The requirements regarding keyframe extraction in a Video Copy Detection (VCD) application can be summarized as follows:

- At least one keyframe per shot
- At least one keyframe per sub-shot
- The extracted keyframes should preserve their temporal position
- Real-time performance

Real-time performance is accomplished by exploiting DCT coefficients of I-frames by partial decoding of MPEG stream

Feature extraction (f_I) of the currently decoded frame

Calculate current ratio R_c (R_c > 1.0):

\[ R_c = \max \left( \frac{f_{I1} - f_{I0}}{f_{I1} - f_{I1}}, \frac{f_{I0} - f_{I1}}{f_{I0} - f_{I0}} \right) \]

Get binary values of R_c and R_f:

\[ (R_c)_{bin} = \begin{cases} 1, & \text{if } R_c > T_c \\ 0, & \text{otherwise} \end{cases} \]

\[ (R_f)_{bin} = \begin{cases} 1, & \text{if } R_f > T_f \\ 0, & \text{otherwise} \end{cases} \]

Calculate k_flag

\[ k_{flag} = (R_c)_{bin} \oplus (R_f)_{bin} \]

As a result of the logical operation, k_flag = 0 if both ratios R_c and R_f follow the same behavior, i.e., either both have values greater, or less than T. Transition is detected if k_flag = 1 where a keyframe is selected.

Store updated values of ratio (R_c ← R_c and feature vectors (f_{I1} ← f_{I0} and f_{I0} ← f_{I1})) and continue.

REMARK:

The index of each candidate keyframe is examined and only if it is greater than the index of the last detected keyframe it is stored as keyframe.

The goal is to detect two keyframes per shot, one at the begging and one at the end of each shot (as well as for each sub-shot for complex scenes involving camera/object motion), which will be used as indicators for effective similarity matching between original videos and copies. Triples of I-frames are examined on the basis of change detection, either when shot transition occurs or camera/object motion is involved (sub-shots).

A subset of eight MPEG-1 video sequences from TRECVid 2007 test video collection [2] were selected for evaluation. The evaluation of the proposed algorithm is carried out in terms of the number and the percentage of the extracted keyframes and the processing time.

### Evaluation

A typical example of how the algorithm works in a real-world application scenario is presented in Fig.5. A popular TV spot was used as query and many versions of it (copies and near-duplicates) were downloaded from YouTube and converted to MPEG-1 format.

The proposed algorithm results in a compact set of keyframes, where the keyframe percentage is limited to a maximum of 2.5% (1.5% on average) (Fig.4b). Real-time performance is achieved (Fig.4c), not only due to DCT-domain feature extraction but also by bypassing the process of temporal video segmentation.

---

**References**

