Duplicate Video Detection - Comparison of Proposed Distance Function with Dynamic Time Warping and Dependence of Detection Accuracy on Keyframe Selection

Anindya Sarkar[1], Vishwakarma Singh[2], Pratim Ghosh[1], B. S. Manjunath[1], Ambuj Singh[2]

[1] Department of Electrical and Computer Engineering, University of California, Santa Barbara
[2] Department of Computer Science, University of California, Santa Barbara

1 Problem Statement

The problem we are considering here is duplicate video detection. We have a database of N videos and we store compact signatures, called fingerprints, for each of them. When a query video is presented, the system first returns the top-K most closely matched videos. Then, a more detailed search is performed among the top-K retrieved model videos to obtain the best match. Finally, a separate module is used to confirm whether the best matched video is indeed a duplicate. A complete overview of our duplicate detection framework is shown in Fig. 1. In this write-up, we focus on two aspects:

1. comparing our proposed distance measure (1) with the dynamic time warping (DTW) [2] distance measure for duplicate detection,

2. studying the importance of keyframe selection for the duplicate detection task.

The database videos are referred to as "model" videos in this write-up.

The N model video signatures in the database are denoted by $\{X^i\}_{i=1}^N$. On presenting a query video signature Q, the aim is to find the K model video signatures that are nearest to Q. The notion of similarity is with reference to a distance measure $d(X^i, Q)$ (1). To simplify matters and improve runtime, a vector quantizer (VQ) based approach is used, where the video signatures are VQ encoded and lookup table based methods are used to make the search faster.

$$d(X^i, Q) \quad = \quad \sum_{k=1}^{M} \left\{ \min_{1 \le j \le F_i} \left\| X_j^i - Q_k \right\|_1 \right\} \tag{1}$$

where $\|X_j^i - Q_k\|_1$ refers to the L_1 distance between X_j^i, the j^{th} feature vector of X^i and Q_k, the k^{th} feature vector of Q. For every vector in Q, the best match is obtained out of all the vectors in X^i and $d(X^i, Q)$ is the summation of the best matched distances.

Glossary of Notations

1. N : number of database videos
2. V_i : i^{th} model video in the dataset
3. V_{i^*} : best matched model video for a given query
4. p : dimension of the feature vector computed per video frame
5. $Z^i \in \mathbb{R}^{T_i \times p}$: feature vector matrix of V_i, where V_i has T_i frames after temporal sub-sampling
6. $X^i \in \mathbb{R}^{F_i \times p}$: fingerprint of V_i, which has F_i keyframes
7. X_j^i: j^{th} vector of video fingerprint X^i
8. U : size of the vector quantizer (VQ) codebook used to encode the model video and query video signatures
9. $Q_{orig} \in \mathbb{R}^{T_Q \times p}$: query signature created after sub-sampling, where T_Q refers to the number of sub-sampled query frames
10. $Q \in \mathbb{R}^{M \times p}$: keyframe based signature of the query video, where M is the number of query keyframes
11. C_i : the i^{th} VQ codevector
12. $\overrightarrow{x_i}$: VQ based signature of V_i
13. $\vec{q}$: VQ based query signature
14. $\mathcal{S}_{X_j^i}$: VQ symbol index to which X_j^i is mapped
15. $\mathbb{D} \in \mathbb{R}^{U \times U}$: Inter VQ-codevector distance matrix
16. ℓ: fractional query length = (number of query frames/number of frames for the actual source video)
17. K: the number of nearest neighbors (NN) returned by the first pass (coarse search) using VQ-based signatures, where 1-NN for a certain video refers to the video itself
18. $|E|$: the cardinality of the set E

We present a short description of the signature creation process. After sub-sampling the i^{th} model video and the query video, we end up with the feature matrices $Z^i \in \mathbb{R}^{T_i \times p}$ and $Q^{orig} \in \mathbb{R}^{T_Q \times p}$, respectively. Say, we want to use only 5% of these frames to create the signatures, i.e. $F_i = T_i.(5/100)$ and $M = T_Q.(5/100)$.

The keyframe selection has been experimented with in the following ways:

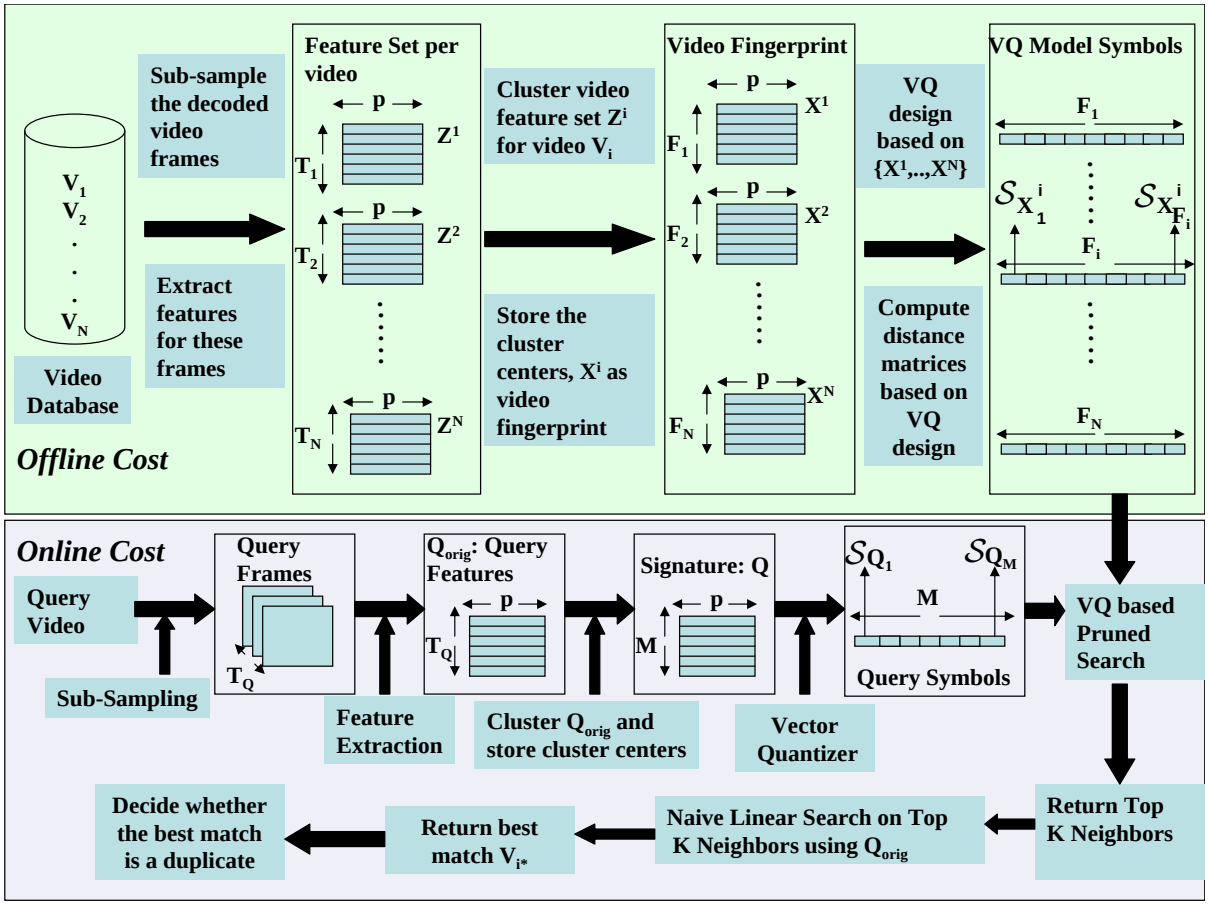


Figure 1: Block diagram of the proposed duplicate detection framework.

- performing k-means clustering on the feature vectors obtained after temporal sub-sampling (Z^i) and then choosing the cluster centers (X^i) as the keyframe vectors,
- choosing the keyframes by uniform sampling and
- choosing the keyframes randomly.

When we perform k-means based clustering on Z^i (or Q_{orig}), the cluster centers X^i (or Q) may not correspond to features actually present in the signature. To simplify matters, we map the cluster centers obtained after k-means on Z^i (or Q_{orig}) to the best matching vector in the original matrix. This is because we want a feature vector in X^i (or Q) to correspond to a keyframe in Z^i (or Q_{orig}) - this can happen only when a feature vector in X^i (or Q) is actually also present in Z^i (or Q_{orig}).

2 Use of VQ-encoded signatures

We develop an algorithm that uses VQ-based encoding on the signature feature vectors. Thus, the distance between any two feature vectors reduces to an inter-symbol distance, after VQ-based encoding. By using a lookup table of inter-VQ codevector distances, the L_1 distance computation cost (e.g. $\|X_j^i - Q_k\|_1$) can be avoided.

Using the features extracted from the database video frames, a vector quantizer of codebook size U is constructed. Since each vector in a video signature can be mapped to one of U codevectors, the effective video signature can be thought of as a U-dimensional vector, where the i^{th} dimension denotes the fraction of vectors in the original signature which get mapped to the i^{th} codevector C_i.

We normalize all the VQ-based signatures. Let $[q_1, q_2, \cdots, q_U]$ denote the normalized query video signature $\overrightarrow{q}$ and $[x_{i,1}, x_{i,2}, \cdots, x_{i,U}]$ denote the normalized model video signature $\overrightarrow{x_i}$ for the i^{th} video V_i.

$$q_k = |\{j : \mathcal{S}_{Q_j} = k,\ 1 \leq j \leq M\}|/M \tag{2}$$

$$x_{i,k} = |\{j : \mathcal{S}_{X_j^i} = k,\ 1 \leq j \leq F_i\}|/F_i \tag{3}$$

Generally, there is a high degree of redundancy among video frames; hence, many of them will get mapped to the same VQ codevector and there will be many VQ codevectors which will have no representative (assuming a large enough U). Let $\{t_1, t_2, \cdots, t_{N_q}\}$ and $\{n_{i,1}, n_{i,2}, \cdots, n_{i,N_{x_i}}\}$ denote the non-zero dimensions in $\overrightarrow{q}$ and $\overrightarrow{x_i}$, respectively.

The distance between them can be expressed as:

$$d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) = \sum_{k=1}^{N_q} q_{t_k} \times \left\{ \min_{1 \leq j \leq N_{x_i}} \mathbb{D}(t_k, n_{i,j}) \right\} \tag{4}$$

$$\text{where } \mathbb{D}(i,j) = \|C_i - C_j\|_1,\ 1 \leq i, j \leq U \tag{5}$$

where $\mathbb{D} \in \mathbb{R}^{U \times U}$ is the inter-VQ codevector distance matrix.

It can be easily shown that the distances in (1) and (4) are identical, apart from a constant scaling factor, when each vector in (1) is represented by its corresponding VQ codevector.

$$d(X^i, Y) = M \times d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) \tag{6}$$

3 An Introduction into Dynamic Time Warping

We present a brief introduction to dynamic time warping (DTW), based on material present in [2]. This is a self-contained section and the notations used in this section are not to be confused with the notations in the glossary. Let us consider two segments $\mathcal{X}$ and $\mathcal{Y}$, of length T_x and T_y, respectively. They are represented by the feature-vector sequences $(x_1, x_2, \ldots, x_{T_x})$ and $(y_1, y_2, \ldots, y_{T_y})$ where x_i and y_j are the i^{th} and j^{th} feature vectors of $\mathcal{X}$ and $\mathcal{Y}$, respectively.

If only linear time alignment (LTA) is used, the dissimilarity between $\mathcal{X}$ and $\mathcal{Y}$ is defined as:

$$D_{LTA}(\mathcal{X}, \mathcal{Y}) = \sum_{i_x=1}^{T_x} \hat{d}(i_x, i_y)$$
$$\text{where } i_y = \frac{T_y}{T_x} i_x$$

and i_x and i_y denote the time indices of $\mathcal{X}$ and $\mathcal{Y}$, respectively. $\hat{d}(i_x, i_y)$ denotes the distance (according to some distance metric) between the feature vectors corresponding to the i_x^{th} frame of $\mathcal{X}$ and the i_y^{th} frame of $\mathcal{Y}$.

A more general time alignment and normalization scheme involves the use of two warping functions, ϕ_x and ϕ_y, which map the i_x and i_y axes (corresponding to $\mathcal{X}$ and $\mathcal{Y}$, respectively) to a common time axis, denoted by k, where both the sequences are warped to a sequence of length T.

$$i_x = \phi_x(k), \ k = 1, 2, \ldots, T$$
$$i_y = \phi_y(k), \ k = 1, 2, \ldots, T$$

Based on the choice of the warping paths $\phi_x(k)$ and $\phi_y(k)$ and the choice of a weighting sequence $m(k)$ which weights the various paths, the path normalized distance $d_\phi(\mathcal{X}, \mathcal{Y})$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ can now be defined as follows, where ϕ is a set consisting of the time-warping functions, i.e. $\phi = (\phi_x, \phi_y)$:

$$D_\phi(T_x, T_y) = \sum_{k=1}^{T} \hat{d}(\phi_x(k), \phi_y(k)) m(k)$$
$$\hat{d}_\phi(\mathcal{X}, \mathcal{Y}) = D(T_x, T_y)/M_\phi$$

$D_\phi(T_x, T_y)$ is the accumulated distance while traveling from $(1, 1)$ to (T_x, T_y) in the (i_x, i_y) grid along the path given by the warping functions $\{\phi_x(k), \phi_y(k)\}$, $d(\phi_x(k), \phi_y(k))$ is the distance between the feature vectors corresponding to the $\phi_x(k)$ and $\phi_y(k)$ - numbered frames of $\mathcal{X}$ and $\mathcal{Y}$, respectively and M_ϕ is a path normalization term, which makes d_ϕ independent of the path lengths.

In [2], various sets of allowable moves are mentioned - of them, *we have used Type I moves (Fig. 2) in our experiments because it allows the warping path to take all possible paths (the only constraint being that it will not go back along time) in the forward direction*. Type II is provided for comparison (Fig. 2).

Based on the allowed moves, there can be a large number of paths from $(1, 1)$ to (T_x, T_y). A dynamic programming (DP) [1] based approach is used to find the best possible path out of all the available moves. The optimal path will return the lowest accumulated distortion while moving from $(1, 1)$ to (T_x, T_y) along

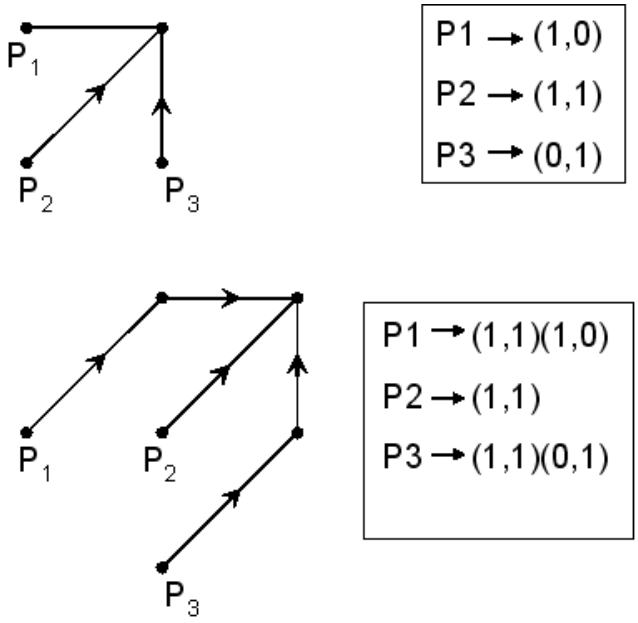


Figure 2: Allowable path specifications - Types I (upper) and II (lower)

any set of allowed paths.

$$D(T_x, T_y) = \min_{\phi} D_{\phi}(T_x, T_y)$$

$$= \min_{\phi_x, \phi_y} \sum_{k=1}^{T} \hat{d}(\phi_x(k), \phi_y(k)) m(k)$$

$$d_{DTW}(T_x, T_y) = D(T_x, T_y)/M_{\phi}$$

where $d_{DTW}(T_x, T_y)$ is the normalized distance along the optimal path between $\mathcal{X}$ and $\mathcal{Y}$, $\phi_x(T) = T_x$ and $\phi_y(T) = T_y$.

The constraints on the warping functions ϕ_x and ϕ_y are as follows:

1. **Endpoint constraints:** The warping functions should ensure that the optimal path starts from $(1, 1)$ and terminates at (T_x, T_y) in the (i_x, i_y) plane.

$$\text{beginning point: } \phi_x(1) = 1,\ \phi_y(1) = 1$$
$$\text{ending point: } \phi_x(T) = T_x,\ \phi_y(T) = T_y$$

2. **Monotonicity conditions:** In order to maintain the temporal continuity, the warping functions should be monotonically increasing:

$$\phi_x(k+1) \geq \phi_x(k) \quad \phi_y(k+1) \geq \phi_y(k)$$

3. **Local continuity constraints:** Normally, since we do not expect a huge change in the feature vector from one frame to the next, the path should not change drastically between two points in the (i_x, i_y) plane.

   Based on the list of allowed paths, there are constraints imposed on $|\phi_x(k+1) - \phi_x(k)|$ and $|\phi_y(k+1) - \phi_y(k)|$. For Type I paths, $(\phi_x(k+1) - \phi_x(k)) \leq 1$ while for Type II paths, $(\phi_x(k+1) - \phi_x(k)) \leq 2$ (the same applies for ϕ_y terms), as shown in Fig. 2 .

   Let the optimum path $\mathcal{P}$ be a sequence of moves, each denoted by the corresponding displacements along the i_x and i_y axes, respectively. Thus, $\mathcal{P} \rightarrow \{(p_1, q_1), (p_2, q_2), \ldots, (p_T, q_T)\}$, implies a displacement of p_i along i_x and q_i along i_y axes, respectively, for the k^{th} time instant, where $1 \leq k \leq T$. For Type I paths, the possible values of (p_i, q_i) are (1,0), (1,1) and (0,1).

   The constraints on the p_i and q_i terms are that the endpoint has to be reached at time $k = T$, which implies

$$\sum_{k=1}^{T} p_k = T_x \quad \sum_{k=1}^{T} q_k = T_y$$

4. **Global path constraints:**

   Based on the allowable paths for p_i and q_i terms, there are certain regions in the (i_x, i_y) plane which cannot be reached by the optimal warping path.

   If Q_{max} is defined as the maximum possible expansion in time warping, where

$$Q_{max} = \max_{\ell} \left[\sum_{i=1}^{T_\ell} p_i^{(\ell)} / \sum_{i=1}^{T_\ell} q_i^{(\ell)} \right]$$

   T_ℓ is the total number of moves in P_ℓ (as in Fig. 2) while ℓ is the index of the allowable path (=1,2 and 3 for Types I and II). Thus, T_1, T_2 and T_3 equal 1,1,1 and 2,1,2 for Types I and II, respectively. Q_{max} equals ∞ and 2, respectively, for Types I and II.

   No time warping can be accomplished if $(T_x - 1) > Q_{max}(T_y - 1)$ or $(T_y - 1) > Q_{max}(T_x - 1)$. For most of the other local continuity based paths, Q_{max} equals 2 - therefore, if the length of the longer segment is more than twice the length of the shorter sequence, their distance cannot be computed in the time-warping framework. To enable the matching path between two sequences be as generic as possible (all possible moves in the forward direction are possible provided that the displacement is at most 1 along each axis for every move), we choose the allowable paths (Type I instead of Type II) that ensured a $Q_{max} = \infty$, thus ensuring that any two video sequences can be compared (without any constraints on their lengths).

5. **DTW computation**

   For DTW computation, we need to fix a set of allowable moves (we use Type I moves) that decide the allowable $\phi_x(k)$ and $\phi_y(k)$ paths from a given point. Secondly, we decide upon the slope weighting that fixes $m(k)$ and the normalization factor M_ϕ.

We use Type (d) slope weighting [2]:

$$m(k) = \phi_x(k) - \phi_x(k-1) + \phi_y(k) - \phi_y(k-1)$$
$$M_\phi = \sum_{k=1}^{T} m(k) = T_x + T_y$$

assuming that the two sequences $\mathcal{X}$ and $\mathcal{Y}$ are of length T_x and T_y, respectively.

Like other DP-based optimal path finding methods, the DTW algorithm has three parts:
1. Initialization:

$$D(1,1) = \hat{d}(1,1)m(1)$$

2. Recursion:
For all points (i_x, i_y) which can be reached by the optimal warping path

$$D(i_x, i_y) = \min_{i'_x, i'_y}[D(i'_x, i'_y) + \gamma((i'_x, i'_y), (i_x, i_y))]$$
where $\gamma((i'_x, i'_y), (i_x, i_y))$ is defined as
$$\gamma((i'_x, i'_y), (i_x, i_y)) = \sum_{l=0}^{L_s} \hat{d}(\phi_x(T'-l), \phi_y(T'-l))m(T'-l)$$

where $D(i_x, i_y)$ is the accumulated distortion along the optimal DTW path from $(1,1)$ to (i_x, i_y), L_s is the number of possible moves from (i'_x, i'_y) to (i_x, i_y) based on ϕ_x and ϕ_y, $\phi_x(T' - L_s) = i'_x$ and $\phi_y(T' - L_s) = i'_y$.

For the specific case of Type I paths, the recursion equations are as follows:

$$D(i_x, i_y) = min \left\{ \begin{array}{l} D(i_x - 1, i_y) + \hat{d}(i_x, i_y) \\ D(i_x - 1, i_y - 1) + 2\hat{d}(i_x, i_y) \\ D(i_x, i_y - 1) + \hat{d}(i_x, i_y) \end{array} \right\} \quad (7)$$

3. Termination:

$$d_{DTW}(\mathcal{X}, \mathcal{Y}) = D(T_x, T_y)/M_\phi = D(T_x, T_y)/(T_x + T_y), \text{ for Type (d) slope weighting} \quad (8)$$

4 Computing DTW between Video Sequences with VQ-based Signatures

As denoted in the glossary, Q_{orig} refers to the query signature obtained after sub-sampling the query frames (before keyframe extraction). Let us denote Q_{orig} by Q^{orig} for ease of expression. Let the compact query

signature after keyframe extraction be represented by M frames $\{Q^{orig}_{a_1}, Q^{orig}_{a_2}, \cdots, Q^{orig}_{a_M}\}$ and the corresponding VQ indices are $\{\mathcal{S}_{Q^{orig}_{a_1}}, \mathcal{S}_{Q^{orig}_{a_2}}, \cdots, \mathcal{S}_{Q^{orig}_{a_M}}\}$, where $a_1 < a_2 \cdots < a_M$. We denote the effective VQ-based query signature computed using the keyframes (and with time-sequence maintained) as $\overrightarrow{e} = [e_1, e_2, \cdots, e_M]$ where

$$\textbf{time sequential query signature } e_k = \mathcal{S}_{Q^{orig}_{a_k}}, \text{ where } Q^{orig}_{a_k} = Q_k,\ a_1 < a_2 \cdots < a_M,\ 1 \leq k \leq M \quad (9)$$

For video V_i, let the model signature be represented by F_i frames $\{Z^i_{b_1}, Z^i_{b_2}, \cdots, Z^i_{b_{F_i}}\}$ and the corresponding VQ indices are $\{\mathcal{S}_{Z^i_{b_1}}, \mathcal{S}_{Z^i_{b_2}}, \cdots, \mathcal{S}_{Z^i_{b_{F_i}}}\}$, where $b_1 < b_2 \cdots < b_{F_i}$. Denoting the effective VQ-based model signature of V_i computed using the keyframes (and with time-sequence maintained) as $\overrightarrow{f^i} = [f^i_1, f^i_2, \cdots, f^i_{F_i}]$ where

$$\textbf{time sequential } i^{th} \textbf{ model signature } f^i_k = \mathcal{S}_{Z^i_{b_k}}, \text{ where } Z^i_{b_k} = X^i_k,\ b_1 < b_2 \cdots < b_{F_i},\ 1 \leq k \leq F_i \quad (10)$$

The normalized query and model histogram based signatures $\overrightarrow{q}$ (2) and $\overrightarrow{x_i}$ (3) can be computed from $\overrightarrow{e}$ (9) and $\overrightarrow{f^i}$ (10), respectively.

$$q_k = |\{j : e_j = k, 1 \leq j \leq M\}|/M$$
$$x_{i,k} = |\{j : f^i_j = k, 1 \leq j \leq F_i\}]/F_i$$

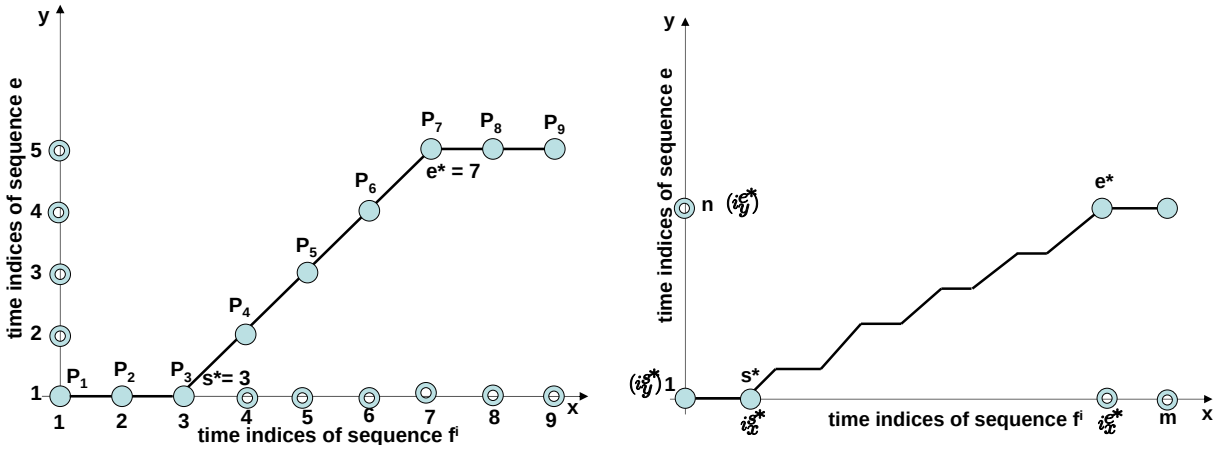
For the DTW distance, the sequence information is used. A thing to note here is that though the query video is obtained as a time window extracted from the model video, the query signature is not computed using all the frames that were originally in the query. We use a certain fraction of keyframes (5%) from the query and model videos to construct the video signatures - *therefore, all query keyframes may not match with model keyframes, that are consecutive along time, along the DTW path*. Even if all the query keyframes do match, there may be some model keyframes along the DTW path which do not match well with the query keyframes, which can increase the effective DTW distance. *To overcome this problem, we only consider the DTW distance between the query and the best matching part of the model signature*. We refer to the total DTW distance as d^1_{DTW} and the distance between the query and the matching model part as d^2_{DTW}.

The DTW path is computed as follows, using (7),

$$D(f^i_{(1:m)}, e_{(1:n)}) = min \left\{ \begin{array}{l} D(f^i_{(1:m-1)}, e_{(1:n)}) + \hat{d}(f^i_m, e_n) \\ D(f^i_{(1:m-1)}, e_{(1:n-1)}) + 2\hat{d}(f^i_m, e_n) \\ D(f^i_{(1:m)}, e_{(1:n-1)}) + \hat{d}(f^i_m, e_n) \end{array} \right\} \quad (11)$$

where

$$f^i{}_{(1:m)} = [f^i_1, f^i_2, \cdots, f^i_m],\ e_{(1:n)} = [e_1, e_2, \cdots, e_n], \text{ and}$$
$$\hat{d}(f^i_m, e_n) = \mathbb{D}(f^i_m, e_n) = \|\mathcal{C}_{f^i_m} - \mathcal{C}_{e_n}\|_1$$



(a) (b)

Figure 3: We assume here that the query is a subset of the i^{th} model video V_i. (a) We obtain the DTW path between the i^{th} model and query signatures; a point on the grid $(1, 3)$ denotes the distance between the VQ codevectors corresponding to the symbols f_1^i and e_3, i.e. $\mathbb{D}(f_1^i, e_3)$. The path needed to compute d^2_{DTW} consist of the points $\{P_3, P_4, \cdots, P_7\}$ where s^* (starting point index among all points in the total DTW path) and e^* (ending point index) are equal to 3 and 7, respectively. (b) We show a practical DTW path when though the query video is a subset of the model video, the matching path is not fully diagonal because both the query and model signatures **consist of keyframes which are not necessarily consecutive along time**.

While finding the DTW path between $\overrightarrow{f^i}$ and $\overrightarrow{e}$, let there be M_ϕ points along the path: $(i_x^1, i_y^1), (i_x^2, i_y^2), \cdots, (i_x^{M_\phi}, i_y^{M_\phi})$ are the M_ϕ point pairs. We refer to the warping path between the query and the best matching model part as "matching path".

$$\textbf{starting point in the matching path } s^* = \arg\max_k i_x^k, \text{ where } i_y^k = 1 \tag{12}$$

$$\textbf{ending point in the matching path } e^* = \arg\min_k i_x^k, \text{ where } i_y^k = M \tag{13}$$

$$d^1_{DTW}(\overrightarrow{f^i}, \overrightarrow{e}) = \frac{D(F_i, M)}{F_i + M} \tag{14}$$

$$d^2_{DTW}(\overrightarrow{f^i}, \overrightarrow{e}) = \frac{D(i_x^{e^*}, i_y^{e^*}) - D(i_x^{s^*}, i_y^{s^*})}{i_x^{e^*} - i_x^{s^*} + i_y^{e^*} - i_y^{s^*}} \tag{15}$$

where $D(\cdot, \cdot)$ is computed using (11).

In Fig. 3(a), we show the DTW path between two signatures: $\overrightarrow{f^i}$ and $\overrightarrow{e}$, along the x and y-axes, respectively. It is seen that $\overrightarrow{e}$ is well-matched to a subset of $\overrightarrow{f^i}$, and the matching part is given by points

P_3 ($s^* = 3$) to P_7 ($e^* = 7$). Thus, the d^1_{DTW} distance (total distance) between $\overrightarrow{f^i}$ and $\overrightarrow{e}$ (14) is given by the effective distance along the warping path defined by the points $\{P_1, P_2, \cdots, P_9\}$. The d^2_{DTW} distance (matching distance) between $\overrightarrow{f^i}$ and $\overrightarrow{e}$ (15) is given by the effective distance along the warping path defined by the points $\{P_3, P_4, \cdots, P_7\}$.

In Fig. 3(b), we show why the DTW distance is higher than the d_{VQ} distance even when the query video is obtained as a subset of the model video. The query signature is obtained after selecting certain frames (keyframes) from the query video and then constructing the query signature using the feature vectors corresponding to these keyframes. The reason why the DTW path is not completely diagonal (*to get a diagonal path, each and every query feature vector should be mapped to a certain model feature vector where the best matched model vectors are consecutive along time*) is that only a fraction (5%) of sub-sampled frames is used for creating the video signature. To construct the DTW-based signatures and ensure a smaller distance between the query and the corresponding model signature, we could have retained a larger number of keyframes but that would result in more storage space for the video fingerprints. Computation time wise, when the query signature (with symbols in time sequence) $\overrightarrow{e}$ and the model signature $\overrightarrow{f^i}$ consist of M and F_i symbols, respectively, both the d_{VQ} (4) and d^1_{DTW} (14) distances involve considering $M.F_i$ distance terms. Though d^2_{DTW} considers only a smaller path as compared to d^1_{DTW}, we need to compute d^1_{DTW} first which gives us the entire warping path in the $F_i \times M$ grid, based on which we compute d^2_{DTW}.

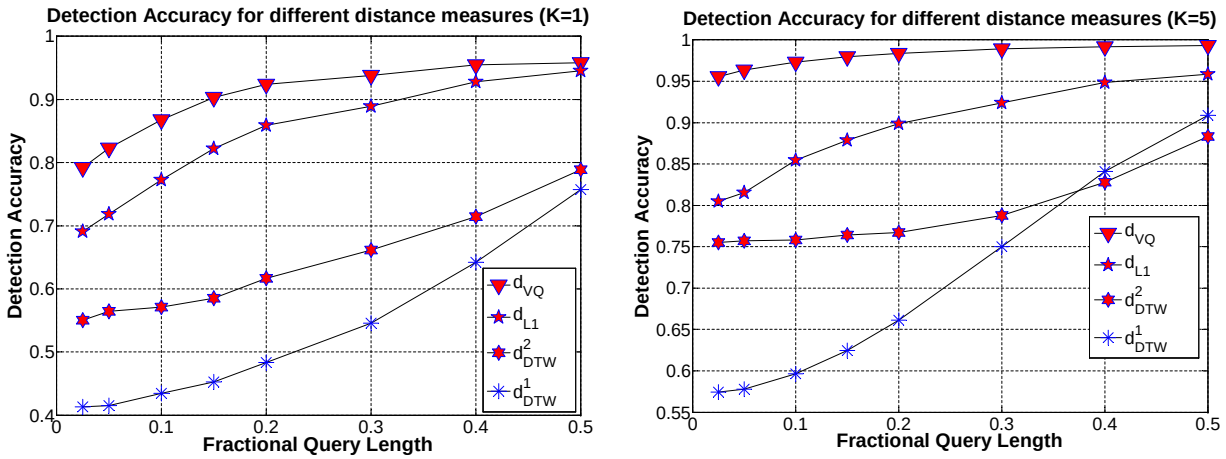
5 Experimental Results

For our experiments, we use 1200 model videos and 18 duplicates (using different image processing and noise addition methods) are created per model video. Each query video is a duplicate of *one and only one* model video. A detection error occurs when the best matching video does not correspond to the actual video from which it was created. To construct the video fingerprints, the number of keyframes is 5% of the number of frames in the feature matrix obtained after sub-sampling, i.e. $F_i = T_i.(5/100)$ (for i^{th} model video) and $M = T_Q.(5/100)$ (for query).

Since the time information is discarded in our VQ-based signatures, our distance measure will give a better match if the query is a collection of frames in the model video, which have been shuffled along time. *For fair comparison, we limit our experiments to queries which contain a certain time window of model video frames - i.e. though the query is a noisy subset of the model video, the quetry frames are not further distorted along time*. Also, we shall be considering the same set of frames to constitute the video signatures, for both the model and query videos, for both the distance measures - the frames will also be arranged along time for creating the signatures to be compared for computing the DTW distance.

While comparing different distance measures for duplicate detection, we also consider d_{L1} which is the L_1 distance between the VQ-based signatures.

$$d_{L1}(\overrightarrow{x_i}, \overrightarrow{q}) = \sum_{k=1}^{U} |x_{i,k} - q_k| \tag{16}$$



(a) (b)

Figure 4: Variation of the detection accuracy with varying query lengths and for different distance measures - the fractional query length (ℓ) is varied from 0.025 to 0.50 and it is seen that in terms of detection accuracy, $d_{VQ} > d_{L1} > d^2_{DTW} > d^1_{DTW}$. The error is averaged over all the queries.

Table 1: Computation of detection accuracy using the d_{VQ} distance for different keyframe selection schemes, for K = 1

method \ ℓ	0.025	0.05	0.10	0.15	0.20	0.30	0.40	0.50
k-means	**0.792**	**0.823**	**0.867**	**0.903**	**0.924**	**0.938**	**0.955**	**0.958**
uniform	0.783	0.804	0.844	0.879	0.900	0.919	0.941	0.944
random	0.737	0.758	0.824	0.868	0.893	0.921	0.933	0.935

In Fig. 4(a) and 4(b), we show the detection accuracy obtained using the VQ-based signatures using K = 1 and 5, in the K-NN setup. We observe that d_{VQ} (4) results in the best detection accuracy followed by d_{L1}, d^2_{DTW} and d^1_{DTW} (in order of decreasing accuracy). Thus, using the modified DTW distance (d^2_{DTW}) results in better performance than the total DTW distance (d^1_{DTW}), though both these distance functions perform worse than our proposed distance function d_{VQ}.

In Tables 1 and 2, we show the detection accuracy obtained using different methods of keyframe selection (k-means based, uniform sampling and random selection) using d_{VQ} as the distance measure. It is observed that the detection accuracy obtained using **k-means based cluster centers** for keyframe features is slightly higher than the accuracy obtained using "uniformly sampled" and "randomly selected" keyframes. *Thus, though intuitively the k-means based cluster centers should lead to properly representative signatures, the results using random keyframes seem to be nearly the same.*

Table 2: Computation of detection accuracy using the d_{VQ} distance for different keyframe selection schemes, for K = 5

method \ ℓ	0.025	0.05	0.10	0.15	0.20	0.30	0.40	0.50
k-means	**0.956**	**0.964**	**0.973**	**0.980**	**0.983**	**0.989**	**0.992**	**0.993**
uniform	0.947	0.953	0.968	0.973	0.977	0.984	0.988	0.989
random	0.922	0.927	0.956	0.971	0.977	0.984	0.985	0.987

6 Distance Threshold based Approach for Duplicate Confirmation

After finding the best matched model video V_{i^*} for a given query, the next problem is to determine if the query is indeed a duplicate of the best matched model video. For that purpose, we propose a distance threshold based approach.

The training phase to obtain the distance threshold involves finding the 1-NN and 2-NN distances for 1200 query videos, over various noise conditions and query lengths. The distance between the fingerprint of the 1-NN video V_{i^*}, X^{i^*}, and the larger query signature Q_{orig}, is computed using (1) and is normalized by the query length T_Q, so as to make the threshold independent of the query length. Thus, the effective 1-NN distance equals $\{d(X^{i^*}, Q_{orig})/T_Q\}$. Since the same 1200 videos were considered as the model videos, the 1-NN always refers to a duplicate video and the 2-NN to a non-duplicate one. The distribution of 1-NN and 2-NN distances is shown in Fig. 5(a) and (b), respectively. Ideally, the threshold δ_s should be such that all the 1-NN (or 2-NN) distances are lesser (or greater) than it. If we equally weigh the probability of false alarm P_{FA} (wrongly classifying the 2-NN retrieval as a duplicate) and of missed detection P_{MD} (failing to classify the 1-NN retrieval as a duplicate), the threshold δ_s is chosen as 230 from Fig. 5(c). The corresponding P_{FA} and P_{MD} values equal 0.07. Depending on whether the emphasis is on minimizing P_{FA} or P_{MD}, δ_s can be decreased or increased, accordingly.

For verifying the effectiveness of the distance threshold, we repeat the duplicate detection experiments *on an unseen dataset of 1700 videos ($\approx$ 75 hours of video), all of which are different from the model videos*. For each video, 18 duplicates are created using various image processing and noise addition operations. Using a threshold δ_s of 230, 3% of the videos were classified as "duplicates" - for them, the 1-NN distance is less than δ_s.

For those cases where the query-to-model distance is very close to the threshold δ_s, we use a registration-based approach which attempts to register the query keyframes with the best matching model keyframes. The registration method is computationally intensive but is more accurate in determining if the query is indeed a duplicate of the retrieved candidate.

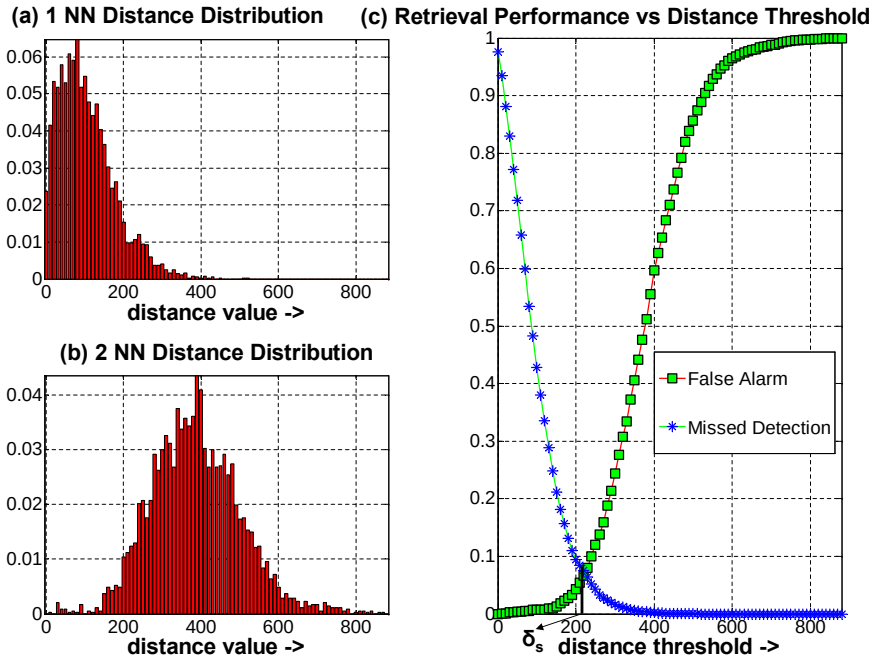


Figure 5: Choosing a distance threshold to distinguish duplicates from non-duplicates, using distance measure $d(\cdot,\cdot)$ (1), where the selected threshold δ_s is obtained such that it minimizes both P_{FA} and P_{MD}

7 Uniqueness of VQ-based Video Signatures

The uniqueness of the histogram based VQ signatures is investigated here. Due to our distance function formulation, the distance of a query video to two model videos (using (4)) will be the same if both the model videos have the same non-zero indices in the VQ-based signatures. For our database of 38000 videos, the percentage of video pairs (considering $\binom{38000}{2}$ pairs) that have exactly the same set of non-zero indices is $3.2 \times 10^{-4}\%$.

We also compute the probability that 2 videos V_1 and V_2 have the same set of non-zero indices $[I_1, I_2, \cdots, I_L]$ in their VQ-based signatures. As per notation introduced in the glossary, the kmeans-based signature for V_1 and V_2 are $X^1 \in \mathbb{R}^{F_1 \times p}$ and $X^2 \in \mathbb{R}^{F_2 \times p}$, respectively. The corresponding VQ-based signatures are $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$, respectively. We define two probability terms p_i and $p_{(I_1,\cdots,I_L)}$ in (17) and (18), respectively.

$$\begin{aligned} p_i &= \text{Prob(a video vector gets mapped to } C_i\text{, the } i^{th} \text{ VQ codevector)} \\ &= \frac{\text{number of database frames that get mapped to } C_i}{\text{number of database frames}} \end{aligned} \tag{17}$$

$$\begin{aligned} p_{(I_1,\cdots,I_L)} &= \text{Prob(a video frame gets mapped to one of } \{I_1, I_2, \cdots, I_L\}) \\ &= 1 - \text{Prob(video frame does not get mapped to one of } \{I_1, I_2, \cdots, I_L\}) \\ &= 1 - \prod_{i=1}^{L} \text{Prob(frame does not get mapped to } I_i) \\ &= 1 - (1 - p_{I_1})(1 - p_{I_2})...(1 - p_{I_L}) \end{aligned} \tag{18}$$

We assume that all frames are uniformly distributed. Of the F_1 vectors that constitute X^1, the k-means based signature for V_1, we first assume that one vector gets mapped to each of $\{I_1, I_2, \cdots, I_L\}$. The remaining $(F_1 - L)$ vectors can get mapped to any VQ index in the set $\{I_1, I_2, \cdots, I_L\}$. The probability p_{V_1} (probability that the non-zero indices in $\overrightarrow{x_1}$, the VQ-based signature for video V_1 obtained using F_1 vectors, are $\{I_1, I_2, \cdots, I_L\}$) is computed as follows:

$$\text{Prob(non-zero indices in } \overrightarrow{x_1} \text{ are } \{I_1, I_2, \cdots, I_L\}) = \binom{F_1}{L} \times (p_{I_1}.p_{I_2}...p_{I_L}) \times (p_{(I_1,\cdots,I_L)})^{F_1 - L} \tag{19}$$

where $\overrightarrow{x_1}$ is computed using the kmeans-based signature $X^1 \in \mathbb{R}^{F_1 \times p}$.

Similarly, the probability p_{V_2} (probability that the non-zero indices in $\overrightarrow{x_2}$, the VQ-based signature for video V_2 obtained using F_2 vectors, are $\{I_1, I_2, \cdots, I_L\}$) is computed as follows:

$$\text{Prob(non-zero indices in } \overrightarrow{x_2} \text{ are } \{I_1, I_2, \cdots, I_L\}) = \binom{F_2}{L} \times (p_{I_1}.p_{I_2}...p_{I_L}) \times (p_{(I_1,\cdots,I_L)})^{F_2 - L} \tag{20}$$

where $\overrightarrow{x_2}$ is computed using the kmeans-based signature $X^2 \in \mathbb{R}^{F_2 \times p}$.

Thus, we see that the probability depends on the set of indices where the video signatures match. To obtain the highest possible probability value, when two videos have L identical non-zero dimensions in the

Table 3: p_{V_1} (19) is computed for varying L, assuming $F_1 = 25$.

L	p_{V_1}	L	p_{V_1}	L	p_{V_1}	L	p_{V_1}	L	p_{V_1}
1	2.65×10^{-9}	2	2.05×10^{-5}	3	1.90×10^{-3}	4	2.84×10^{-2}	5	1.08×10^{-1}
6	2.07×10^{-1}	7	2.31×10^{-1}	8	1.47×10^{-1}	9	6.62×10^{-2}	10	2.10×10^{-2}
11	4.90×10^{-3}	12	8.92×10^{-4}	13	1.32×10^{-4}	14	1.59×10^{-5}	15	1.55×10^{-6}
16	1.26×10^{-7}	17	8.44×10^{-9}	18	4.65×10^{-10}	19	2.09×10^{-11}	20	7.52×10^{-13}

VQ-based signature, we choose the L topmost probability values out of $\{p_i\}_{i=1}^{U}$. Based on the kmeans-based signatures $\{X^i\}_{i=1}^{N}$, we observe that the average number of vectors in the model signatures, $\bar{F}$, equals 25. Also, the average number of non-zero dimensions in the VQ-based model signatures $\bar{L}$ is observed to be equal to 18. We compute p_{V_1} for $L = 1, 2, \ldots, \bar{L}$, assuming $F_1 = \bar{F}$ - as shown in Table 3. When two VQ-based signatures $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$ have the same non-zero dimensions and when $F_1 = F_2$, $p_{V_1} = p_{V_2}$, from (19) and (20). Hence, the total collision probability of two signatures $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$ having the same set of non-zero dimensions $\{I_1, I_2, \cdots, I_L\}$ equals $p_{V_1} \times p_{V_2} = (p_{V_1})^2$, assuming $F_1 = F_2$.

This analysis is incomplete in the sense that we have not computed *how likely the event, that a set of L dimensions $\{I_1, I_2, \cdots, I_L\}$ will appear as the non-zero VQ indices in the same video, is based on the VQ-based signatures of the 38000 database videos* - i.e. the probability of co-occurrence of a certain set of VQ dimensions which can be empirically computed from the database signature statistics has not been computed. After arranging the probability terms $\{p_i\}_{i=1}^{U}$ in descending order, let us assume that the re-arranged sequence is $p_{I_1^*}, p_{I_2^*}, p_{I_3^*}, \ldots, p_{I_U^*}$ where $p_{I_1^*} \geq p_{I_2^*} \geq p_{I_3^*} \geq \ldots p_{I_U^*}$. Thus, when we compute p_{V_1} assuming a set of L non-zero dimensions, we actually consider the set $\{I_1^*, I_2^*, \cdots, I_L^*\}$. We define $p^{\text{database}}_{(I_1^*, I_2^*, \cdots, I_L^*)}$ (21) which denotes the co-occurrence probability of a given set of VQ indices $(I_1^*, I_2^*, \cdots, I_L^*)$, while considering all the database video VQ-based signatures.

$$p^{\text{database}}_{(I_1^*, I_2^*, \cdots, I_L^*)} = \text{Prob(a database video has all the elements in } (I_1^*, I_2^*, \cdots, I_L^*) \text{ as non-zero indices)} \quad (21)$$
$$= \frac{\text{number of database videos which have } (I_1^*, I_2^*, \cdots, I_L^*) \text{ as non-zero indices}}{\text{number of database videos}}$$

Using (21), we find that $p^{\text{database}}_{(I_1^*, I_2^*, \cdots, I_L^*)}$ equals 0.0082, 0.0068, 0.0011, 1.25×10^{-4}, and 8.09×10^{-5}, for $L = 1$, 2, 3, 4 and 5, respectively. For $L > 5$, the probability value equals 0. Thus, though the three highest values of p_{V_1} are obtained for $L = 6$, 7 and 8 (Table 3), the corresponding L-tuples of VQ indices $(\{I_1^*, I_2^*, \cdots, I_L^*\})$ *have not occurred together* in any of the 38000 database videos.

8 VQ-M1 - Method 1 for Dataset Pruning for VQ-based Signatures

VQ-M1 uses a multi-pass approach for pruning. The motivating logic is that *for a given query, the model videos which are nearest to it are likely to have some or all of the non-zero dimensions, as the query signature itself, as non-zero*.

The storage costs involved with VQ-M1 are listed below.

• We store a proximity matrix $\mathbb{P} \in \mathbb{R}^{U \times U}$ which stores the U nearest neighbors, in proper sequence, for a certain VQ codevector. E.g. $\mathbb{P}(i,j)$ denotes the j^{th} NN for the i^{th} VQ codevector. For $U = 8192(2^{13})$, the storage cost of $\mathbb{P} = U^2.13$ bits (each of the U^2 terms represents an integer $\in [0, 2^{13} - 1]$ and hence, is represented using 13 bits, giving a total storage cost of 109 MB).

• We also store U clusters $\{\mathbb{C}(i)\}_{i=1}^{U}$, where $\mathbb{C}(i)$ denotes the cluster which contains those model video indices whose signatures have the i^{th} dimension as non-zero. The storage cost for 8192 clusters containing 38000 videos in all is found to be equal to 6.3 MB.

$$\mathbb{C}(i) = \{j : x_{j,i} > 0,\ 1 \leq j \leq N\} \tag{22}$$

We also maintain a distance matrix $\mathbb{D}' \in \mathbb{R}^{U \times U}$ which stores the NN distances, in ascending order, for each VQ codevector. Here, $\mathbb{D}'(i,j)$ denotes the distance of the $\{\mathbb{P}(i,j)\}^{th}$ codevector from the i^{th} VQ codevector, i.e. $\mathbb{D}'(i,j) = \mathbb{D}(i, \mathbb{P}(i,j))$. We do not need to store $\mathbb{D}'$ explicitly as it can be computed using $\mathbb{D}$ and $\mathbb{P}$. We now provide a list of symbols used in VQ-M1 (Algorithm 1) along with their definitions:

• $\mathbb{S}_j$: the set of distinct model videos considered in the j^{th} pass,

• G: the set of non-zero query dimensions, where $G = \{t_1, t_2, \cdots, t_{N_q}\}$,

• d_j^*: the minimum of the distances of all non-zero query dimensions to their j^{th} NN codevectors,

$$d_j^* = \min_{t_k \in G} \mathbb{D}'(t_k, j) \tag{23}$$

• A_j: the set of distinct VQ indices which are encountered on considering the first j NN for all the elements in G. Therefore, $(A_j \setminus A_{j-1})$ denotes the set of distinct (not seen in earlier passes) VQ indices encountered in the j^{th} pass, when we consider the j^{th} NN of the elements in G.

We maintain an ascending priority queue L of size K, for the K-NN videos, which is updated after every iteration. For the j^{th} pass, we terminate the search for top-K NN if $d_j^* \geq L_{K,2}$ (or if all the N model videos have already been considered). The proof that we are assured of finding the top-K NN is shown in Appendix 9. In the first pass, we consider the union of the clusters which correspond to the non-zero query dimensions. We consider all the model videos from this union for distance computation. For the 1^{st} pass, d_1^* equals 0 and the second pass is almost always required. In the j^{th} pass, we find the j-NN codevector of the non-zero query dimensions and the new codevectors (not seen in the earlier passes) are noted. We obtain the new model videos which have common non-zero dimensions with these newly encountered dimensions and consider them for distance computation. If the terminating condition is satisfied at iteration $j = J$, the sequence of model videos considered is given by $\{\mathbb{S}_1, \mathbb{S}_2, \cdots, \mathbb{S}_{J-1}\}$.

We find that the maximum number of iterations (J) needed to obtain all the K-NN for a given query increases with both K and the fractional query length (ℓ).

Algorithm 1 Algorithm for VQ-M1 - here, unique(E) returns the unique (without repeats) elements in E

Input: N model video signatures, $\overrightarrow{x_i} \in \mathbb{R}^U,\ 1 \leq i \leq N$

Input: the query signature $\vec{q}$, and lookup matrices $\mathbb{P}$ and $\mathbb{D}'$ (along with the lookup tables needed by the distance computation method VQLS-A/B)

Output: Best sequence to search N videos for top-K NN and also top-K NN (model video indices)

Initialization: (1^{st} pass)
$G = \{t_1, t_2, \cdots, t_{N_q}\}$, the non-zero query dimensions
$A_1 = G$, set of 1-NN of elements in G is G itself
$\mathbb{S}_1 = \bigcup_{1 \leq i \leq N_q} \mathbb{C}(t_i)$, set of model videos having at least 1 non-zero dimension from G
$d_1^* = \min_{t_k \in G}\{\mathbb{D}'(t_k, 1)\} = 0$
We maintain an ascending priority queue L of length K, based on the elements in $\mathbb{S}_1$, where $d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})$ is found using (4), if VQLS-A is being used.
End of 1^{st} **pass**
for $j = 2$ to U **do**
  $d_j^* = \min_{t_k \in G}\{\mathbb{D}'(t_k, j)\}$, minimum distance between non-zero query dimensions to their j^{th} NN
  if $L_{K,2} \leq d_j^*$ **or** $\sum_{k=1}^{j} |\mathbb{S}_k| = N$ (all model videos have been considered) **then**
    break;
  end if
  $B_i = \mathbb{P}(t_i, j),\ 1 \leq i \leq N_q$, $B =$ set of VQ indices which are j^{th} NN of elements in G
  $E = B \setminus A_{j-1},\ E =$ unique(E), set of VQ indices that are j^{th} NN of elements in G and were not seen in earlier iterations
  $\mathbb{S}_j = \bigcup_{1 \leq i \leq |E|} \mathbb{C}(E_i)$
  $\mathbb{S}_j = \mathbb{S}_j \setminus \bigcup_{1 \leq i < j} \mathbb{S}_i$, set of all model videos having at least one element in E as a non-zero dimension and these videos were not seen in earlier iterations
  $A_j = A_{j-1} \cup E$, set of all VQ indices which belong to one of the top j-NN for elements in G
  Update the priority queue L based on the elements in $\mathbb{S}_j$
end for
return the sequences observed so far $\{\mathbb{S}_1, \mathbb{S}_2, \cdots, \mathbb{S}_{J-1}\}$ (assuming that the search terminates at iteration $j = J$) and top-K NN from the priority queue L

9 Proof that top-K NN in VQ-based signature space are returned for VQ-M1 based pruning

The VQ-M1 algorithm along with the relevant notations have been explained in Sec. 8. The proof that we are guaranteed to return the top-K NN through this pruned search is explained below.

Given a dataset of $\{\overrightarrow{x_i}\}_{i=1}^{N}$ signatures, we present a lower bound of the minimum model-to-query distance, $\{\min_{1\leq i\leq N} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\}$, found for all signatures in the dataset (24). Here, $\beta(i, t_k)$ denotes the best matching dimension in $\overrightarrow{x_i}$ for dimension t_k.

$$\min_i d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) = \min_i \sum_{k=1}^{N_q} q_{t_k} \times \mathbb{D}(t_k, \beta(i, t_k)) \geq \min_i \sum_{k=1}^{N_q} q_{t_k} \times \{\min_j \mathbb{D}(t_j, \beta(i, t_j))\}$$

$$(\text{using } \sum_{k=1}^{N_q} q_{t_k} = 1) \ = \min_i \{\min_j \mathbb{D}(t_j, \beta(i, t_j))\} \tag{24}$$

Thus, the lower bound equals the smallest distance between a non-zero query dimension and the corresponding best-matched non-zero model dimension (24).

If any video index in $\mathbb{S}_j$, the sequence of model videos returned in the j^{th} iteration, is a j'^{th} NN of an element in G (the set of non-zero query dimensions), where $j' < j$, then that video would have been encountered in the first j' iterations and would not have been an element in $\mathbb{S}_j$. *Hence, the best match we can expect for a VQ index in G among the VQ indices in model videos in $\mathbb{S}_j$ is a VQ index which is a j^{th} NN of an element in G.*

We now show that the minimum model-to-query distance among all videos in $\mathbb{S}_j$ $\{\min_{i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\} \geq d_j^*$. As shown earlier, the *smallest "query dimension-to-best matched model dimension" distance* is due to a model dimension which is the j^{th} NN of a certain non-zero query dimension. Thus, the best matched model dimension for t_k, $\beta(i, t_k)$, where $i \in \mathbb{S}_j$, equals $\mathbb{P}(t_k, j)$, where $t_k \in G$.

$$\min_{i,\ i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) \geq \min_{i,\ i\in\mathbb{S}_j} \{\min_{t_k\in G} \mathbb{D}(t_k, \text{best matching model dimension in } \overrightarrow{x_i} \text{ for } t_k)\}, \text{ from (24)}$$

$$\min_{i,\ i\in\mathbb{S}_j} \{\text{best matching model dimension in } \overrightarrow{x_i} \text{ for } t_k\} = \mathbb{P}(t_k, j), \text{ where } t_k \in G$$

$$\therefore \min_{i,\ i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) \geq \min_{i,\ i\in\mathbb{S}_j} \min_{t_k\in G} \mathbb{D}(t_k, \mathbb{P}(t_k, j)), \text{ and } \mathbb{D}(t_k, \mathbb{P}(t_k, j)) = \mathbb{D}'(t_k, j)$$

$$\text{Now, } d_j^* = \min_{t_k\in G} \mathbb{D}'(t_k, j), \text{ as expressed before in (23)}$$

$$\therefore \min_{i,\ i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q}) \geq d_j^* \tag{25}$$

When we consider videos in $\mathbb{S}_j$, during the j^{th} pass, $\{\min_{i,\ i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\} \geq d_j^*$. If $d_j^* \geq L_{K,2}$, it is assured that $\{\min_{i,\ i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\} \geq L_{K,2}$, using (25). Now, if for the j^{th} pass, it is ensured that

$\{\min_{i,\, i\in\mathbb{S}_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\} \geq L_{K,2}$, then is it guaranteed that for videos in the j'^{th} pass (for $j' > j$), $\{\min_{i,\, i\in\mathbb{S}'_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\} \geq L_{K,2}$? If yes, then the minimum model-to-query distance seen in later iterations will always be greater than or equal to $L_{K,2}$ and these models will never be among top-K candidates.

Explanation : $\because$ the elements in $\mathbb{D}'$ are sorted in ascending order along a row,

$$\mathbb{D}'(t_k, j) \leq \mathbb{D}'(t_k, j'), \forall t_k \in G, j < j',$$

$$\therefore \min_{t_k \in G} \mathbb{D}'(t_k, j) \leq \min_{t_k \in G} \mathbb{D}'(t_k, j') \Rightarrow d^*_j \leq d^*_{j'}, \text{ using (23)}$$

$$\therefore L_{K,2} \leq d^*_j \text{ (given condition)}, \; d^*_j \leq d^*_{j'}, \; d^*_{j'} \leq \{\min_{i,\, i\in\mathbb{S}'_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\}, \text{ using (25)}$$

$$\Rightarrow \; L_{K,2} \leq \{\min_{i,\, i\in\mathbb{S}'_j} d_{VQ}(\overrightarrow{x_i}, \overrightarrow{q})\}$$

Hence, it is confirmed that **if $d^*_j \geq L_{K,2}$, we will not find a model video in any sequence $\mathbb{S}_{j'}$, where $j' > j$, with model-to-query distance less than $L_{K,2}$.**

References

[1] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[2] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, Englewood Cliffs, New Jersey, 1993.