Video Rushes Summarization Using Spectral Clustering and Sequence Alignment

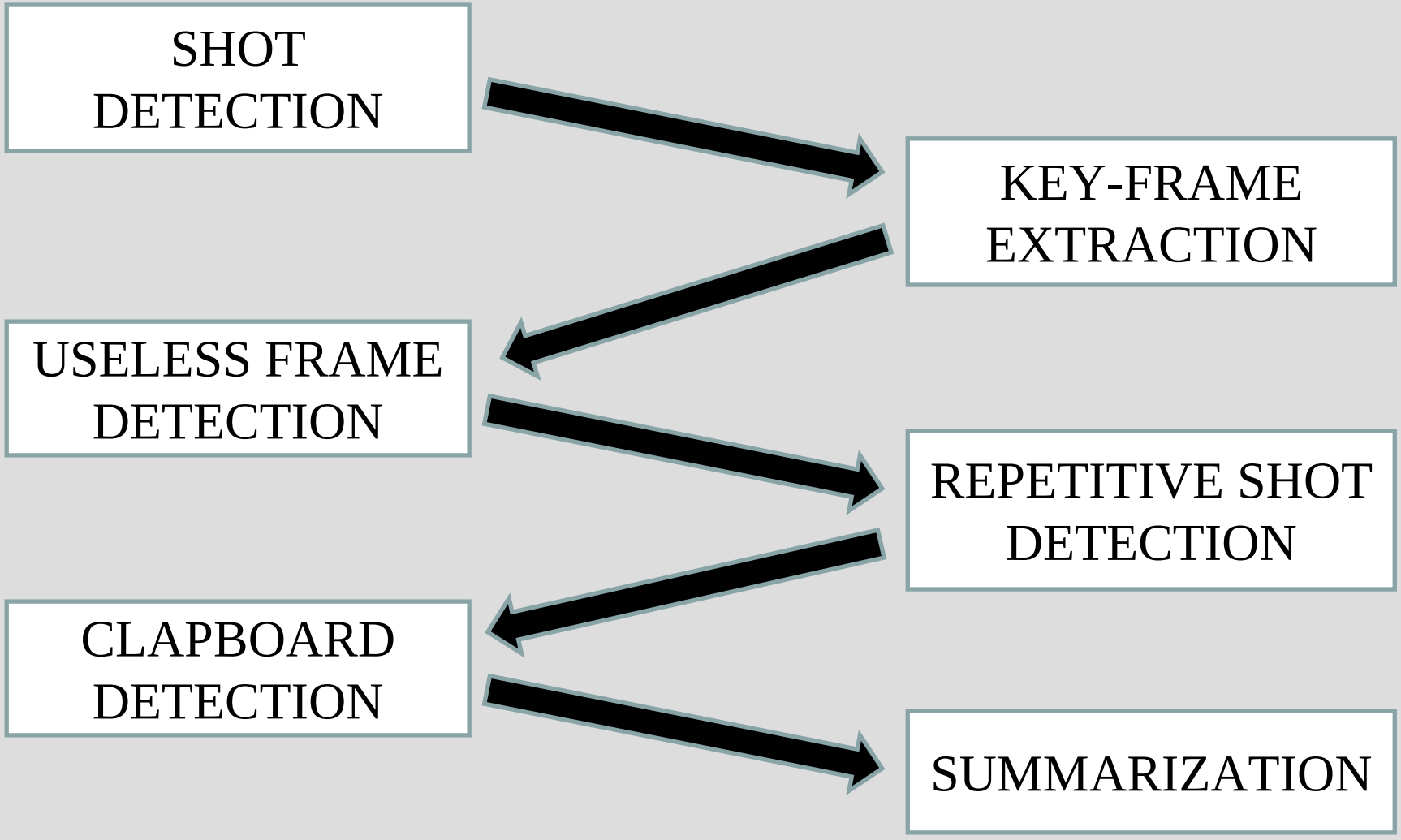
Vasileios Chasanis and Aristidis Likas
Department of Computer Science
University of Ioannina, Greece

Nikolaos Galatsanos
Department of Electrical and Computer Engineering
University of Patras, Greece

Challenges and Contribution

- Efficient representation of video segments.
  - Key-frame extraction based on spectral clustering.
  - Employment of fast global k-means.
  - Estimation of number of key-frames.
- Removal of useless frames.
  - Edge direction histograms and SIFT descriptors.
- Detection of similar video segments.
  - Shot similarity metric based on sequence alignment algorithm.

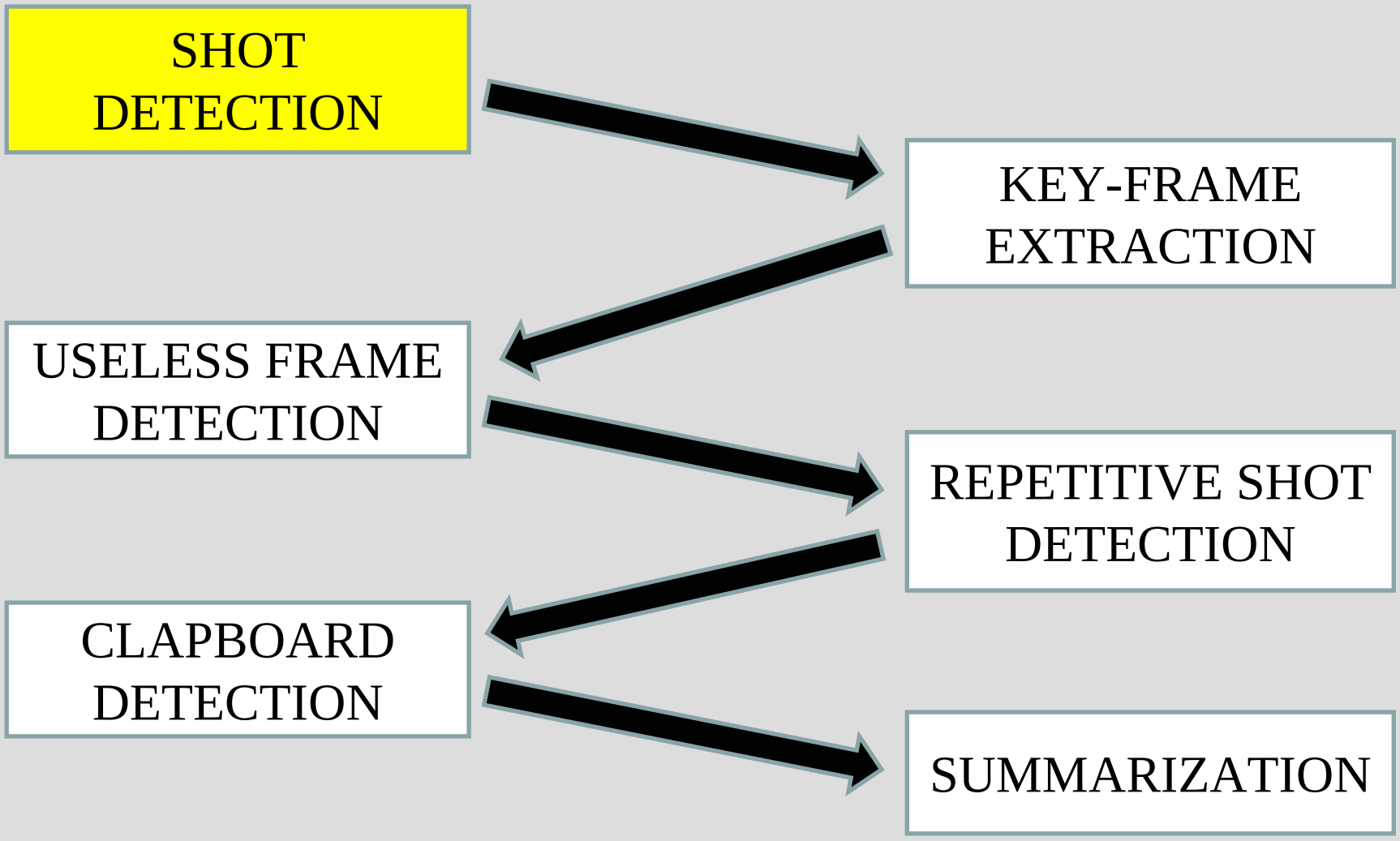
Proposed Method



Feature Extraction

- Each video is sampled uniformly keeping only 5 frames per second.
- For each frame an HSV normalized histogram is used, with 8 bins for hue and 4 bins for each of saturation and value resulting to 128 (8×4×4) bins.

Proposed Method



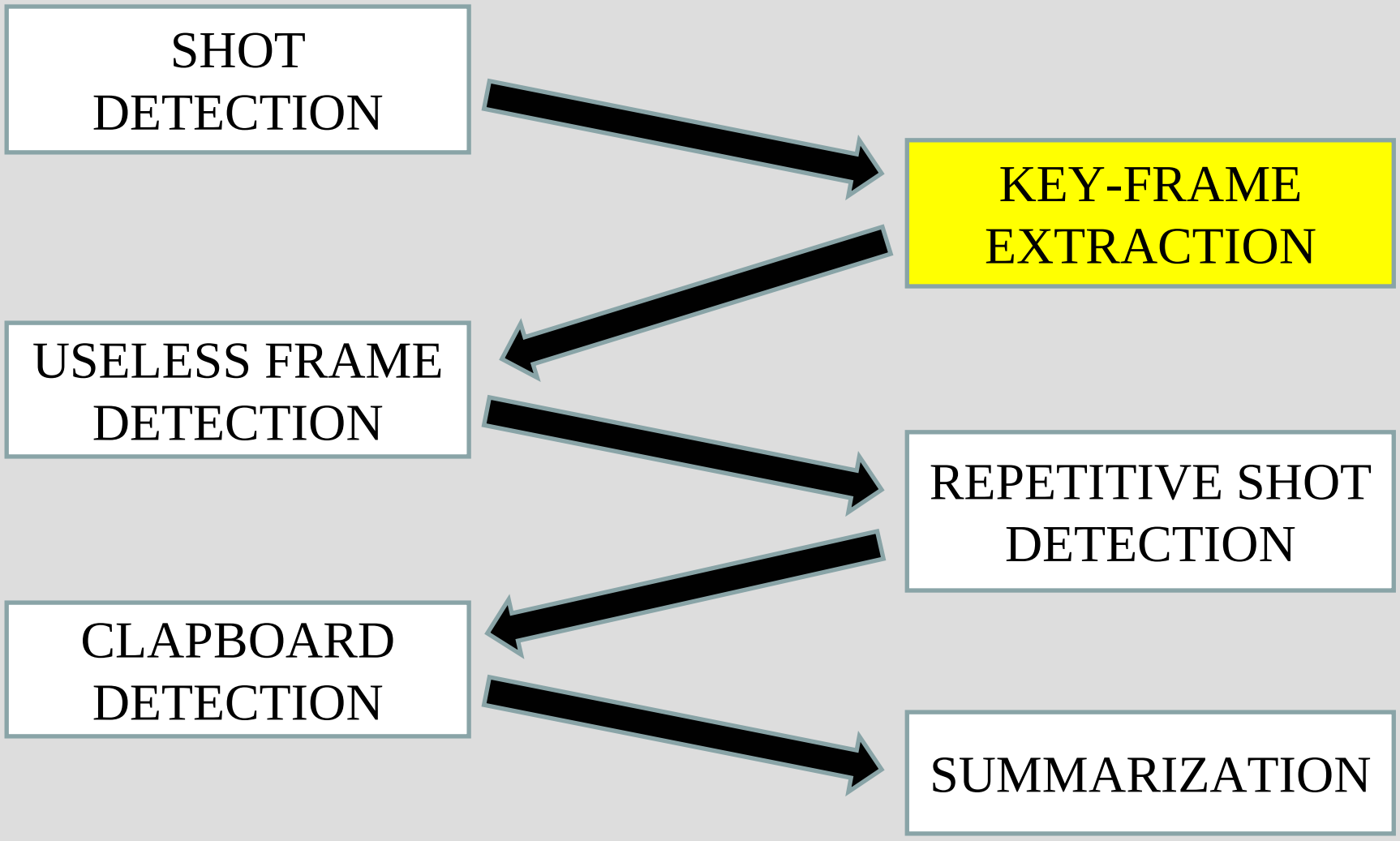
Shot Detection

- Calculate the sum of the bin-wise differences of adjacent frames and compare them to a threshold.
- Given two images I_i and I_j and their corresponding histograms H_i and H_j , their difference is:

$$d(I_i, I_j) = \sum_{k=1}^{128} \frac{\left(H_i(k) - H_j(k)\right)^2}{H_i(k) + H_j(k)}$$

- Threshold was set to 0.15 and shots shorter than 1 second were removed.

Proposed Method



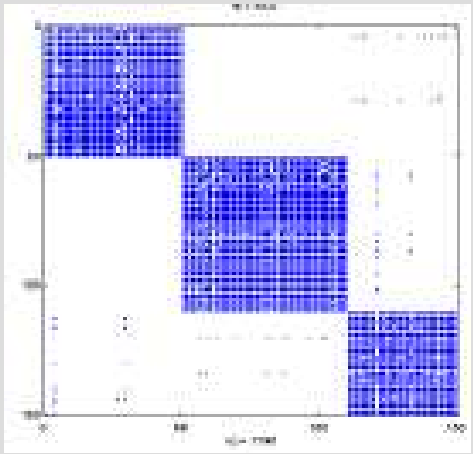
Key-frame Extraction

- Each shot must be represented by unique frames that capture the whole content of a shot.
- The frames of each shot are clustered into groups using an improved spectral clustering algorithm.
- The medoids of the obtained groups are selected as the key-frames of the shot.
- A medoid is defined as the frame of a group whose average pairwise similarity to all other frames of this group is maximal.

Spectral Clustering

Given a set of frames $F=\{F_1,\ldots,F_N\}$ to be partitioned into M groups.

1. Compute similarity matrix A=[α(i,j)], with α(i,j)=sim(F_i,F_j)
2. Eigenvalue computation of matrix $\Phi = I - D^{-1/2} A D^{-1/2}$
3. Construct the eigenvector matrix U=[u_1,…u_M] (top eigenvectors)
   - each frame F_k is represented by an M-dimensional real vector y_k corresponding to the k-th row of U
4. Cluster the rows of U into M groups using k-means



$$\begin{bmatrix} U_1 & U_2 & U_3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Spectral Clustering - Analysis

- $Z=[Z_1,\ldots,Z_M]$: partition matrix representing a clustering solution
  - Column vector Z_j is the binary indicator vector for group G_j:

$$Z(i,j)=1:\text{if } i\in G_j$$

$$Z(i,j)=0:\text{otherwise}$$

$$Z^T Z = I_M$$

- The optimal partition is defined as the optimal solution to the following problem (Spectral clustering objective):

$$\max_Z trace\left(Z^T \Phi Z\right)$$

$$\text{s.t. } Z^T Z = I_M \text{ and } Z(i,j)\in\{0,1\}$$

Spectral Clustering - Analysis

- The spectral approach (for *M* clusters) provides solution to the following continuous optimization problem (relaxation):

$$\max_{Y} trace\left(Y^T \Phi Y\right)$$

$$\text{s.t. } Y^T Y = I_M$$

- Relaxing Y into the continuous domain turns the discrete problem into a continuous optimization problem.

- Optimal solution attained at: $Y^* = U_M = [u_1, \ldots, u_M]$
  - u_i are the eigenvectors corresponding to the ordered top *M* eigenvalues λ_i of Φ.

Number of Key-frames

- The optimal value of the objective function for M clusters is: $sol(\mathrm{M}) = \max_{Y} trace(Y^T \Phi Y) = \lambda_1 + \lambda_2 + \ldots + \lambda_M$
- The improvement from adding cluster M+1 is:

$$sol(M+1) - sol(M) = \lambda_{M+1}$$

- When λ_{M+1} lower than a threshold, the improvement is negligible and the number of clusters is assumed to be M.

1. The proposed approach:
   - Compute and sort eigenvalues: $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_N$.
   - Determine the largest eigenvalue $\lambda_{M+1} < T$ (T=0.005 in all experiments).
   - Select M as the number of clusters.

Global k-Means

- Global k-means [Likas et al, 2003] is an incremental, deterministic clustering algorithm that runs *k*-Means several times to provide near optimal solutions.

- Idea: a near-optimal solution for k clusters can be obtained by running k-means from an initial state $(m_1, m_2, ..., m_{k-1}, x_n)$
  - the k-1 centers are initialized from a near-optimal solution of the (k-1)-clustering problem $(m_1, m_2, ..., m_{k-1})$
  - the k-th center is initialized at some data point x_n (which?)

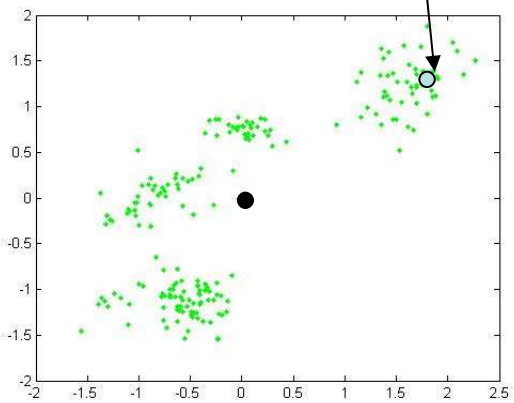
Global k-Means

In order to solve the *M*-clustering problem:

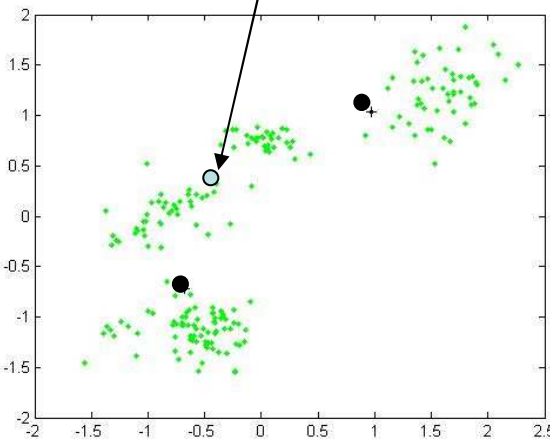
1. Solve the 1-clustering problem (trivial)
2. Solve the *k*-clustering problem using the solution of the (k-1)-clustering problem $(m_1, m_2, ..., m_{k-1})$
   a) Execute *k*-Means *N* times, initialized as at the *n*-th run (n=1,…,N). $(m_1, m_2, ..., m_{k-1}, x_n)$
   b) Keep the solution corresponding to the run with the lowest clustering error as the solution with *k* clusters $(m_1, m_2, ..., m_k)$
3. k:=k+1, Repeat step 2 until k=M.

ü All intermediate solutions for k=1, …, M-1 are also found

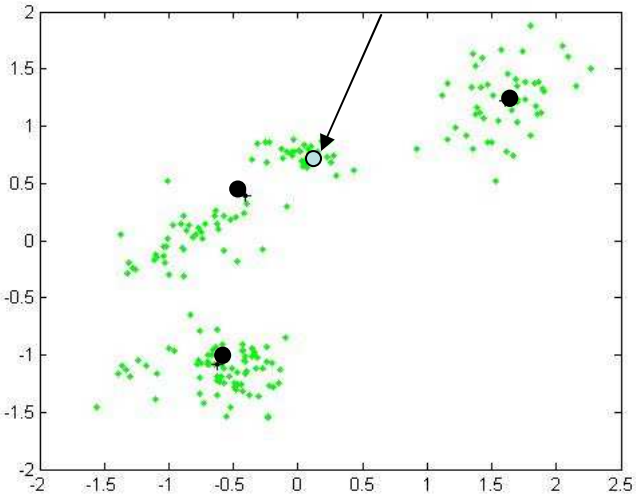
Best Initial m_2



Best Initial m_3

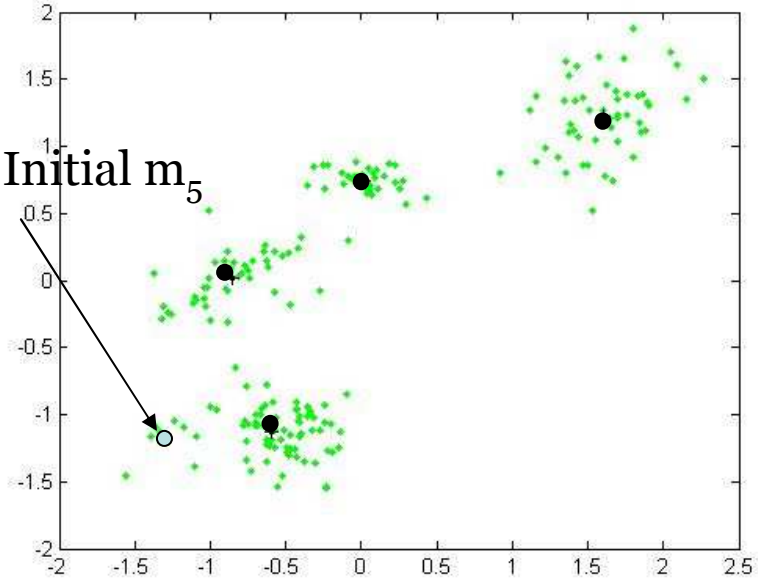


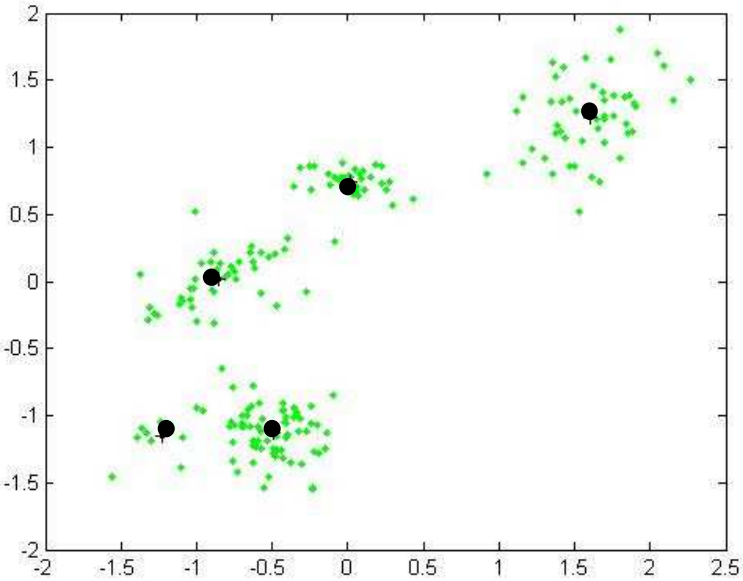
Best Initial m_4



Empty circle: optimal initial position of the cluster center to be added

Best Initial m_5

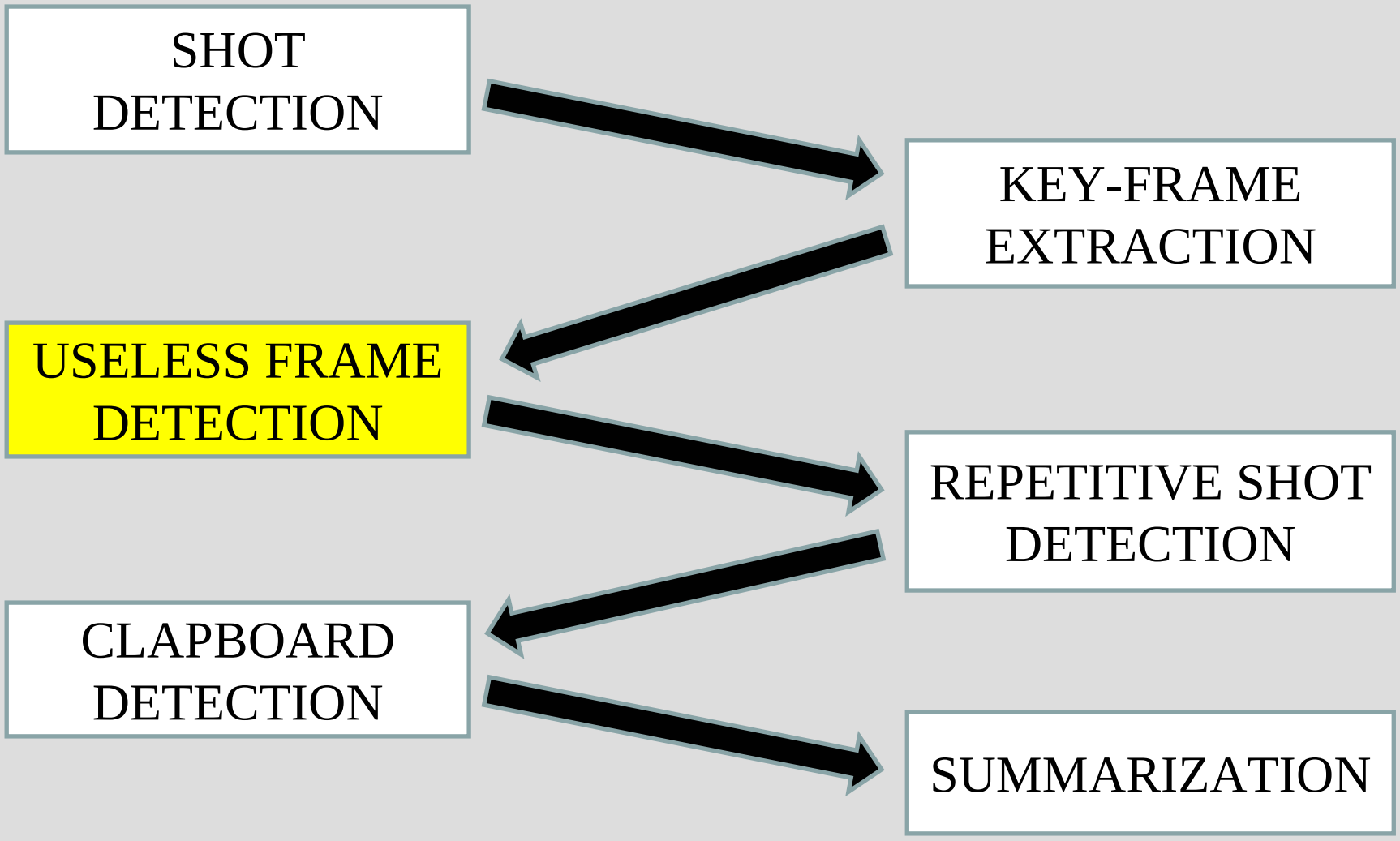




Fast Global k-Means

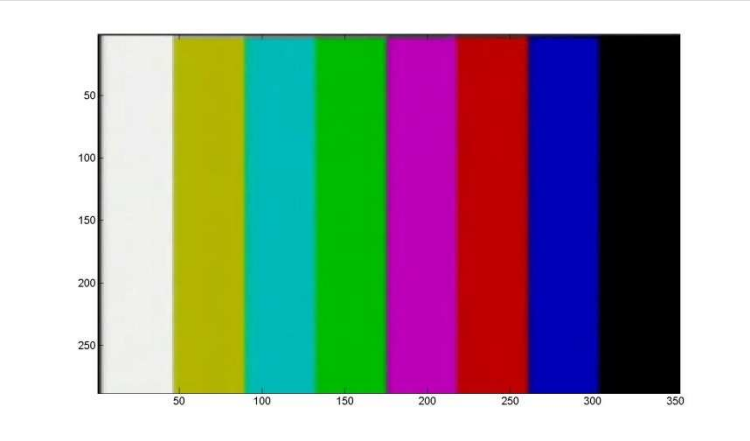
- Fast Global k-means algorithm reduces the computational cost of the global k-means algorithm without significant loss in the quality of the solution.
- Initially, a new cluster center is placed at position x_n and an upper bound E_n of the final clustering error is obtained.
- The initial position of the new cluster center is selected as the point x_i for which E_n is minimum and the k-means runs only once for each k.

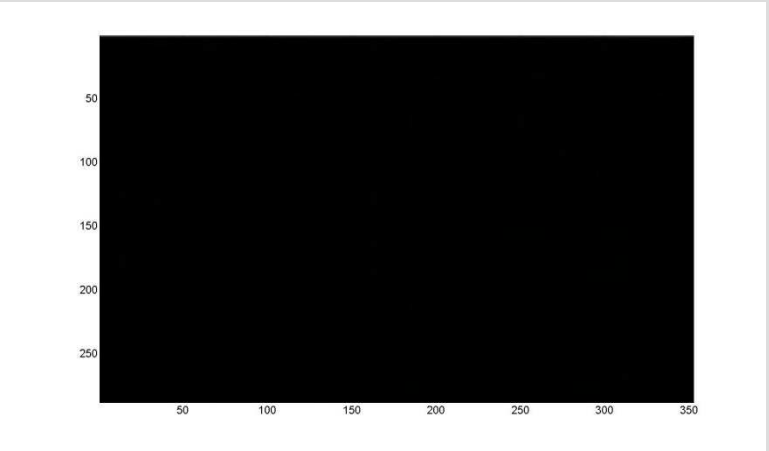
Proposed Method



Useless Frames Detection

- Video rushes contain many useless frames such as colorbars and monochrome frames.





- The shot detection algorithm usually isolates colorbars or monochrome frames into single shots.

Useless Frames Detection

- Thus, to speed up the implementation process the first key-frame of each shot is checked.
- If it is a useless frame, then the corresponding shot is removed from the summarization process.
- The edge direction histogram of the first key-frame is computed.

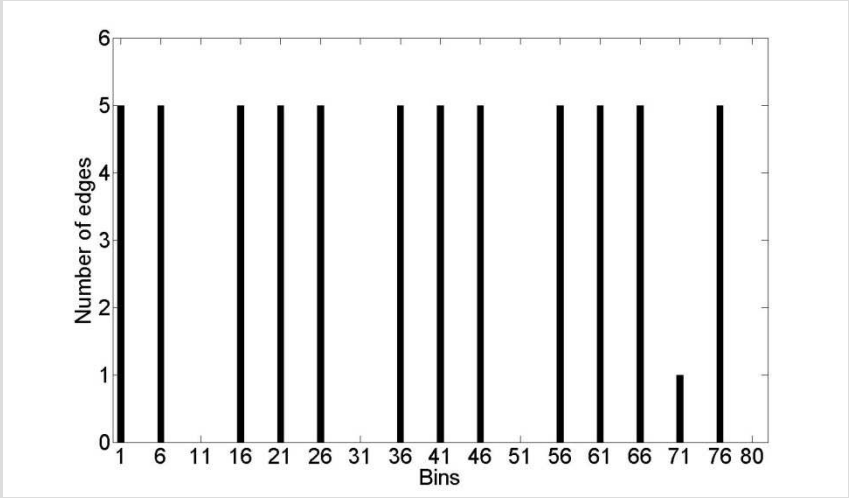
Edge Direction Histogram

- The key-frame is divided into 16 sub-images.
- The local edge histogram for each sub-image is computed.
- Edges are grouped into five categories:
  - Vertical
  - Horizontal
  - 45 diagonal
  - 135 diagonal
  - Isotropic (nonorientation specific)

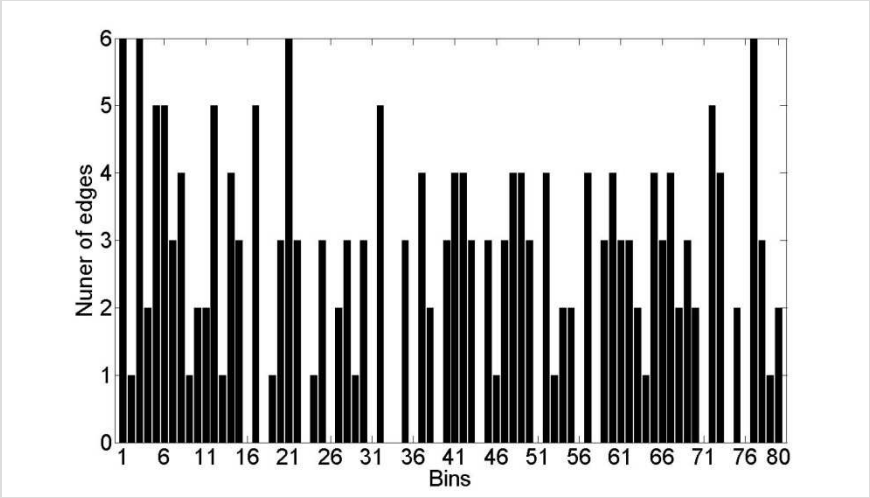
Edge Direction Histogram

- The final edge direction histogram is a 80-bin histogram.
- The edge direction histogram for a colorbar produces peaks in vertical and horizontal bins whereas the other bins are close to zero.
- The bins of the edge direction histogram for a monochrome frame are all close to zero.

Edge Direction Histograms



Colorbar



Normal frame

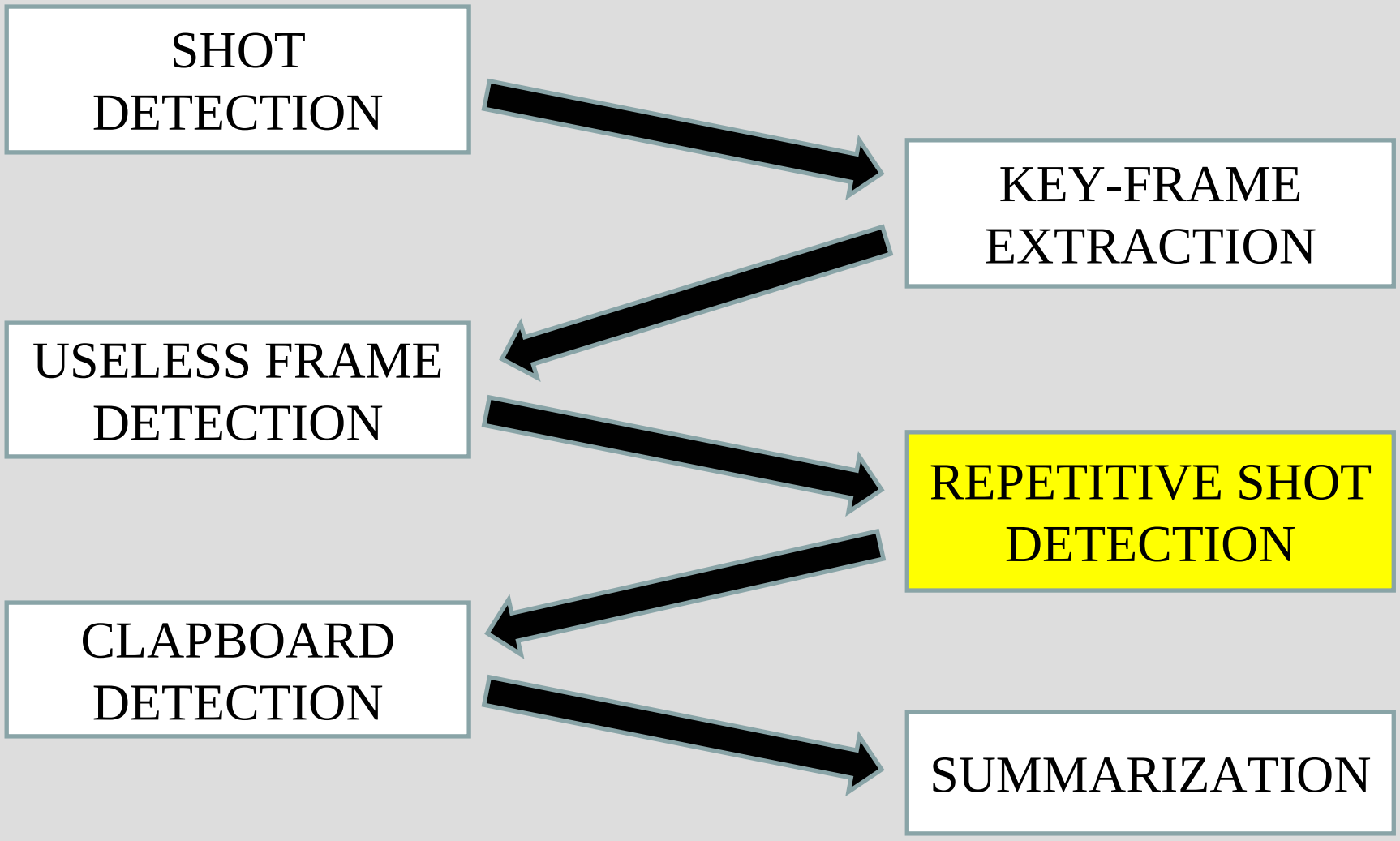
Useless Frames Detection

- To detect a useless frame we compare the sum of all bins and the sum of vertical and horizontal bins.
- The image I_i with corresponding edge direction histogram E_i is characterized as useless if:

$$\sum_{k=1}^{128} E_i(k) - \sum_{m=0}^{15} E_i(5m+1) - \sum_{m=0}^{15} E_i(5m+2) < T_{edh}$$

  - $E_i(5m+1)$ are the vertical bins.
  - $E_i(5m+2)$ are the horizontal bins.
  - Threshold T_{edh} was set to 10.

Proposed Method



Redundant Information Removal

- The same scene is usually taken many times until the desired result is produced, thus producing repetitive information.
- Our goal is to group “similar” shots and keep only one representative for each group that will be further analyzed and contribute to the final summary.

Visual Shot Similarity Metric

- In rushes, two shots that describe the same scene are expected to be represented by key-frames that follow the same time order.
- Thus, a segment of one shot or the whole shot will also appear in the other shot.
- To find similar segments in two shots we use a sequence alignment algorithm between the sets of their key-frames.

Visual Shot Similarity Metric

- Each key-frame is “matched” with the most similar (visually) key-frame of the other shot.
- Temporal order of key-frames is also taken into consideration.
- Suppose we are given one shot describing the following events E_1, E_2, E_3 , E_4 , E_5, E_6 and another shot describing events E_2, E_3, E_5, E_6 .

Sequence Alignment Algorithm

- An optimal alignment of two shots is:

$$Seq_1 \quad : \quad E_1E_2E_3E_4E_5E_6$$
$$Seq_2 \quad : \quad E_2E_3E_5E_6$$

Seq_1	E_1	E_2	E_3	E_4	E_5	E_6
Seq_2	–	E_2	E_3	–	E_5	E_6

- The score of the sequence alignment algorithm constitutes the final shot similarity metric.
- The “Smith-Waterman” local sequence algorithm is used which compares segments of all possible lengths and optimizes the similarity measure.

Substitution-Similarity Matrix

- This algorithm requires a substitution matrix.
- Suppose we are given two shots S_i and S_j and their corresponding key-frame-sets:
  - $KF_i = \{KF_i^1, KF_i^2, \ldots, KF_i^m\}$
  - $KF_j = \{KF_j^1, KF_j^2, \ldots, KF_j^n\}$
- Construct a $m \times n$ similarity matrix SM with elements:

$$SM(m,n) = VisSim\left(KF_i^m, KF_j^n\right)$$

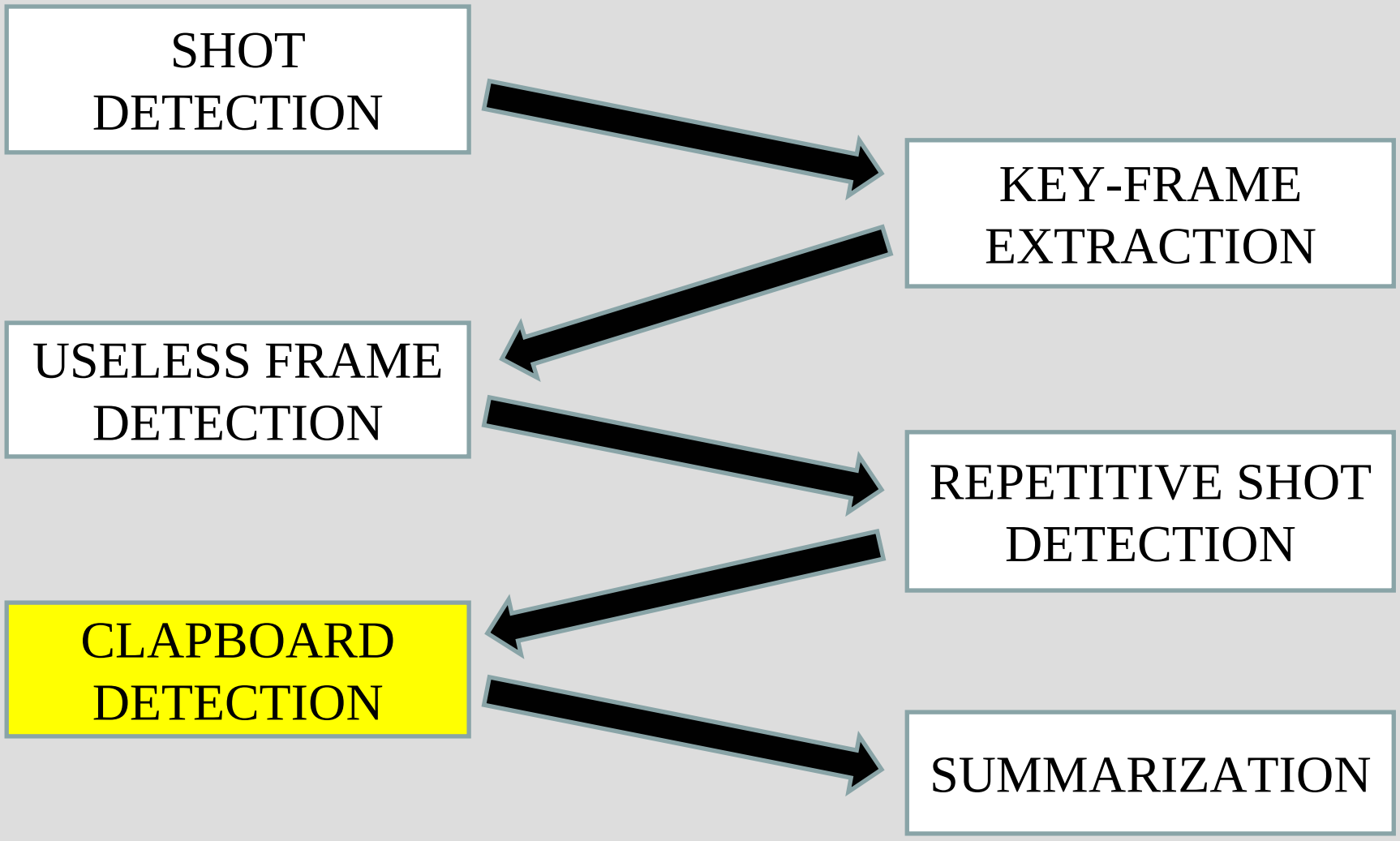
where $VisSim$ is the visual similarity between two frames I_i and I_j.

$$VisSim(I_i, I_j) = 1 - d(I_i, I_j) \qquad d(I_i, I_j) = \sum_{k=1}^{128} \frac{\left(H_i(k) - H_j(k)\right)^2}{H_i(k) + H_j(k)}$$

Repetitive Shot Detection

- To find groups of repetitive and similar shots, each shot is compared with the next three.
- If one of the threes shots is similar with the shot under consideration, then all the shots between these two shots and the shots under consideration form a group.
- If none of these shots is similar then a new group of shots is created.
- Two shots are considered similar if the score of the sequence alignment of their key-frames exceeds a predefined threshold (set 0.88).
- Finally, the shot of each group with the largest duration is selected as the representative of this group.

Proposed Method



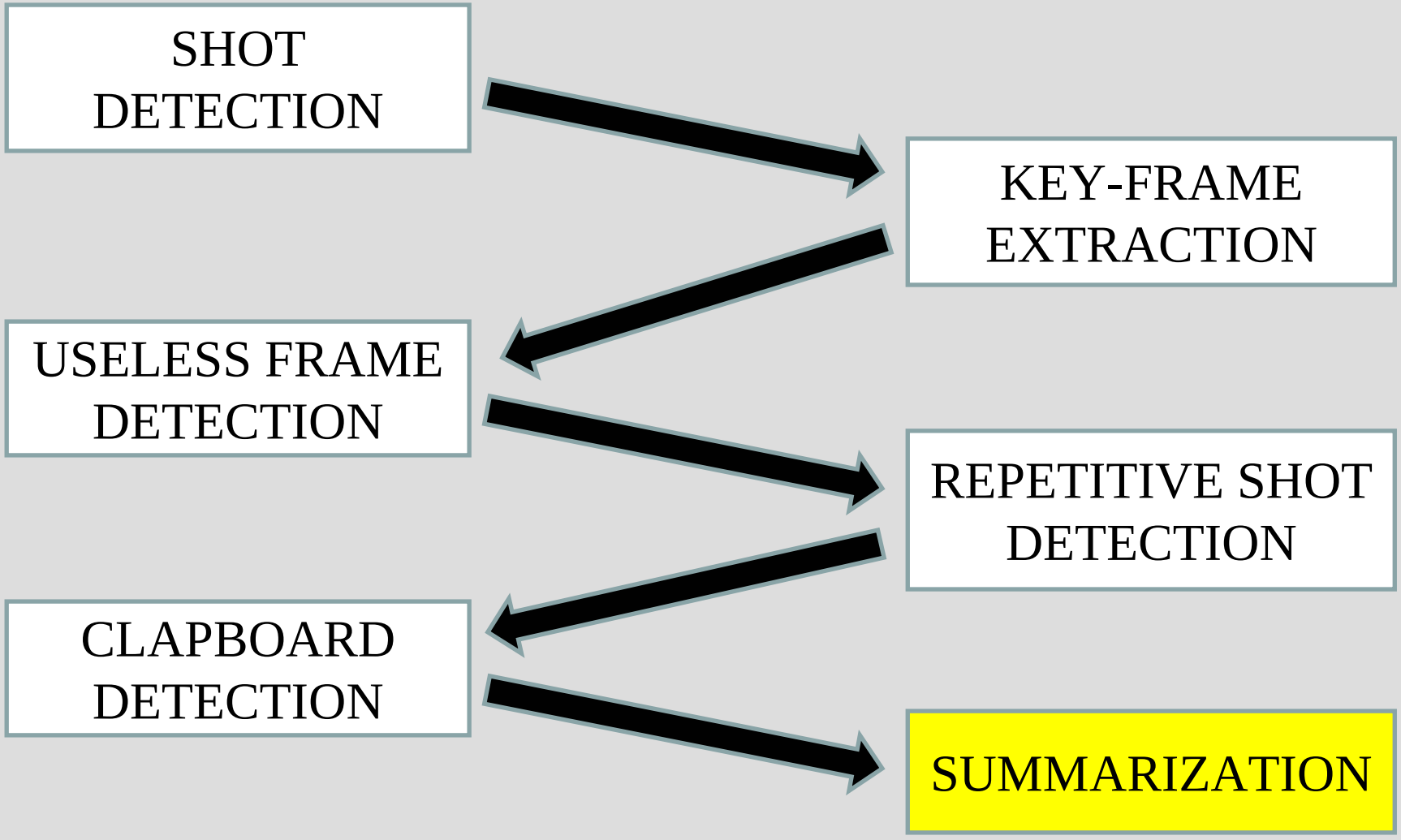
Clapboard Removal

- So far we have selected unique and non-repetitive shots represented by their key-frames.
- To detect clapboards, we compute for each key-frame the scale invariant feature transforms (SIFT).
- Using the TRECVID 2007 Development Data, a database of approximately 150 frames containing only clapboards was generated and their SIFT descriptors were calculated.

Clapboard Removal

- Compare the descriptors of each key-frame with the descriptors of the database.
- If the number of matching descriptors is over a predefined threshold:
  - Key-frame is characterized as a clapboard.
  - Corresponding cluster is removed.
  - New key-frames are extracted.

Proposed Method



Summarization

- Create video summary with duration less than the 2% of the original video.
- For each group of repetitive shots, the shot with the largest duration is selected as the representative of this group.
- The duration of the summary of such a group is:

  $T_{sum} = 0.02 T_{all}$, where T_{all} is the duration of the group.
- Each shot is represented by *k* frames. To each key-frame a duration of $T_{kf} = T_{sum} / k$ is assigned to.
- Finally, sampling every 3 frames , the $\lfloor T_{kf} / 2 \rfloor$ preceding and $\lfloor T_{kf} / 2 \rfloor$ following frames of each key-frame are selected to summarize the shot (and corresponding group) under consideration.

Experiments

	Our method		All	
	Mean	Median	Avg.(Mean)	Avg.(Median)
DU (secs)	25.07	28.00	27.01	28.25
XD (secs)	6.64	5.17	4.69	3.93
TT (secs)	39.86	41.33	40.76	39.91
VT (secs)	27.57	30.33	29.31	30.47
IN (0-1)	0.53	0.56	0.44	0.44
JU (1-5)	3.31	3.33	3.17	3.21
RE (1-5)	3.16	3.33	3.30	3.36
TE (1-5)	2.50	2.33	2.76	2.75

Conclusions

- The proposed key-frame extraction algorithm provides efficient representation of shots.
- The score for the removal of useless frames (JU) of our method is above the average.
- However, clapboard removal could be further investigated and improved.
- The identification of repetitive information (RE) also needs improvement as indicated from the results.