and perform the expensive image matching only within each small partition with the

WOODSTOCK '97 El Paso, Texas USA

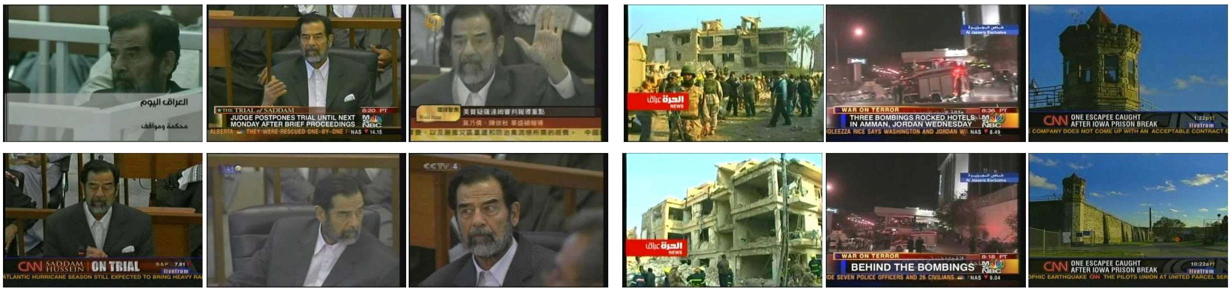
THE TRIAL of SADDAM
JUDGE POSTPONES TRIAL UNTIL NEXT MONDAY AFTER BRIEF PROCEEDINGS
WAR ON TERROR
THREE BOMBINGS ROCKED HOTELS IN AMMAN, JORDAN WEDNESDAY
ONE ESCAPEE CAUGHT AFTER IOWA PRISON BREAK
SADDAM HUSSEIN ON TRIAL
WAR ON TERROR
BEHIND THE BOMBINGS
ONE ESCAPEE CAUGHT AFTER IOWA PRISON BREAK

Figure 1: Examples of near-duplicate keyframes from TRECVID06. The 6 images on the left column are NDKs about news of Sadam trial from MSNBC, CCN, LBC, CCTV and NTDTV

computational complexity of $O((N^{'})^2)$ rather than $O(N^2)$.

To achieve this, we exploit the domain knowledge and corpus characteristics to partition the keyframe based on three dimensions of temporal, semantic and visual.

(a) Temporal Partition: The temporal distribution of NDKs [14] makes temporal partition an effective strategy to group potential NDKs together. This is because NDK pairs are rarely found in temporally distant news videos. By allowing overlapping between adjacent temporal partitions, NDKs from different partitions can be linked based on transitivity propagation.

(b) Semantic Partition: The efficiency improvement of temporal partition is limited as each partition must be long enough to cover the majority of NDKs during a certain period. To further reduce the NDK detection complexity, for each temporal partition, we divide it semantically into news story genre groups. It is observed that the same group of NDKs usually come from the same continuing or relevant news and therefore, they are seldom from different news story genres. By utilizing this news domain knowledge, we can reduce the complexity of NDK detection in each temporal partition by a factor of G, where G the number of news story genres.

(c) Visual Partition: On top of the temporal and semantic partitions, we further visually partition them into small groups by performing clustering based on a set of robust semi-global image features. The proposed semi-global features are designed to be resistent to photometric distortions, geometric deformations and robust to scale changes among NDKs. The robust semi-global image features function as a relaxation matching criteria that enables potential NDKs with similar feature values to be clustered together, though they are not necessarily NDKs.

Finally within each small visual partition, we perform the accurate SIFT-based image matching to detect NDKs. We speed the keypoint matching drastically by checking the local color of keypoints before matching the SIFT descriptors, as the truly corresponding keypoints must share similar local color due to the high repeatability of keypoints.

After the NDK detection in each partition, the NDKs are linked by the transitivity propagation based on shared NDKs from different partitions. This is because one keyframe could belong to multiple partitions due to the partition overlappings.

The two main contributions of our approach are: (1) By exploiting the domain knowledge and corpus characteristics, we develop a temporal, semantic and visual partitioning model to divide the video corpus into many small overlapping partitions. Efficient NDK detection can therefore be achieved by performing image matching in each partition separately, and then combining the NDK detection results later. (2) We propose a fast keypoint matching scheme to drastically speed up image matching by checking the local color of keypoints before matching the keypoint descriptors. The experimental results show that our approach is able to reduce the NDK detection complexity drastically, while at the same time to improve the detection accuracy, as most of the outlier non-NDKs are excluded from image matching in each partition.

2. RELATED WORK

Many effective NDK detection algorithms have been developed. Duygulu, et al [3] proposed an approach to detect NDKs by ordering and examining the L neighbors of a keyframe and investigating the derivation of keyframe similarity. The weakness of this approach is that it is heuristic and sensitive to the setting of several empirical parameters. Zhang and Chang [17] proposed a NDK detection model based on stochastic attributed relational graph (ARG) matching. The ARGs matching, constrained by the spatial relation, makes it very accurate and effective for the NDK identification. However, the slow matching speed impedes its use in large-scale corpus. Ke [8] formulated the detection of near-duplicate images and sub-images into a part-based image matching task by representing the images with a bag of PCA-SIFT keypoints [7]. He addressed the speed issue by indexing the PCA-SIFT keypoints of all the keyframes with locality-sensitive hashing technique. Though the indexing can accelerate the accessing speed by optimizing the indexing layout and access to disk, the approach still attempts to match one query images against all the images in the database. This quadratic matching process makes it inefficient to use on large-scale dataset.

To overcome this problem, Ngo [14] exploited the temporal distribution and symmetry and transitivity characteristics of NDKs to divide the keyframe dataset into M temporal partitions and perform PCA-SIFT image matching only within each temporal partition. This approach improves the efficiency of NDKs identification by a factor of the M, the number of temporal partitions. However, this efficiency improvement is fairly limited, as M cannot be too large to ensure the duration of each temporal partition is long enough to cover the majority of NDKs at that period. The NDK detection speed reported in [14] is 2.3 days for 7K keyframes in TRECV 03. Even with the speedup of M times, it is still unacceptable to detect all NDKs in large-scale corpus

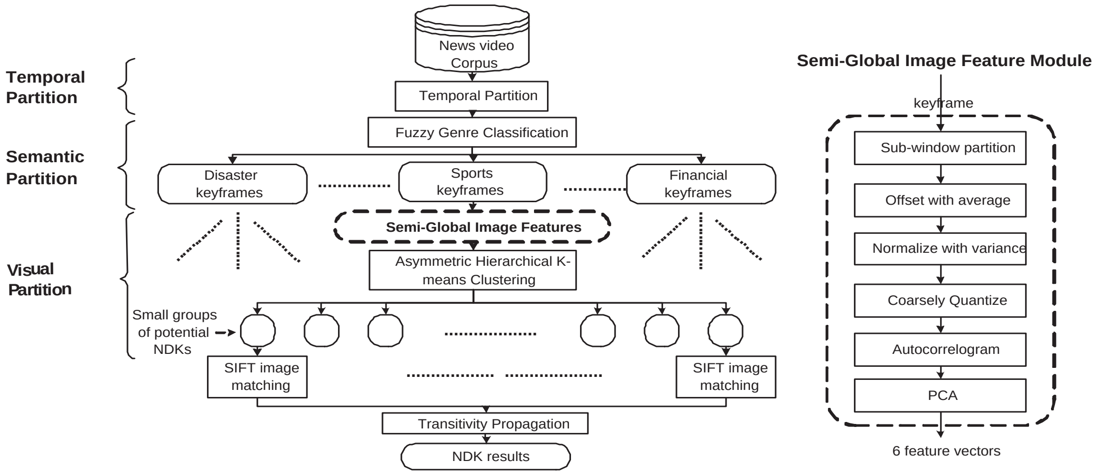

Figure 2: Framework of the proposed NDK detection system

such as TRECV 06. For example, TRECV 06 corpus consists of 62k live scene keyframes. With the approach of [14] and same parameter setting, the time for NDK detection in TRECV 06 will amount to $2.3 \times \mu(\pi_1/\pi_2)^2 \doteq 91$ days [1].

In [18], the authors proposed to visually group potential NDKs into small groups to further reduce the complexity. Comparing with previous approaches, our system are different mainly in three aspects. First, our approach aims to reduce the NDK detection complexity by dividing the corpus temporally, semantically and visually. Second, instead of solely relying on computationally expensive keypoint-based image matching, we exploit a set of robust semi-global image feature that functions as a relaxation matching criteria to cluster potential NDKs together. Third, rather than utilizing any indexing technique, we exploit the local color information of keypoints to achieve a fast image matching.

3. FRAMEWORK

As shown in Figure 2 , the proposed framework takes news corpus as input and preprocesses it by filtering out commercial, anchor person and weather forecast shots based on a rule-based system [2]. The approach then temporally partitions the corpus into M divisions, with v days overlapping between adjacent partitions. For each temporal partition, the approach divides it semantically into G news story genre groups and then visually clusters potential NDKs into small groups based on the proposed semi-global image features. In the small groups of potential NDKs, the approach performs SIFT-based image matching to detect NDK pairs. Finally, the detected NDKs in different groups are linked up by transitivity propagation based on NDKs that belong to multiple groups. We provide details of temporal and semantic partition here. The visual clustering of potential NDKs and accurate NDK detection will be discussed in Sections 4 and 5 respectively.

3.1 Temporal Partition

The framework temporally partitions the corpus into M divisions, in which the span of division is d days and the overlapping between adjacent divisions is v days. Let D be the total number of days under investigation, then M, d and v hold the following relation: $D \doteq M \times d - (M-1) \times v$. The v days overlapping between adjacent partitions provides a basis to link NDKs from different temporal partitions via transitivity propagation. The temporal partition can reduce the NDK detection complexity by a factor of M.

3.2 Fuzzy Semantic Partition

The semantic partitioning is to divide each temporal partition into G news story genre groups. The rational here is that the same group of NDKs are from related or continuing news stories or even the same repeated footage at different time, channels, etc. Therefore, they tend to fall into the same news topic, and more broadly, the same news story genre. While accurate topic tracking and detection remain a challenging problem, the news story genre classification can provide much more reliable results. However, its error rate is still unacceptable for subsequent NDK detection. The mean recall of normal genre classification can reach approximately 80%, which will propagate 20% misclassification errors to the subsequent NDK detection. Rather than classification accuracy, our concern is the recall or completeness of NDKs in each genre. Therefore, we propose a fuzzy news story genre partition, where one news story is allowed to be classified into more than one genres. In this way, the genre classification may lose precision, however, it can achieve a fairly high recall (completeness) for subsequent NDK detection.

The news story genres chosen for semantic partition are: *political* , *scientific*, *sports*, *financial*, *disaster* and *general (unclassified)* [13]. The genre classification is achieved by employing a text classifier over the automated speech recognition (ASR) transcripts as well as the temporal properties of the respective channels [13]. When the classification confidence of a news story for a particular genre is above a fuzzy threshold ε, we will assign it to the genre. If the classification confidences of the news story for all genres are below ε, we will assign the stories to *general (unclassified)* genre. In this way, we can achieve an average recall of 97.4% for the

[1] π_1 and π_2 are the average number of keyframes per partition for TRECV 06 and 03 respectively. In [14], the duration of each partition is 1 day, so π_1=62k/(59 days)=1.05k and π_2=7k/(30 days)=0.23k. μ=(59 days)/(30 days)=1.9.

5 genre groups.

The significance of news story genre partition are twofold. First, it can reduce the complexity of NDK detection for each temporal partition with a factor of G. Second, as the visual clustering of potential NDKs is actually a problem of clustering with outliers (non-NDKs), the news story genre partition can reduce the number of outliers for the visual clustering in next step drastically.

4. CLUSTERING OF POTENTIAL NDKS

The visual clustering aims to group NDKs into the same partition or link them together via transitivity propagation. To achieve this goal, we propose a set of semi-global image features that are resilient to visual differences of NDKs including: (1) photometric distortion like illumination change; (2) geometric deformations like viewpoint change, rotation, etc; and (3) scale changes like partially near-duplicate sub-images.

4.1 Robust Semi-Global Image Feature

4.1.1 Robustness to Photometric Changes

The photometric change describes the way in which the intensities in the red, green and blue channels (R,G, B) transform between NDKs. This kind of changes may be caused by changes of lighting or shooting condition, camera sensors, etc [12]. Mindru et al [12] proposed three models to describe the photometric transformation, which are *diagonal model, scaling and offset model*, and *affine model*. Let R denote a color pixel of a keyframe, and R' denote the corresponding color pixel of its NDK. The following equations mathematically depict the transformations of these three models:
Diagonal model:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} \doteq \begin{pmatrix} s_R & 0 & 0 \\ 0 & s_G & 0 \\ 0 & 0 & s_B \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1)$$

Scaling and an offset model:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} \doteq \begin{pmatrix} s_R & 0 & 0 \\ 0 & s_G & 0 \\ 0 & 0 & s_B \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_R \\ o_G \\ o_B \end{pmatrix} \quad (2)$$

Affine model:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} \doteq \begin{pmatrix} a_{RR} & a_{RG} & a_{RB} \\ a_{GR} & a_{GG} & a_{GB} \\ a_{BR} & a_{BG} & a_{BB} \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_R \\ o_G \\ o_B \end{pmatrix} \quad (3)$$

Here, we adopt the *scaling and offset model* to simulate the photometric changes among NDKs, as the model has been reported to be a good compromise between complexity and accuracy [8]. Note that the pre-condition of utilizing this model is that the lighting condition and illumination changes must be global and identical to all the scenes and objects in the image.

The photometric offset (o_R, o_G, o_B) of NDKs can be eliminated by subtracting the intensity of each channel with its average intensity; while the photometric scale (s_R, s_G, s_B) can be canceled out by normalizing the intensity with its variances [12]. Though the photometric transformations of NDKs can be partially eliminated by the process of offsetting and normalization, this process works well only on NDKs that are primarily near-duplicate in visual content. It might not be effective for sub-image NDKs, which share only partial near-duplicate visual content. To address this issue, the *"neutralized"* intensities are coarsely quantized. In our experiments, we adopt 4 bins to quantize each color channel. This wide quantization interval provides further tolerance on the illumination changes.

4.1.2 Robustness to Geometric Changes

The geometric deformations between NDKs are caused by different poses and viewpoints and it can be considered as an affine transformation between NDKs. Mindru and et al [12] proposed the generalized color moments to cancel the geometric deformation, which requires high computation. Here, we adopt the autocorrelogram [5] of coarsely quantized intensities that are already *"neutralized"* by offsetting and normalization to partially cancel the geometric deformation. Autocorrelogram depicts the global distribution of local spatial correlations of the pixels with the same color. When the measured distance of autocorrelogram is small (say 1, 3 or 5 pixels), the pixels in the color homogeneous regions in the keyframe play dominant roles in the autocorrelogram. When geometric deformation such viewpoint change, rotation, etc occurs, the homogeneous color regions remain homogeneous except for some changes to its size and shape. Thus, all the pixels, except the ones on the border of the region, will approximately give the same autocorrelogram values. Therefore, the autocorrelogram can be fairly resistant to geometric deformation and the respective autocorrelograms of NDKs are expected to be similar.

4.1.3 Sub-Image NDKs Detection with Sub-Windows

So far, the proposed image features have the risk of losing sub-image NDKs that are partially near-duplicate due to the camera movement or image scale changes. In order to address this problem, we spatially partition the frames into sub-windows to extract the local image features for each spatial partition. This is also the reason why we call ours features *semi-global*.

The problems now are how large the sub-window should be, how many sub-windows should be selected, and where they should be located in the keyframe. The choice of sub-windows size is actually a problem of scale selection. Ideally, it should correspond to the scale changes between NDKs, which is a difficult and computationally expensive problem. As the proposed image features function as a relaxation matching procedure before keypoint-based image matching, we can determine the sub-window size by estimating how large scale changes the keypoint-based image matching, like SIFT [9], can handle. We evaluate the robustness of scale variation by examining the number of keypoints generated by the keypoint detectors, as more keypoints produce a higher chance to match its NDKs at the respective scale level. The keypoint detectors examined are: Hessian-affine detector [11] and Maximally Stable Extremal Region (MSER) detector [11], which are detectors used in our system. Figure 3 displays the average number of keypoints generated in different octave scale levels from 100 randomly selected keyframes from TRECV 06 corpus. Each successive octave means the image is spatially downsampled by half. Scale level of octave 0 means the keyframe is of its original size. As shown in Figure 3, when the scale goes to octave 2 and beyond, the number of keypoints becomes deficient to produce positive matches. This is because the TRECV 06

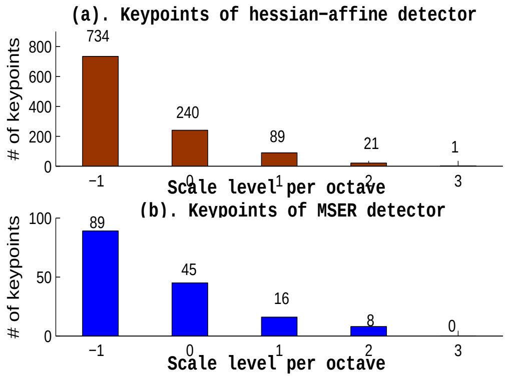

Figure 3: The number of keypoints versus the scale levels. (a) keypoints generated by Hessian-affine detector; (b) keypoints generated by MSER detector

news video is of MPEG1 format and their keyframes have limited resolution of 352 × 240.

As the keypoint-based image matching cannot effectively detect NDKs with more than 1 octave scale changes, we set the size of sub-window to 1 octave scale (one quarter) of the original keyframes. Moreover, we adopt a heuristic approach to partition a keyframe into 5 sub-windows positioned at upper-left, upper-right, lower-left, lower-right quarter and central quarter. Instead of concatenating image features from five sub-windows, we regard each sub-window as a separate frame and extract image features of them respectively. This is because the spatial arrangement may not be the same in different NDKs. Altogether, a keyframe is represented by a group of 6 feature vectors $\{v_i\}$, where $i = 1$ refers to the whole keyframe and $i = 2, ..., 6$ are for the 5 sub-windows.

The sub-window partitions of NDKs may not exactly cover the same scene or objects; however, the robustness and flexibility of the proposed image feature can limit this disadvantage.

4.1.4 Dimensionality Reduction by PCA

Due to the spatial continuity of keyframe, the dimensions of the semi-global feature vectors have strong correlations with each other. Therefore, prior to clustering, we apply Principal Component Analysis (PCA) [6] to reduce the number of dimensions. In the experiments, we quantize the color intensity in R, G, B channel into 4 bins, and set the distance parameters as [1, 3, 5] for autocorrelogram. Consequently, the dimensionality of feature vector is $4 \times 4 \times 4 \times 3 = 192$. We perform PCA to reduce it to 52, where the sum of eigenvalues of selected new dimensions is above 95% of all dimensions. This dimensionality reduction does not only reduce the computational cost, but also contributes to more accurate and distinctive representation of keyframes.

4.2 Asymmetric Hierarchial K-means Clustering on Semi-Global Image Features

The visual clustering of potential NDKs is actually a large-scale clustering task with many outliers. In order to achieve both efficiency and accuracy, we need to address two problems: (1) despite of many outliers, the same group of NDKs should be clustered together; and (2) the clustering process should be efficient as the number of keyframes is large. To address these two problems, we adopt the *asymmetric hierarchial k-means clustering*, which performs k-means clustering recursively on the dataset with minimum cluster size constraint. This clustering approach can reasonably address the first problem by setting the number of clusters k = 2 for each k-means. With k = 2, k-means behaves like a dichotomizer and clusters the data sample based on its distance from two cluster centroids. As true NDKs tend to have similar feature values, they are highly probable to be assigned to the same cluster at each level of k-means clustering. The efficiency is achieved by the hierarchical approach of clustering as well. In the experiments, the whole clustering process with on TRECV 06 corpus can be accomplished within 10 min.

One problem with hierarchial k-means clustering is that it overlooks the within-cluster scatter and the size of the cluster; it simple performs the hierarchial clustering until the required depth α is reached. Here we propose an *asymmetric hierarchial k-means clustering* by specifying a constraint that when the size of a cluster is reduced below $s_{constraint}$, there will be no recursive clustering on this partition. Consequently, the number of clusters will be less than 2^{α}, as α increases. More importantly, the resulting clusters may not be in the same level of the hierarchy and the large clusters with many outliers can be splitted further without dividing the desired coherent small-size clusters of NDKs. In the experiments, we set $s_{constraint}$ to be the number of unique keyframes rather than the number of data samples in the cluster.

Besides efficiency and accuracy, the asymmetric hierarchical k-means clustering gives us another advantage of natural hierarchical organization of clusters. If the search on NDKs is not satisfactory, the search can go up by one level to find NDKs in a coarser and larger cluster. This is to address the following error condition: as there exist many outlier keyframes (non-NDKs) and the clusters might not be well separable clouds of points, it is possible that NDKs might be separately grouped in the neighboring clusters.

5. ACCURATE NDK IDENTIFICATION VIA KEYPOINT-BASED IMAGE MATCHING

Within the small partitions resulted from the temporal, semantic partition and visual clustering of potential NDKs, we perform the accurate NDK identification via the keypoint-based image matching approach. In keypoint-based image matching, each image is represented by a number of keypoints and the matching of two images is done by matching the descriptors of keypoints from two images respectively. The successful NDK detection systems ([14] [8]) have demonstrated the effectiveness of keypoint-based image matching. Differing from [14] and [8], our approach focuses on the aspects of keypoint detection and descriptor matching to achieve better accuracy and speed efficiency.

5.1 Keypoint Detection

The keypoint is detected in both spatial and scale space. Its location determines which local characteristics of the image are extracted for image matching. A number of keypoint detectors are available, such as Difference of Guassian (DoG) [9], Maximally Stable Extremal Region (MSER) [11], Harris-affine [11], Hessian-affine [11], etc. Each keypoint detector has its preferences of responding to different image features and shapes, and therefore, delivers different sets of keypoints even over the same image. For example, the Gaussian kernel function of DoG produces high response to edges and regions with color contrast; while the Hessian matrix of Hessian-affine detector penalizes the responses on edge structure but favors the blob-like structures. Similarly, MSER fires on homogeneous color regions with high contrast against its surrounding area [10].

In order to achieve accurate NDK matching, we adopt both the Hessian-affine [11], and MSER [11] as the keypoint detectors in our approach, in the spirit of [15]. Both Hessian-affine and MSER have been reported to be highly stable and repeatable over different viewing conditions [11]. Moreover, Hessian-affine and MSER can complement each other by responding to different image structures and deliver a more comprehensive keypoint presentation of images.

We exploit SIFT [9] as the keypoint descriptor here, as it was reported to deliver superior performance [10].

5.2 Fast Keypoint Matching

Previous keypoint matching schemes ([8] [9] [14]) only focus on the comparison of SIFT descriptors, which is the global histogram information of keypoint support region, but overlook the importance and distinctiveness of the local color information of the keypoint. Moreover, they solely rely on database indexing to improve the efficiency, in which the overhead of computing the index and subsequent comparison of descriptors with index are not trivial as compared to the direct similarity matching of two descriptors.

In our approach, instead of adopting any indexing technique, we exploit the local color information of keypoints to achieve fast keypoint matching as below.
Step 1. Compare the pixel gray color intensity c_i and c_j of keypoint i and j from two keyframes.
Step 2. If Eq. (4) holds

$$\parallel c_i - c_j \parallel < \frac{256}{speedup\_factor} \tag{4}$$

then we perform the 128D SIFT descriptor matching. Else terminate the current keypoint matching.
Step 3. Go to next keypoint matching.

The rational here is that the affine covariant keypoints, such as Hessian affine, MSER tend to correspond to the same pre-image for different viewpoints, namely the projections of the same 3D surface patch [11]. This ensures the high repeatability of keypoints. Moreover, the Hessian matrix of Hessian affine and local extremal detection of MSER make them fire on blob-like and homogenous color regions, in which the true corresponding keypoints tend to share similar local color. Eq. 4 effectively filters out the keypoints whose color difference is above the threshold from subsequent expensive SIFT matching. The speedup is approximately proportional to *speedup_factor*. For example, if the *speedup_factor* is set to 4 and assume the color of keypoints follows uniform distribution, we can approximately filter out 75% of negative keypoints by performing a scalar value matching, which is fairly trivial as compared to the 128D SIFT key matching. Note that when the *speedup_factor* is high, there exists a tradeoff between the speedup and illumination robustness.

5.3 Keypoint Match Re-Checking based on Spatial and Angular Consistency

The keypoint matching criteria here adopts the distance ratio between the first and second nearest neighbor (NN) [9]. Two keypoints are regarded as a match, if

$$\frac{\parallel D_i - D_{1stNN} \parallel}{\parallel D_i - D_{2ndNN} \parallel} < l \tag{5}$$

where D_i, D_{1stNN} and D_{2ndNN} are the descriptor of keypoint i, its first and second nearest neighbor in the other image and l is the threshold.

The matching keypoints are re-checked by a confidence mechanism in [18], which examines the similarity of spatial and angular distribution of the keypoint's 8 nearest neighbors. The rational here is that if two matching keypoints are true positive, they must have similar spatial and angular relations with their neighboring matching keypoints.

If the cardinality of the matched keypoints from keyframes is above the gating threshold γ, these two keyframes are deemed be a pair of NDKs.

6. NDK TRANSITIVITY PROPAGATION

After performing accurate image matching in each small group, we link the detected NDKs from different groups by utilizing the shared keyframes in these groups through transitivity propagation. A keyframe could belong to more than one group in the following scenarios: (1) the adjacent temporal partitions have overlapping keyframes of v days (2) the semantic partition is a soft partition, where a keyframe can be classified into multiple genre groups and (3) in the step of visual clustering, each keyframe is represented by 6 feature vectors, which are possible to fall into different clusters. The NDK transitivity propagation complements the proposed partitioning model by linking NDKs from different partitions. Moreover, we postprocess the detected NDK groups within the same temporal and semantic partition by matching their respective keyframes that have largest similarity of semi-global features. This is to address the issue that the visual clustering might split the true NDK groups.

7. EXPERIMENTS AND DISCUSSIONS

7.1 Testing Dataset and Evaluation Criteria

The testing dataset is the TRECV 06 news video corpus [16], which covers 259 news videos from CNN, MSNBC, LBC, HURRA, CCTV and NTDTV over 59 days. The number of live scene keyframes amounts to approximately 62k. From the ground truth annotated by student helpers , 10632 keyframes are found to form 57647 pairs and 3041 groups of NDKs. The size of NDK groups ranges from 2 to 101, and the time span of NDK groups ranges from 1 day to 35 days, which is the news about Sadam trial.

We evaluate the proposed NDK detection approach in 3 aspects: speed, recall of corpus partitioning and accuracy of NDK detection. The evaluation criteria for corpus partitioning are measured by the percentage of NDKs pairs that fall into the same partition or can be linked by shared NDKs in different partitions. The evaluation criteria for the whole NDK detection system are Precision, Recall and F1 of detected NDK pairs.

7.2 Speed Efficiency Analysis

In the experiments, we set the parameter $speedup\_factor$ for fast keypoint matching in Eq. (4) to 8 for DoG and 16

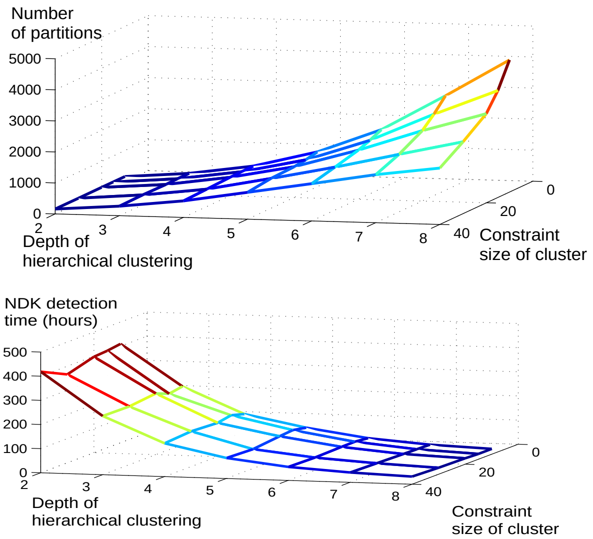

Figure 4: Number of partitions and NDK detection time over different values of α and $s_{constraint}$

for MSER. With this speedup, the time t_{SIFT} of matching 2 keyframes with around 500 keypoints is reduced from 0.35s to 0.046s approximately on a P4 3.2G PC with 1G memory.

The time of NDK detection of our approach depends on the distribution of keyframes in each partition. The time required to complete the NDK detection is:

$$T = \sum_{i=1}^{\chi} 1/2 \times n_i \times (n_i - 1) \times t_{SIFT} \qquad (6)$$

where t_{SIFT} is the time required in each SIFT image matching, n_i is the number of keyframes in partition i and χ is the total number of partitions. χ is equal to $M \times G \times K$, where M and G is the number of temporal and semantic partitions respectively and K is the average number of visual clusters for each temporal and semantic partition.

The brute force and temporal partition [14] approaches require approximately $62k \times (62k-1)/2 \times t_{SIFT} \doteq 1023$ days and $1023/M$ days respectively to complete the NDK detection. When the sizes of most groups are approximately equally small, the proposed approach can achieve approximately χ times of speedup over the brute force approach and $\chi/M = G \times K$ times of speedup over [14]. Given M and G, χ depends on the depth of the hierarchical clustering α and constraint size of clusters $s_{constraint}$. Figure 4 illustrates how χ and T change over different values of clustering depth α and $s_{constraint}$. As α increases and $s_{constraint}$ decreases, the number of partitions χ increases from 200 to 4255, and accordingly the time required for NDK detection decreases drastically from 410 hours to 25 hours.

7.3 Recall of Corpus Partitioning

The recall of corpus partition indicates how many NDKs can be linked together for further SIFT-based image matching. This is actually the upper bound recall of the proposed system, as the NDK detection results are based on the image matching in the partitions.

In the temporal partition, we set the partition duration $d = 8$ days and overlapping days $v = 2$, which leads to $M = 10$. This relatively large span of temporal partitions ensures a good recall of NDKs in temporal partitions. The semantic partition divides each temporal partition into $G = 5$ genre groups. Together, the temporal and semantic partitions divide the corpus into $G \times M = 50$ groups. These 50 groups give a 96.7% recall or completeness of NDK pairs. In the visual clustering phase, we set the number of clusters for each k-means to 2 and generate different number of partitions χ with different values of α and $s_{constraint}$.

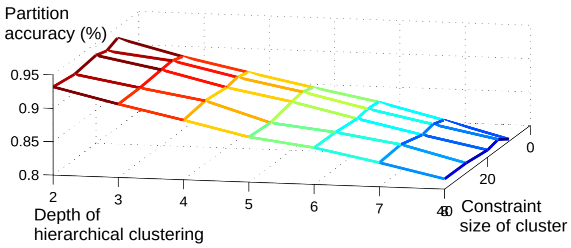

Figure 5: Accuracy (recall) of corpus partitioning over different values of α and $s_{constraint}$

Figure 5 illustrates the corpus partition accuracies over different values of α and $s_{constraint}$. As shown, the recall decreases as α increases and $s_{constraint}$ decreases. This is because more visual clusters are generated with larger α and smaller $s_{constraint}$. The best recall reaches 93.35% with $\chi = 200$, $\alpha = 2$ and $s_{constraint} = 10$. The time T required for NDK detection in this case amounts to 410 hours. The worst recall is 80.33% with $\chi = 4255$, $\alpha = 8$ and $s_{constraint} = 10$, and T is only 25 hours.

Obviously there exists a dilemma between the time required for NDK detection and the accuracy of corpus partition, as the more visual clusters generated means lower NDK detection complexity but higher probability to split true NDK groups into different partitions. However, one interesting observation is that unlike the NDK detection time, the corpus partition accuracy does not drop exponentially as α increases. This is due to the constraint size of cluster $s_{constraint}$ in the asymmetric hierarchical k-means clustering. As true NDKs tend to form coherent small-size clusters in the feature space, $s_{constraint}$ prevents the division of these useful clusters, but tends to further split the large-size clusters with many outlier non-NDKs.

7.4 Accuracy of NDK Detection

As the NDK detection is extremely time consuming, we only perform it on a number of representative runs with different number of partitions (from 200 to 4255) and run time required (from 410 to 25 hours). Table 1 illustrates the precision, recall and F1 of 5 selected runs. Specially, *run* 5 gives the fastest speed of 25 hours with a satisfactory accuracy of 73.38%. Due to the infeasible computational time of brute force approach (around 1023 days) and the approach of [14] (1023/M days), we have no means to compare accuracies of our approach with them. As shown in Table 1, when the number of partitions χ goes from 200 to 4255, the resulting accuracies in terms of F1 are fairly stable and satisfactory. This is because (1) the NDKs that are not clustered or linked together by the semi-global image features are visually too disparate to be detected by SIFT-based image matching either; and (2) as the number

Table 1: Accuracies of NDK detection

run	1	2	3	4	5
χ	200	836	1438	2616	4255
α	2	5	6	7	8
$s_{constraint}$	10	10	30	15	10
Time(hours)	410	142	99	38	25
Prec(%)	75.64	76.01	76.79	78.18	82.45
Recall(%)	77.33	75.47	74.31	70.69	66.12
F1(%)	76.47	75.74	75.53	74.24	73.38

of partitions grows, the recall might be lowered, but the precision will be increased as most of the non-NDKs are excluded from the image matching in each partition. Moreover, due to the high NDK recall of temporal and semantic partitions, we thereby speculate that the accuracies might not have very large fluctuation as χ grows from 1 (the brute force approach) to M (temporal partition in [14] approach) to $M \times G$ (temporal & semantic partition) and eventually to $M \times G \times K$ (all partitions). In fact, when χ grows within a reasonable range, both precision and recall are expected to be improved as more outlier non-NDKs will be excluded from the SIFT-based image matching.

8. CONCLUSION AND FUTURE WORK

We proposed a fast and robust system to detect NDKs in large scale news video corpus. The system achieved high efficiency by dividing the corpus into small partitions on temporal, semantic and visual dimensions by exploiting domain knowledge and corpus characteristics. The significance of such partition is twofold. First, it drastically reduces the NDK search complexity, and therefore, enables the system to rapidly detect NDKs in large-scale corpus. Second, it helps to improve the NDK detection accuracy by excluding most of the outlier non-NDKs out of the image matching process. In order to further improve the speed efficiency, a fast keypoint matching scheme was proposed to match the local color information of keypoints before matching the SIFT descriptors. The experimental results on TRECV06 corpus with 62k keyframes show that our system can delivers satisfactory NDK detection accuracy and start-of-art speed efficiency without utilizing any indexing technique.

Several open issues do exist. First, we do not exploit any indexing techniques, which can definitely improve the image matching speed further. Second, the proposed semi-global image features address the scale changes among NDKs in a heuristic manner. Some scale invariance methods can be explored to tackle this problem.

9. REFERENCES

[1] S.-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D.-Q. Zhang. Columbia university trecvid-2005 video search and high-level feature extraction. In *TREC Video Retrieval Evaluation Proceedings*, March 2006.

[2] T.-S. Chua, S.-Y. Neo, Y.-T Zheng, H.-K. Goh, Y. Xiao, S. Tang, and M. Zhao. Trecvid-2006 by nus-i2r. In *TREC Video Retrieval Evaluation Proceedings*, March 2006.

[3] P. Duygulu, J.-Y. Pan, and D. A. Forsyth. Towards auto-documentary: Tracking the evolution of news stories. In *Proceedings of the ACM Multimedia Conference*, pages 820–827, 2004.

[4] W. Hsu and S.-F. Chang. Topic tracking across broadcast news videos with visual duplicates and semantic concepts. In *Proceedings of International Conference on Image Processing*, Atlanta, USA, 2006.

[5] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[6] I. T. Joliffe. *Principal Component Analysis.* Springer-Verlag, 1986.

[7] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of Conference on Computer Vision and Pattern Recognition 2004*, volume 2, pages II–506–II–513 Vol.2, 2004.

[8] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *Proceedings of ACM International Conference on Multimedia*, pages 869–876, New York City, USA, October 2004.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[12] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94(1-3):3–27, 2004.

[13] S.-Y. Neo, J. Zhao, M.-Y. Kan, and T.-S. Chua. Video retrieval using high level features: Exploiting query matching and confidence-based weighting. In *Proceedings of ACM International Conference on Image and Video Retrieval*, pages 143–152, 2006.

[14] C.-W. Ngo, W.-L. Zhao, and Y.-G. Jiang. Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. In *Proceedings of ACM International Conference on Multimedia*, pages 845–854, Santa Barbara, USA, 2006.

[15] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1470, 2003.

[16] TRECV. TREC Video retrieval evaluation. http://www.nlpir.nist.gov/projects/trecvid.

[17] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *Proceedings of ACM International Conference on Multimedia*, pages 877–884, New York City, USA, October 2004.

[18] Y.-T. Zheng, S.-Y. Neo, T.-S. Chua, and Q. Tian. Fast near-duplicate keyframes identification in large-scale corpus for video search. In *Proceedings of International Workshop on Advanced Image Processing (IWAIT)*, Bangkok, Thailand, 2007.