



NIKON CORPORATION

Nikon Multimedia Event Detection System

Takeshi Matsuo and Shinich Nakajima
Optical Research Laboratory, Nikon Corporation

November 16, 2010

Contents

- Basic Concept
- Explanation of Nikon MED System
- Experimental Result
- Conclusion

Basic Concept

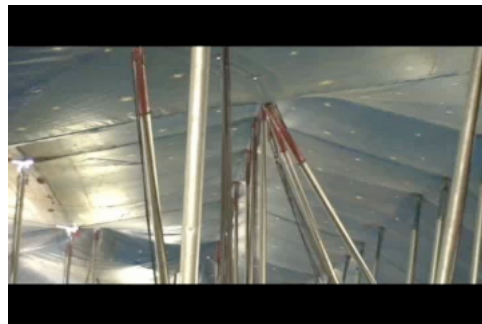
- We reduce the event detection of a set of video to the classification problem of one of images.
 - We don't think of audio information.
 - We rely on the assumption that a small number of images (**key-frames**) in a given video contains enough information for event detection.



A Key-frame should represent relevant contents in a given video.

Basic Concept

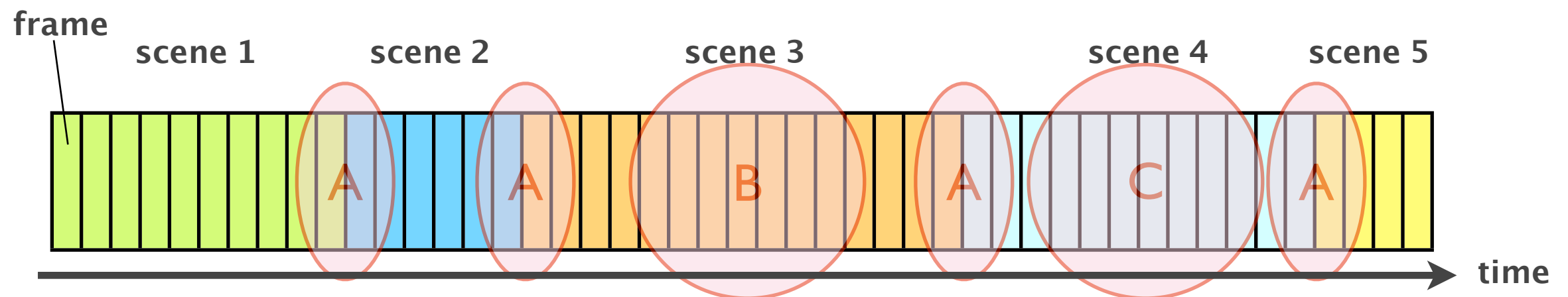
- We are interested in the key-frame extraction.
 - However, it is **hard** to extract **the best key-frame** of the video with the contents analysis such as the object recognition, human detection, motion analysis, etc.
 - We want to extract key-frame(s) **more easily** without these analysis.



Which is the best key-frame?

Basic Concept

- Where is/are the key-frame(s) in the video?
 - We focus on the characteristics of **scene** and **its length**.
 - Video consists of a time-ordered set of images.
 - A scene is a part of video and the unit of semantically-divided contents.



Frames near each scene change (A) are not key-frames.

Because there are in changing a photographer's interest, searching next objects, video effect, power on/off, etc.

Some frames in longer scenes (B and C) may be key-frames.

Because he/she keeps being more interested.

Basic Concept

- Our approach for key-frames extraction:
 - We extract a small number of frames which are not near scene change (edges of scene) in longer scenes of a given video.
 - As almost all frames in each scene are similar semantically and picture-compositionally (if the scene-cutting does well), we **don't need to extract the best key-frame in the scene.**
 - Multiple key-frames extraction reduces risk that a key-frame is not feasible.
- In feature extraction and classification, we adopt commonly-used methods respectively:
 - Scale invariant feature transform (SIFT) + bag-of-words,
 - Support vector machine (SVM).

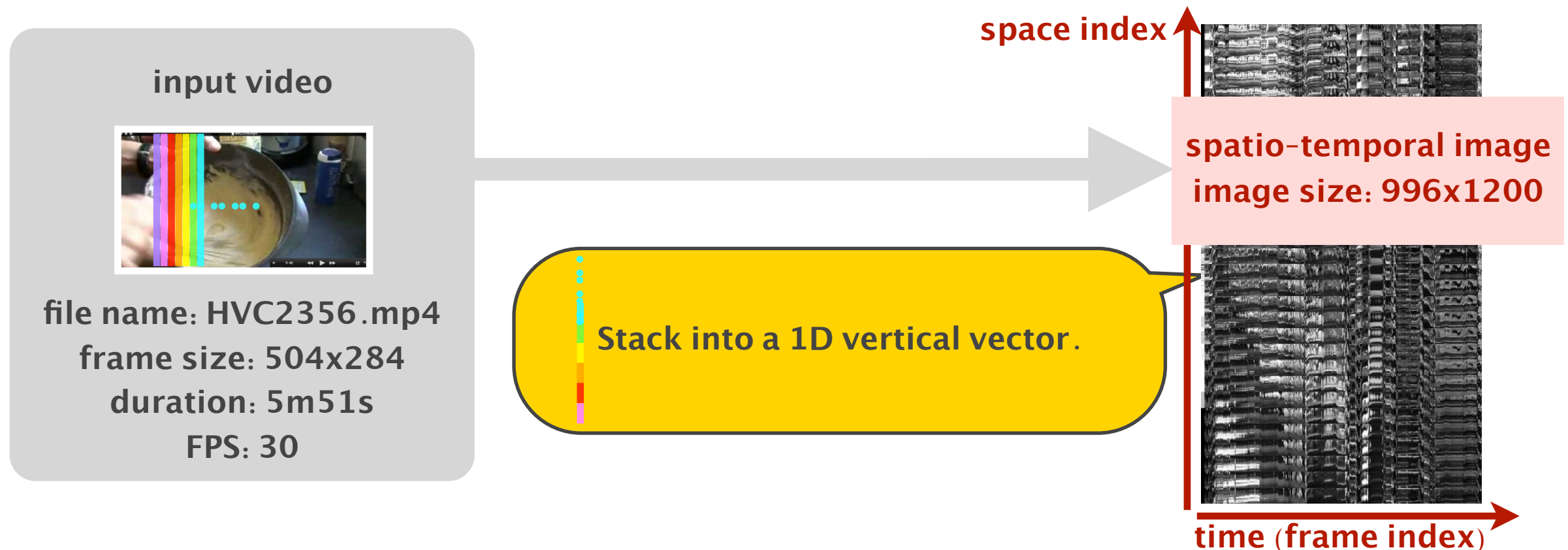
Explanation of Nikon MED System

- **System Overview**

- Step 1: **Spatio-temporal Image Creation**
- Step 2: **Scene-cut Detection**
- Step 3: **Key-frames Extraction**
- Step 4: **Bag-of-words Histogram Construction**
- Step 5: **Classification with SVM**

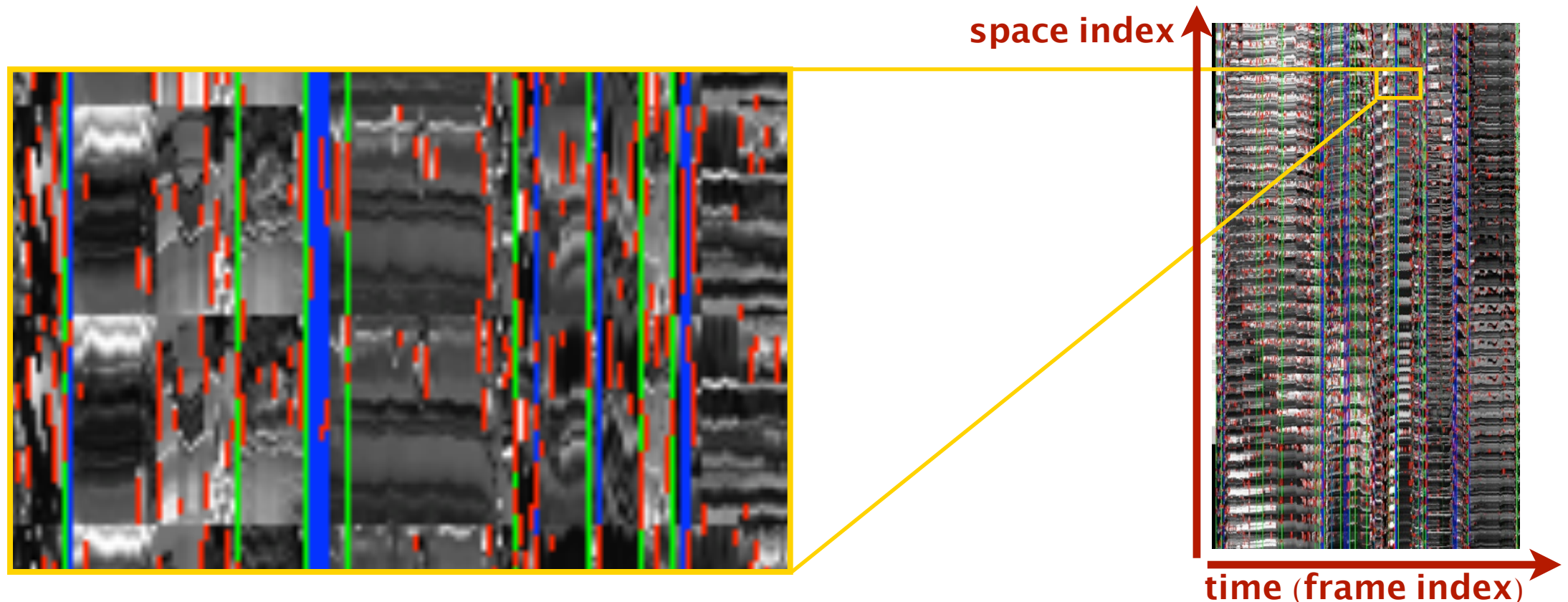
Step 1: Spatio-temporal Image Creation

- A given video is converted to a large 2D image.
 - Like as “visual rhythm” (Guimarães, et al. 2003).
 1. Sample frames at every 0.5 sec,
 2. Trim the frames into 4:3 and resize to 40x30 pixels,
 3. Convert the frames into gray images,
 4. Unfold the 2D structure of an images into a 1D vector.



Step 2: Scene-cut Detection

- Finding vertical line in the spatio-temporal image.
 1. Vertical edges detected by Canny detector (|),
 2. Frames gotten more than 1/60 votes (|),
 3. Scene-cuts sufficing minimum 2 sec internal constraint (|).
- There are room for improvement (ex. using “visual rhythm” or texture analysis).



Step 3: Key-frames Extraction

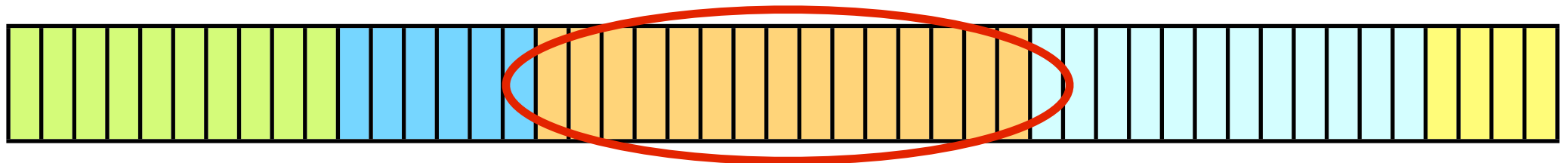
- **Key-(1,1) method:**

- This is **the most naive and simplest.**

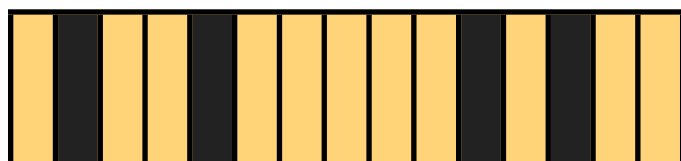
We extract a small number of frames which are not near scene change in longer scenes of a given video.

Almost all frames in each scene are similar semantically and picture-compositionally.

1. Select **the longest scene** in a given video.



2. Exclude dark frames in the scene.



3. Extract **the center** of the remain.



Step 3: Key-frames Extraction

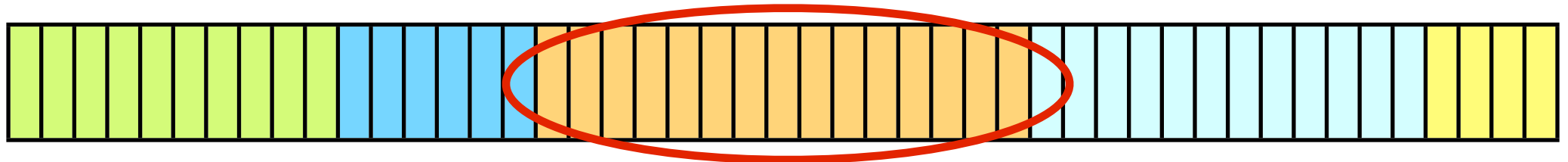
- **Key-(1,N) method:**

- This is **naive extension of Key-(1,1).**

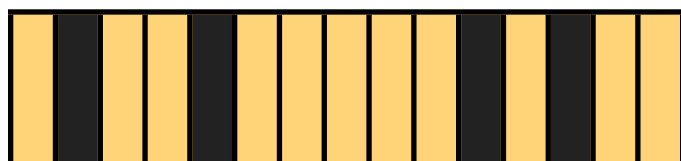
We extract a small number of frames which are not near scene change in longer scenes of a given video.

Almost all frames in each scene are similar semantically and picture-compositionally.

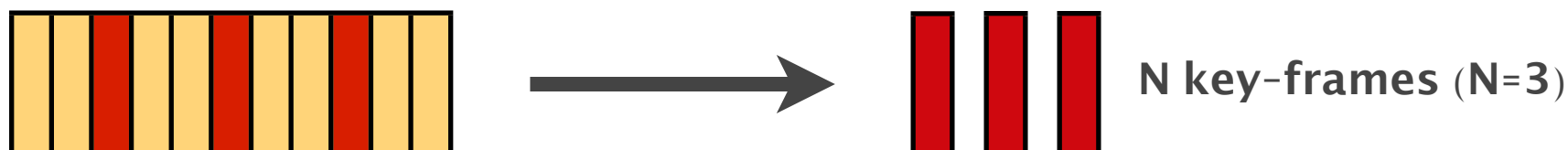
1. Select **the longest scene** in a given video.



2. Exclude dark frames in the scene.



3. Extract **N-frames** of the remain on a regular grid ($N=3$).



Step 3: Key-frames Extraction

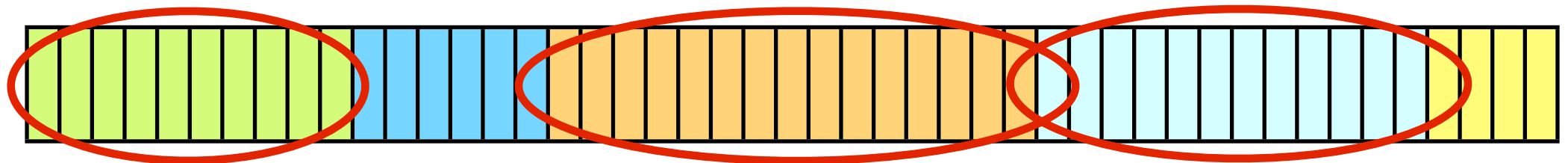
- **Key-($M,1$)** method:

- This is **another extension of Key-(1,1)**.

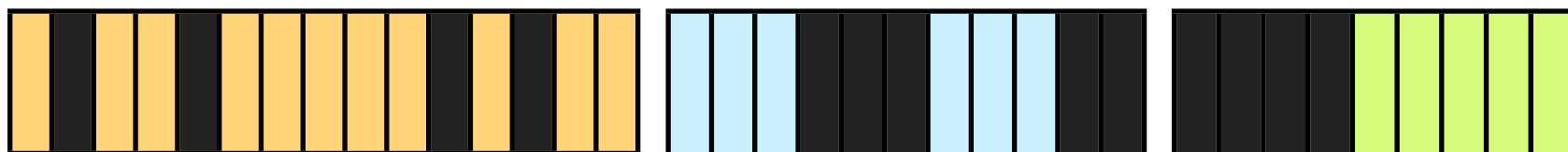
We extract a small number of frames which are not near scene change in longer scenes of a given video.

Almost all frames in each scene are similar semantically and picture-compositionally.

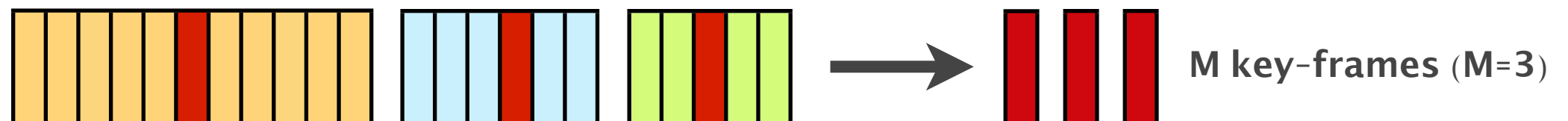
1. Select the **M -longest scenes** in a given video ($M=3$).



2. Exclude dark frames in the scenes.



3. Extract **the center** of each remains.



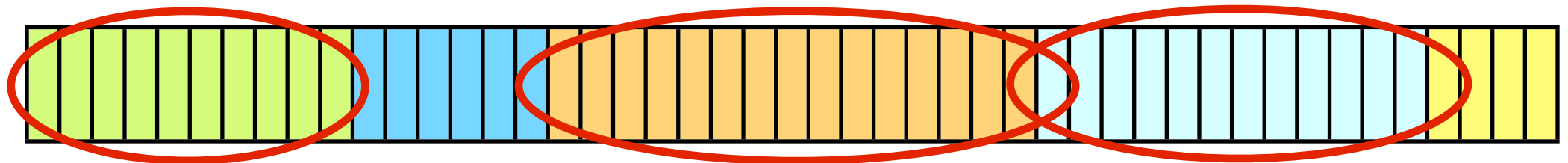
Step 3: Key-frames Extraction

- **Key- (M,N) method:**

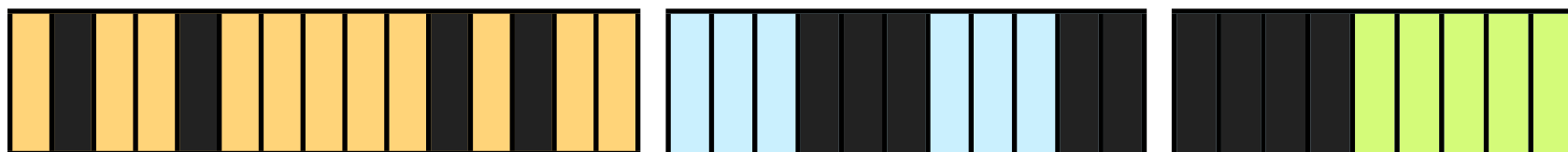
- This is **most general extension of Key- $(1,1)$** .

We don't implement yet.

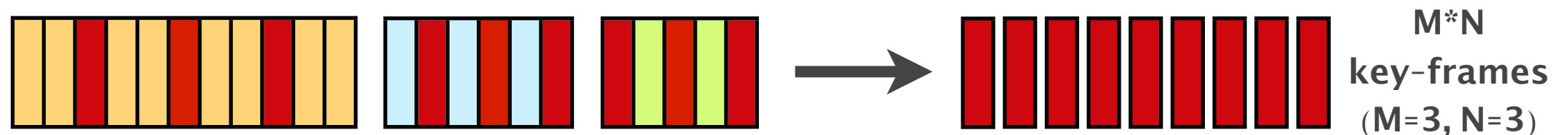
1. Select the **M -longest scenes** in a given video ($M=3$).



2. Exclude dark frames in the scenes.



3. Extract **N -frames** of the remain on a regular grid ($N=3$).



Step 3: Key-frames Extraction

- Example: HVC1123.mp4 (assembling shelter)

Key-(1,1)



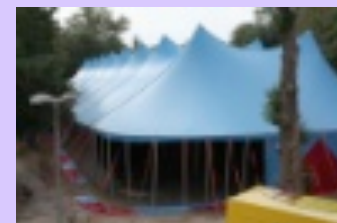
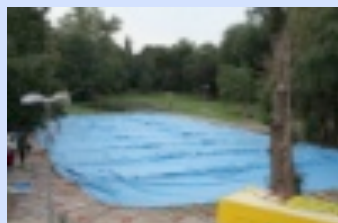
Key-(1,3)



Key-(1,5)



Key-(3,1), (5,1)



Step 3: Key-frames Extraction

- Example: HVC1976.mp4 (butting in run)

Key-(1,1)



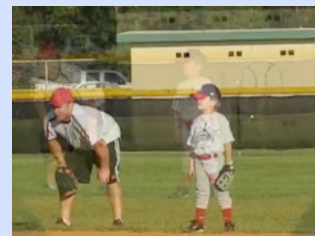
Key-(1,3)



Key-(1,5)



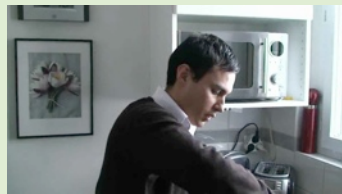
Key-(3,1), (5,1)



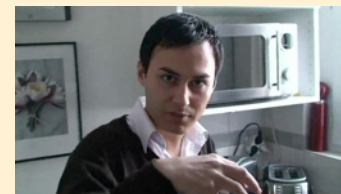
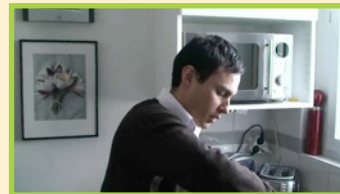
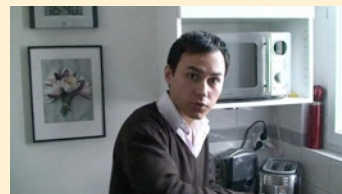
Step 3: Key-frames Extraction

- Example: HVC2795.mp4 (making cake)

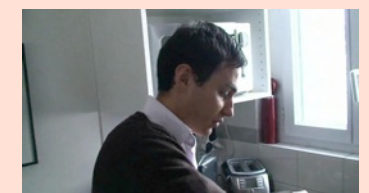
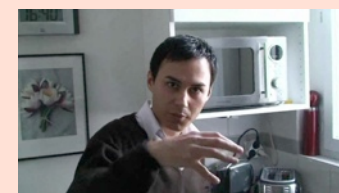
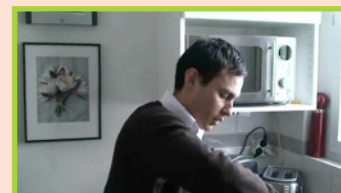
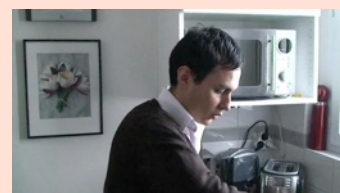
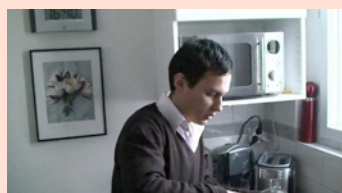
Key-(1,1)



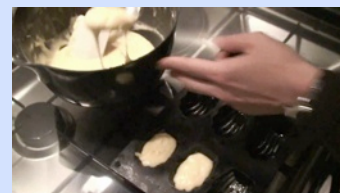
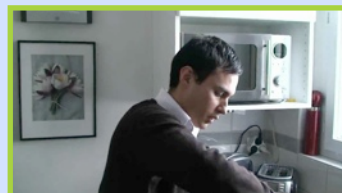
Key-(1,3)



Key-(1,5)



Key-(3,1), (5,1)




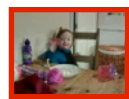
Step 3: Key-frames Extraction

- We think the case that the longest scene contains relevant information for event detection.
- **The Key-(1,N) extracts similar frames ($N > 1$).**
 - In the case, Key-(1,N) will be better. However, otherwise worse.
 - Key-(1,N) will **emphatically** extract relevant or irrelevant information.
- **The Key-(M,1) extracts various frames ($M > 1$).**
 - In the case, Key-(M,1) may not be better than Key-(1,1). However, otherwise will be better.
 - Key-(M,1) will **usually** extract relevant information.

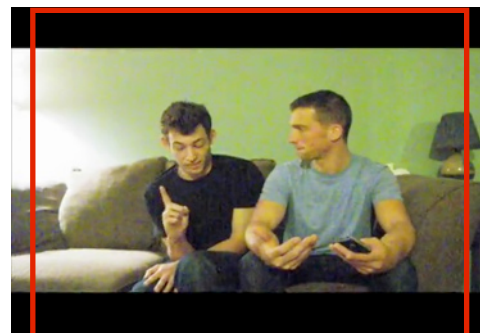
Step 4: Bag-of-words Histogram Construction

- We represent a set of key-frames with a bag-of-words histogram based on SIFT.
 - We trim each of the key-frames into 4:3, and resize it to 320x240 pixels, before SIFT descriptor extraction (Sande, 2010).

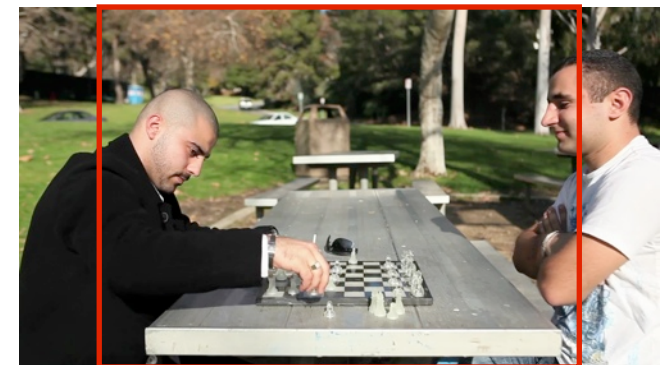

Aspect is 4:3



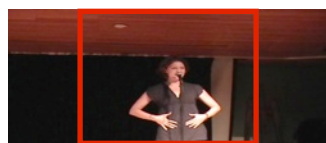
240x180, 4:3



640x432, 4.4:3



1280x720, 16:9



640x272, 21:9



640x480, 4:3



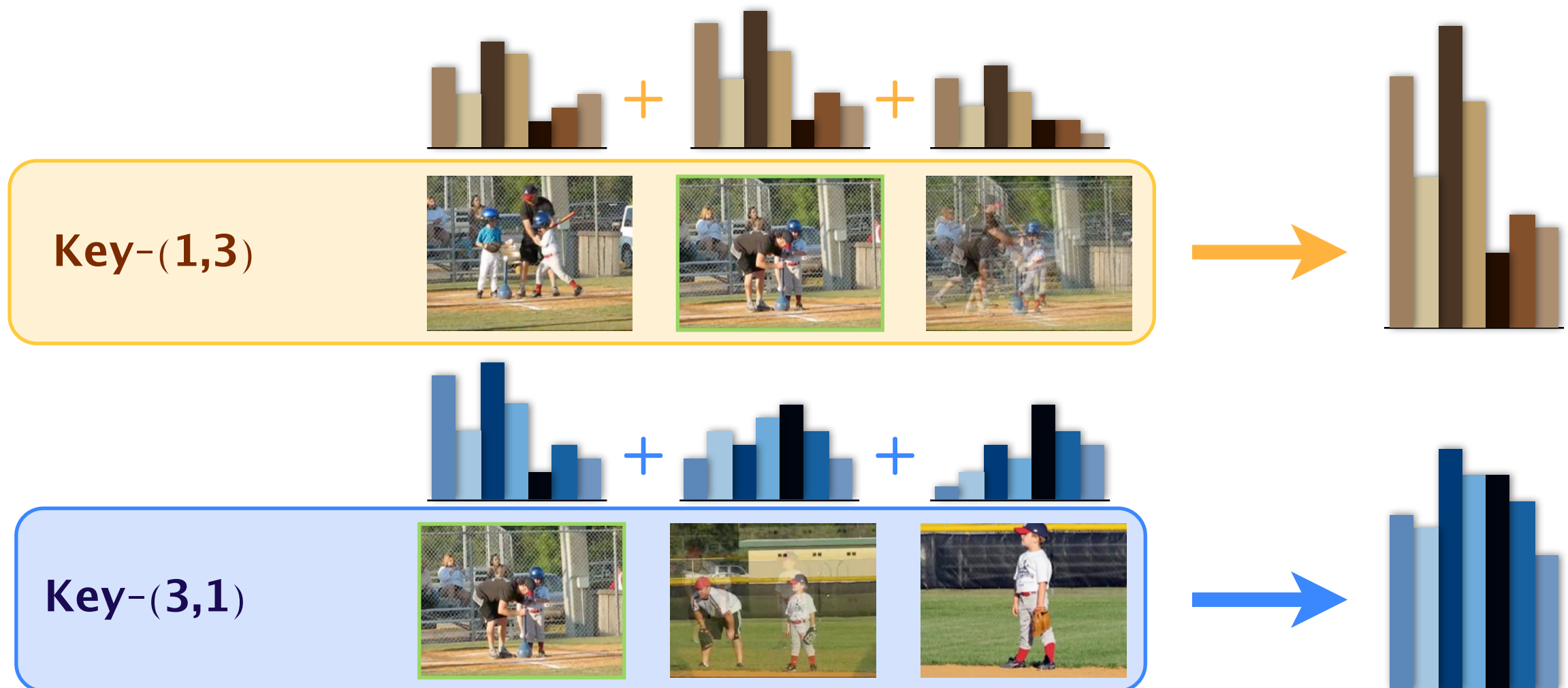
1280x720, 16:9

Step 4: Bag-of-words Histogram Construction

- We use the code-book with 1000 visual words in this bag-of-words procedure.
 - The code-book is created by K-means (of OpenCV 2.1) with all SIFT descriptors from all key-frames over the training set.
 - **Because of memory limitation** of the OpenCV 2.1 and our computer, we **randomly choose** 2^{21} ($\sim 2 \times 10^6$) descriptors if the total number of descriptors is more than 2^{21} .
 - The number of ...
 - the training set: **1744**,
 - key-frames at each video: $M \times N$,
 - SIFT descriptors at each key-frame with resizing: about **1000**.
 - The total number of SIFT descriptors is about $M \times N \times 10^6$.

Step 4: Bag-of-words Histogram Construction

- We represent each video by the sum of bag-of-words histogram in the key-frames.



Step 5: Classification with SVM

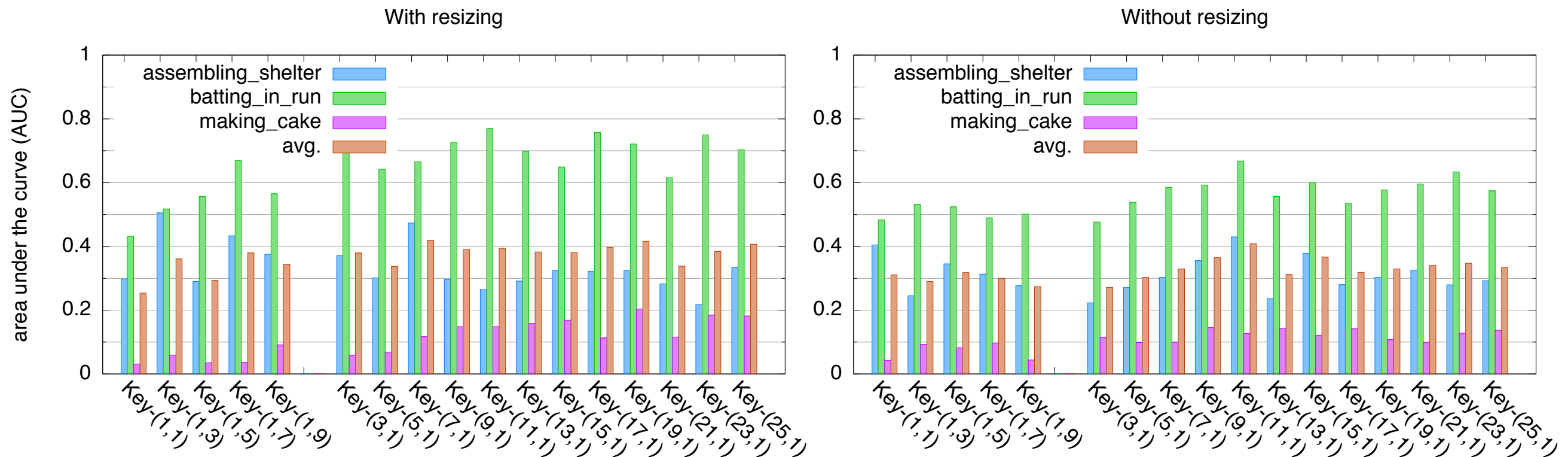
- As we got video features, we execute the learning procedure by support vector machine (SVM).
 - The LIBSVM (Chang and Lin, 2000) is trained with chi-square kernel.
 - The kernel width and the regularization trade-off are optimized by grid search with 5-fold cross validation.

Experimental Result

- **Evaluation by area under the curve (AUC)**
 - The curve consists of the recall (r) vs the precision (p):
 - $r = |A \cap B| / |A|$, $p = |A \cap B| / |B|$.
 - A is the set of true positive event.
 - B is the set of positively detected events.
 - The AUC is calculated by trapezoidal approximation with 500 points over the threshold.

Experimental Result

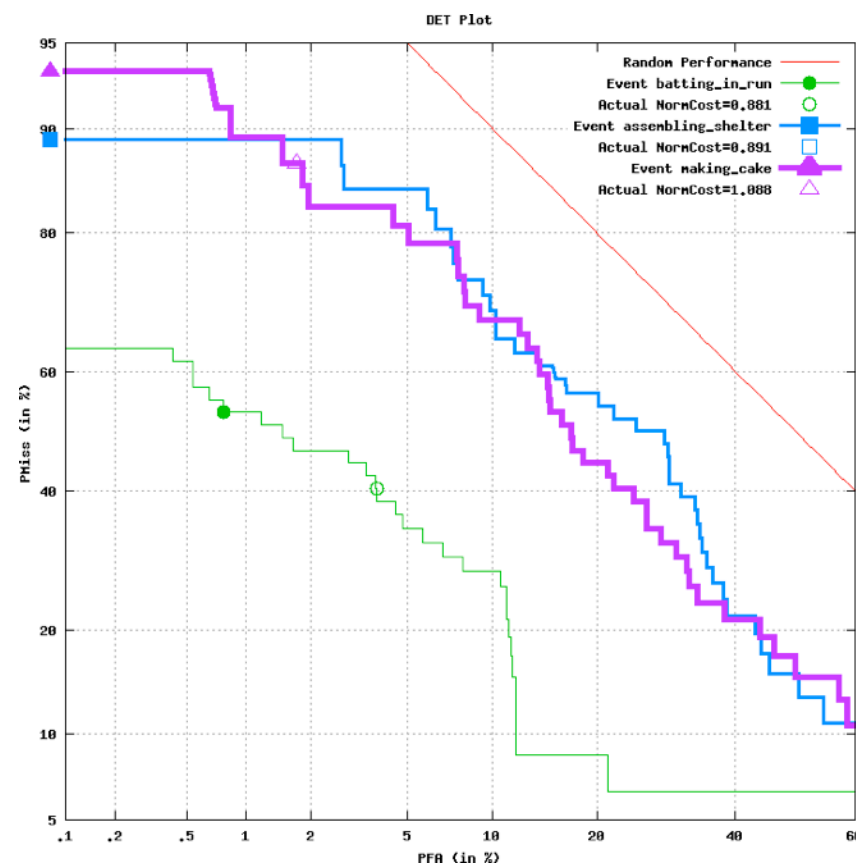
● Evaluation by area under the curve (AUC)



- **Resizing boosts performance.**
- **$M > 1$ (multiple scenes) is better than $M = 1$ (the longest scene).**
- **Key-(7,1) with resizing performs the best in average over all the events in our experiment.**

Our Primary Outputs of TRECVID 2010

- We chose Key-(3,1) without resizing as the primary outputs.
 - There are few results at that time.
 - Then, we thought that results without resizing was better than that with resizing because each image without resizing has more many (relevant) information.



Conclusion

- Our system consists of key-frames extraction based on scene length.
 - The Key- (M,N) is defined to extract N frames from each the M longest scenes as $M*N$ key-frames.
 - The Key- $(M,1)$ is better than Key- $(1,N)$ for $M > 1$ and $N > 1$.
 - Not only the longest scene but also another longer scenes contain relevant information.
 - Resizing before SIFT extraction improves performance.
- We would like to try the Key- (M,N) for $M > 1$ and $N > 1$ and evaluate the best output by TRECVID evaluation.



Thank you.