# Nikon Multimedia Event Detection System

Takeshi Matsuo and Shinichi Nakajima

Optical Research Laboratory, Nikon Corporation

March 1, 2011

**Abstract**

This note presents Nikon's approach to the multimedia event detection (MED) task of TRECVID 2010. We explain the basic concept of our system which detects events by multiple keyframes extraction based on scene length. We describe the algorithm in detail and show experimental results with the MED dataset. Our simple system got the third place among seven teams of participants.

## 1 Basic Concept

We rely on the assumption that a small number of images in a given video contains enough information for event detection. With this assumption, we reduce the event detection task to the classification problem for a set of images, which we call keyframes, by sampling images that potentially represent necessary information.

The keyframes extraction is based on a scene cut detection technique, and the assumption that the longer a scene is, the more relevant information it contains. The classification step employs the bag-of-words (BoW) framework [1] based on the SIFT descriptor [2]. The obtained histogram is fed into the support vector machine (SVM), which is trained over the training dataset for each event category.

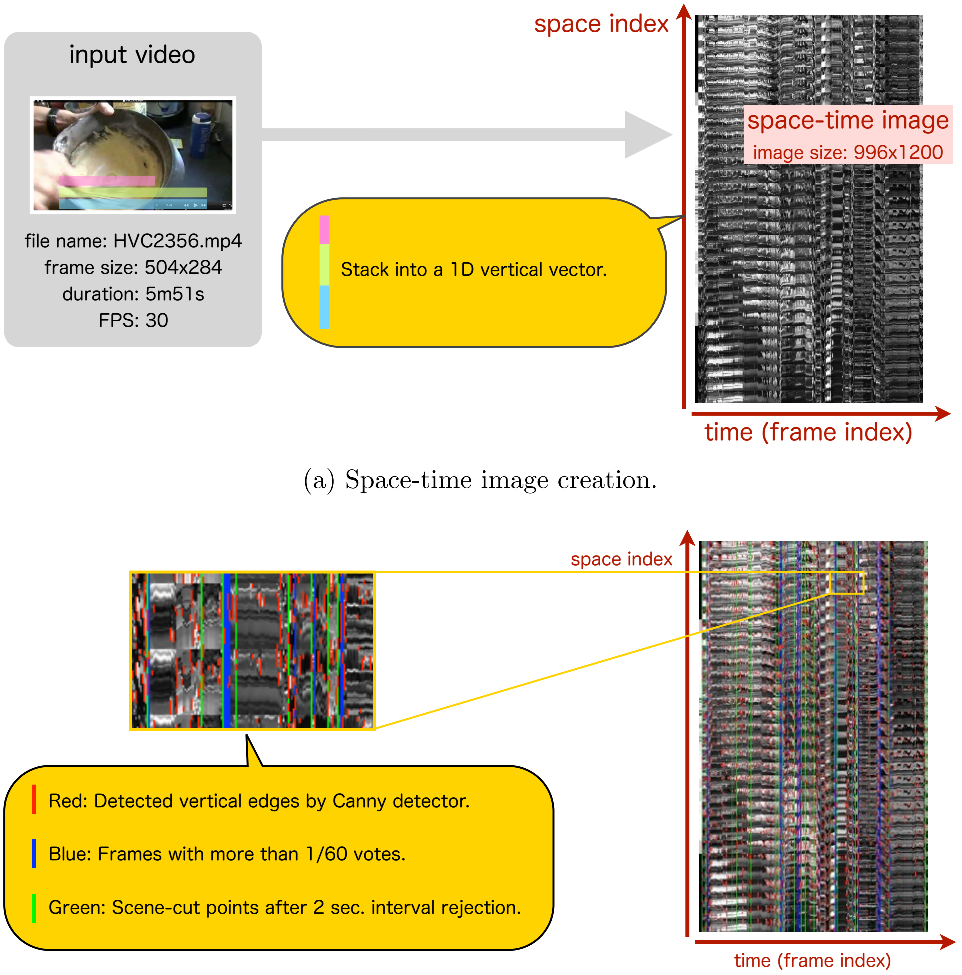
We refer to our system as Nikon multimedia event detection (MED) system in this note.

## 2 Detailed Description of Nikon MED System

Nikon MED system consists of the following five major steps:

1. Create a space-time (ST) image from a video.
2. Perform scene-cut detection based on the ST image.
3. Extract keyframes from each scene.
4. Construct a BoW histogram from the set of keyframes.
5. Classify the histogram by SVM.

The details of each step are described in the following subsections. Our implementation is built on the OpenCV library [3].


input video
file name: HVC2356.mp4
frame size: 504x284
duration: 5m51s
FPS: 30
space index
space-time image
image size: 996x1200
Stack into a 1D vertical vector.
time (frame index)
space index
Red: Detected vertical edges by Canny detector.
Blue: Frames with more than 1/60 votes.
Green: Scene-cut points after 2 sec. interval rejection.
time (frame index)

(a) Space-time image creation.

(b) Scene-cut detection with the space-time image.

**Fig.** 1: Space-time (ST) image creation (a) and scene-cut detection (b). The ST image is created by stacking 2D pixel array of each frame, sampled at regular intervals of a video, into a long 1D vertical vector.

### 2.1 Step 1: Space-time Image Creation

Guimarães et al. proposed a scene-cut method, called a *visual rhythm* [4]. It extracts pixel values on the two diagonal lines from each frame, and stacks them into a 1D vector. The obtained 1D vectors are concatenated to form a 2D *space-time* (ST) image. Scene-cut is performed by applying a vertical edge detector to the ST image.

We follow their spirit, and adopt a more robust way; we utilize all pixel values. A given video is converted to a large ST image by sampling frames at every 0.5 second, and unfolding the 2D structure of images into 1D vector (see Figure 1). Before applying this procedure, we convert a color video into a gray one, trim the frame into 4:3, and resize it to 40 × 30 pixels. Thus, the size of an ST image is

$$\lfloor \mathrm{duration} \cdot \mathrm{FPS} \cdot 0.5 \rfloor \times 1200,$$

where $\lfloor \cdot \rfloor$ denotes the floor of a real number.

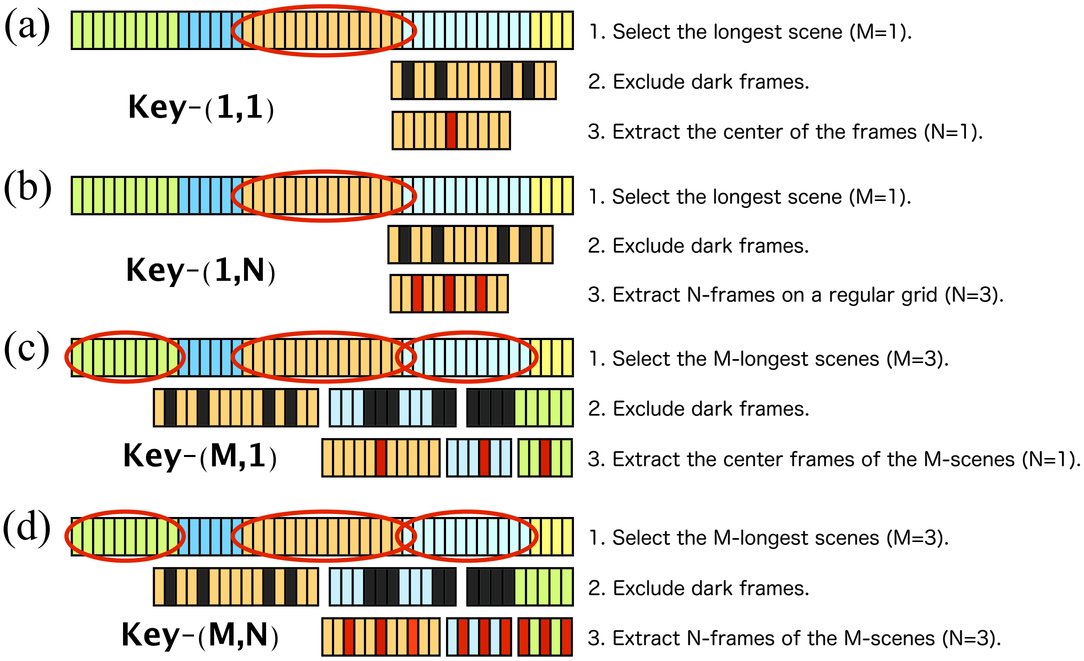

(a)
Key-(1,1)
1. Select the longest scene (M=1).
2. Exclude dark frames.
3. Extract the center of the frames (N=1).
(b)
Key-(1,N)
1. Select the longest scene (M=1).
2. Exclude dark frames.
3. Extract N-frames on a regular grid (N=3).
(c)
Key-(M,1)
1. Select the M-longest scenes (M=3).
2. Exclude dark frames.
3. Extract the center frames of the M-scenes (N=1).
(d)
Key-(M,N)
1. Select the M-longest scenes (M=3).
2. Exclude dark frames.
3. Extract N-frames of the M-scenes (N=3).

**Fig.** 2: Keyframes extraction based on scene-cut.

## 2.2 Step 2: Scene-cut Detection

Scene-cut detection is performed by finding vertical lines in the ST image. We first extract vertical edges by using the Canny detector [5], and then find vertical lines based on the Hough voting, which is in this case simply the sum of the number of the detected vertical edges at each time frame.

We consider the time frames which got more than 20 votes (1/60 of the 1200 pixels lying in the same time frame of the ST image) to be scene-cut points, neglecting the time frames within 2 seconds after the previous scene-cut point. See Figure 1 for illustration.

## 2.3 Step 3: keyframes Extraction

We denote by $Key\text{-}(M, N)$ the keyframes extraction where $N$ keyframes are extracted from each of the $M$ longest scenes. See Figure 2 for examples. We exclude the dark frames whose average brightness is less than 80/256, before we sample $N$ frames from each scene with $l_i/(N+1)$ intervals. Here $l_i$ denotes the length of $i$-th scene. In case that the number of the frames after the dark frame exclusion is less than $N$, supplementary frames are extracted from shorter scenes.

## 2.4 Step 4: Bag-of-words (BoW) Histogram Construction

We represent a set of keyframes with a bag-of-words (BoW) histogram based on SIFT descriptors at interest points.

We trim each of the keyframes into 4:3, and resize it to 320 × 240 pixels, before SIFT descriptor extraction, for which we adopted Sande's software [6]. The code-book with 1000 visual words is created by K-means with all the extracted SIFT descriptors from all the keyframes over the training set, unless the total number of descriptors is more than $n_{\text{lim}} = 2^{22} \approx 4 \times 10^6$. In case that more descriptors are found, we randomly choose $n_{\text{lim}}$ descriptors for memory limitation.

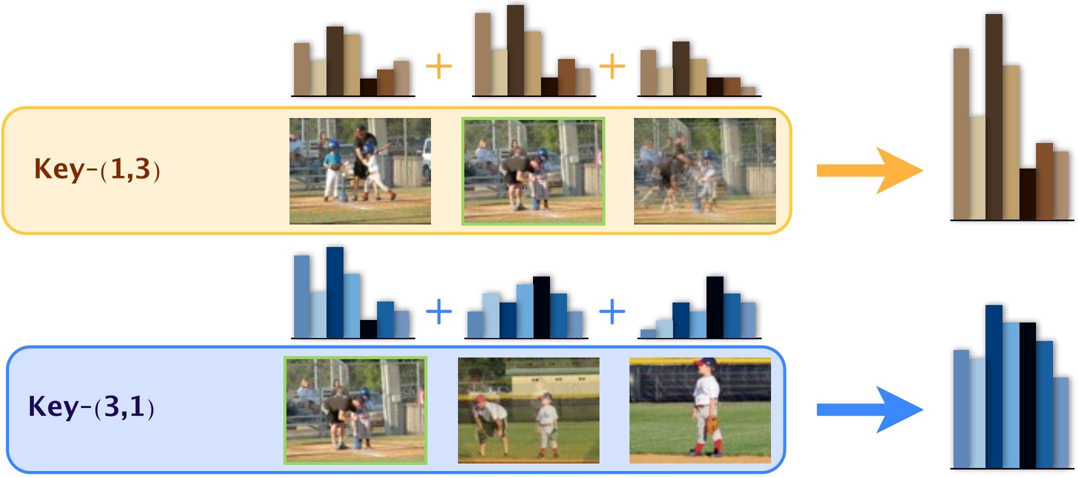


**Fig.** 3: Each video is represented by the sum of the BoW histograms extracted from multiple keyframes.

With the created code-book, each video in the training and the test datasets is represented as a histogram of visual words.

### 2.5 Step 5: Classification with Support Vector Machine (SVM)

The LIBSVM [7] is trained with $\chi^2$ kernel. The kernel width and the regularization trade-off are optimized by grid search with 5-fold cross validation.

## 3 Experimental Result

Here, we report the official evaluation result based on the test dataset. We also conducted performance evaluation of our system with our own criterion, to compare different settings of our methods. Remember that we denote keyframe extraction methods by *Key*-$(M, N)$ with the number $M$ of scenes on which we focus and the number $N$ of frames extracted from each scene.

### 3.1 Official Evaluation Result with Test Dataset of TRECVID 2010

We submitted to TRECVID 2010 the outputs with the following combinations for *Key*-$(M, N)$:

- $(1, 1), (1, 3), (1, 5)$ with resizing,
- $(1, 1), (1, 3), (1, 5), (3, 1), (5, 1), (7, 1)$ without resizing.

Among them, *Key*-$(7, 1)$ *without* resizing performed best, of which the miss detection probabilities over the false alarm probabilities are shown in Fig. 4. It got the third place among all seven participants, although different settings further improve the performance as shown in the next subsection.

### 3.2 Cross Validated Evaluation with Development Dataset

Here, we show additional evaluation results based on the cross validation. We adopted the area under the curve criterion defined below.

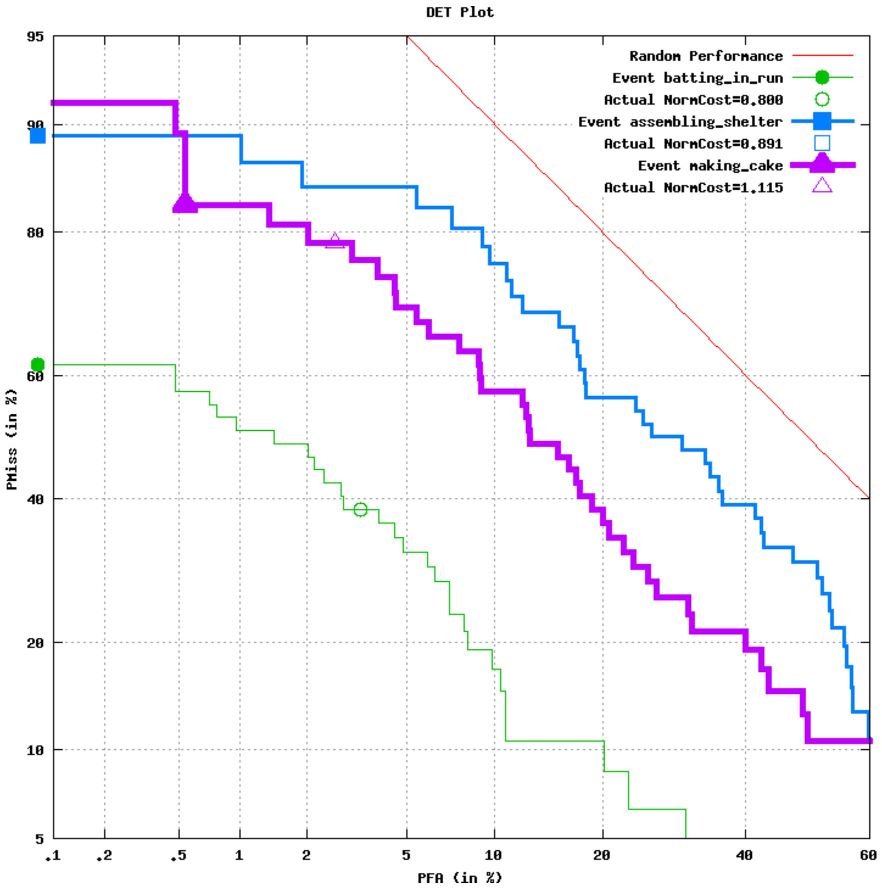


**Fig.** 4: The official evaluation result of *Key*-(7, 1). The horizontal and vertical axes indicate the miss detection and the false alarm probabilities, respectively.

The recall $r$ and the precision $p$ are defined by

$$r \equiv \frac{|A \cap B|}{|A|} \ , \qquad p \equiv \frac{|A \cap B|}{|B|} \ ,$$

where $A$ is the set of true positive events and $B$ is the set of positively detected events. The area under the recall-precision curve (AUC) is calculated by trapezoidal approximation with 500 points over the threshold.

We first evaluated *Key*-$(1, N)$ for $N = 1, 3, 5, 7, 9$ and *Key*-$(M, 1)$ for $M = 3, 5, 7, 9, 11$ with and without resizing for the SIFT descriptor extraction. Figures 5 and 6 show the AUC (with 5-fold cross validation) of our system on the MED development data (1744 video clips) of TRECVID 2010. In Figure 5, the results with (left) and without (right) resizing for *Key*-$(1, N)$ and *Key*-$(M, 1)$ are shown. We see in the figure that our system successfully detected "batting in run" events, while poorly "making cake" events. We also find that that the resizing works positively, and that keyframes extraction from plural scenes $(M > 1)$ tends to perform better than that from only the longest scene $(M = 1)$.

Next, we evaluated *Key*-$(M, N)$ $(M \geq 1,\ N \geq 1)$ with every combination of $M = 1, 2, \ldots, 7$ and $N = 1, 2, \ldots, 7$ with resizing[1]. Fig. 6 shows the results. *Key*-$(6, 5)$ performs the best in average over all the events in our experiment.

[1] In these cases, we had to lower the dark frame threshold to retain a sufficient number of frames.

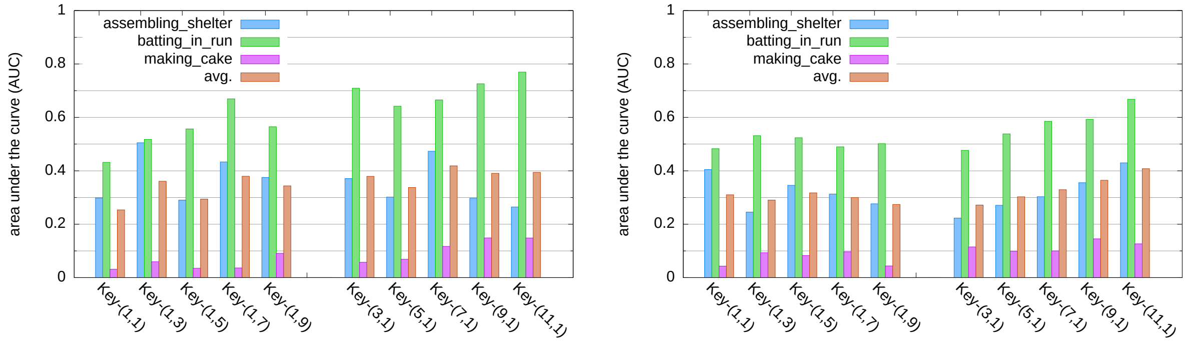


**Fig.** 5: Area under the recall-precision curve (AUC) with (left) and without (right) resizing.

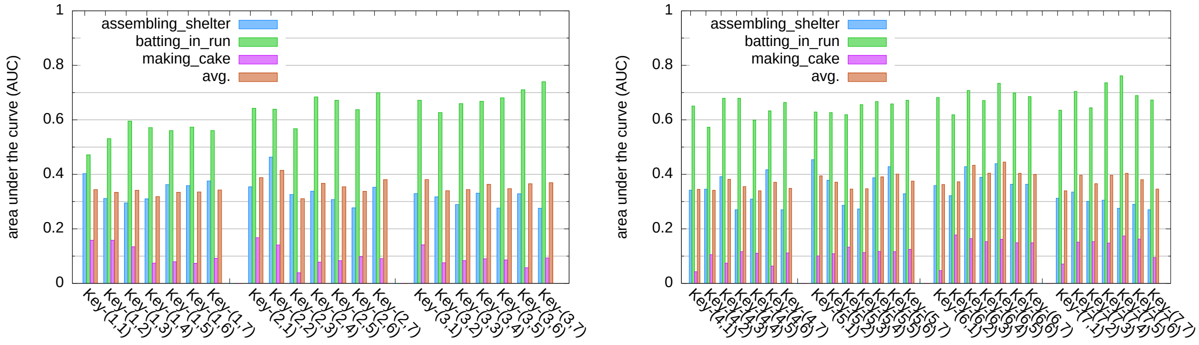


**Fig.** 6: AUC of *Key*-$(M, N)$ with every combination of $M = 1, 2, \ldots, 7$ and $N = 1, 2, \ldots, 7$ with resizing.

Observations are summarized in the following:

- Resizing for SIFT calculation improves the performance.
- Multiple keyframes extraction from plural scenes ($M > 1$) *and* plural frames ($N > 1$) improves performance.

# 4 Summary

We described the Nikon MED system for event detection. It extracts multiple keyframes from longest scenes in a video clip, and classifies the video based on the bag of visual words approach. We reported cross validated evaluation results along with the official evaluation result, and analyzed effect of resizing and dependency on the keyframe extraction strategy. Our simple method got the third place among the seven participants of the TRECVID 2010 multimedia event detection task.

# References

[1] Gabriella Csurka and Christopher R. Dance and Lixin Fan and Jutta Willamowski and Cédric Bray, "Visual categorization with Bags of keypoints", Proc. of IEEE Computer Vision and Pattern

Recognition, pp.59–74, 2004.

[2] David Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”. *International Journal of Computer Vision*, 60(2): 91–1110, 2004. http://www.cs.ubc.ca/~lowe/keypoints/.

[3] OpenCV. http://opencv.willowgarage.com/wiki/Welcome

[4] Silvio Jamil Ferzoli Guimarães, Michel Couprie, Arnaldo de Albuquerque Araújo, and Neucimar Jerônimo Leite. “Video segmentation based on 2D image analysis”, *Pattern Recognition Letters*, 24(7): 947–957, 2003.

[5] Canny, J., “A Computational Approach to Edge Detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6): 679–698, 1986.

[6] Koen E. A. van de Sande and Theo Gevers and Cees G. M. Shoek, “Evaluating Color Descriptors for Object and Scene Recognition”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010

[7] Chih-Chung Chang and Chih-Jen Lin, LIBSVM. http://www.csie.ntu.edu.tw/~cjlin/libsvm/,