Event Detection by Velocity Pyramid

Zhuolin Liang, Nakamasa Inoue, and Koichi Shinoda

Department of Computer Science, Tokyo Institute of Technology.
{zhuolin,inoue}@ks.cs.titech.ac.jp,{shinoda}@cs.titech.ac.jp

Abstract. In this paper, we propose velocity pyramid for multimedia event detection. Recently, spatial pyramid matching is proposed to introduce coarse geometric information into Bag of Features framework, and is effective for static image recognition and detection. In video, not only spatial information but also temporal information, which represents its dynamic nature, is important. In order to fully utilize it, we propose velocity pyramid where video frames are divided into motional sub-regions. Our method is effective for detecting events characterized by their temporal patterns. Experiment on the dataset of MED (Multimedia Event Detection) has shown 10% improvement of performance by velocity pyramid than without this method. Further, when combined with spatial pyramid, velocity pyramid provides an extra 3% gains to the detection result.

Keywords: Event detection, spatial pyramid, velocity pyramid, GMM supervectors

1 Introduction

With the development of various web services, the amount of videos available on the Internet is growing exponentially. These open source videos usually have a large variety of contents composed by scenes, objects, motions and audio cues. How to search videos effectively becomes a heated issue. Event detection in large scale unconstrained videos is a research topic towards promoting understanding of video contents. Here, an event is defined as a complex activity occurring at a specific place and time which involves people interacting with other people and/or object(s) [1]. For example, the event of “Birthday party” may be indicated by the observation of the following aspects: a scene as indoor, an object like a birthday cake, activity of opening a gift, and even audio cues such as people cheering. Event detection is thus more challenging than object detection in a still image or activity detection in a video.

A typical flow to detect an event from a video is described as follows: 1) feature extraction, 2) feature encoding, 3) recognition. In order to capture different characteristics of an event, researchers tend to integrate multiple kinds of features into an event detection framework [4]. For example, GIST [5] features can be used for capturing global scene characteristics of an image; HOG, SIFT, HOG3D can capture object appearance information; STIP [15], Dense Trajectory [16], MoSIFT [18] are able to encode temporal evolution. After feature extraction, features are often encoded into fixed-length histogram. Usually a standard Bag-of-Words approach is used for this encoding process. Soft encoding methods such as Fisher vector encoding [9], GMM supervector encoding [12] have also been applied. After obtaining the encoded vectors, classifies such as SVMs are used for detection with early or late fusion of features.

Spatial pyramid [8], which is originally proposed for scene recognition of static images, is often used as features in event detection. Inspired by this spatial pyramid, we propose velocity pyramid in this paper. Instead of utilizing pyramid structure which represents spatial information, we construct a pyramid structure which represents dynamic nature of video. We first divide video into several components using motion information. Then, for each motion component, we model the distribution of features, which is expected to follow a fixed pattern. For example, in the event *parade*, people are likely to move in the horizontal direction than the vertical direction. In the event *repairing_an_appliance*, people's hands tend to move in every direction.

We construct a framework for event detection with GMM supervectors. The GMM supervector is used for feature encoding, which has been applied to event detection and outperforms Bag-of-Words models [12]. We verify the system's effectiveness on the challenging dataset of Multimedia Event Detection (MED) task of TRECVID [1]. In this dataset, multiple conditions of scenes, objects, motion patterns exist. We will show that the velocity pyramid can capture the rough dynamic information of the video. Furthermore, when combined with spatial pyramid, the performance of the system is further improved.

The rest of this paper is organized as follows: Section 2 describes related works for event detection; Section 3 focuses on the proposed method of velocity pyramid; Section 4 introduces the steps for constructing the detection system; Section 5 describes the evaluation dataset, evaluation measures, and experiment results; Finally Section 6 gives a conclusion.

2 Related Work

As video has its own nature as a spatial and temporal sequence, various spatial-temporal features have been applied to event detection in video. STIP [15] selects spatial-temporal interest points by detecting 2D corners with rapid velocity change. MoSIFT [18] finds interest points that have both discriminative appearance and sufficient amount of motion. Dense Trajectory [16] represents videos as a set of trajectories obtained from tracked points. Motion Histogram [17] integrates appearance and motion by calculating a motion histogram for each visual word. Relative motion histogram is also computed for each visual word pair between every two frames. However, all these features explore deeply into the internal structure of an video volume by detecting local maximum or by tracking objects, thus have the problem of a heavy computation cost and face the storage problem especially when applied to large-scale unconstrained videos. Our purpose is to construct a structure that can encode both spatial and dynamic information effectively and efficiently.

Spatial pyramid matching [8] is proposed to introduce coarse geometrical information into Bag-of-Words approach. Several methods to construct the spatial pyramid have been explored. In [13], both feature specific and event specific tiling has been examined. They show that for all kinds of features, including both appearance and motion, spatial pyramid improves the detection performance. Also in [14], 12 different Regions of Interest (ROI) are defined and their contributions to the detection cost are evaluated separately, In [10] soft tiling is proposed where a sample point is assigned to several tiles and gives significant improvement.

Feature encoding methods used in event detection include Bag-of-Words [2], GMM supervectors [12], Fisher Vectors [9] etc. GMM models a set of features as mixture of Gaussians with different means and covariances. Fisher vector consists of the first and second order differences to the cluster center. The latter two are based on generative probability models and use soft assignment to mitigate the influence of code word miss assignment.

Here we focus on how to integrate appearance and motion features effectively. Since a global motion histogram and appearance features from a certain frame may be independent with each other, it's better to calculate motion histogram for each object, e.g. tree, people, etc. From another view, appearance features with similar motion should belong to the same feature set characterized by the motion. This idea is similar to spatial pyramid representation in images, which models feature distribution in each spatial subregion. Instead of spatial information, we utilize the dynamic features of video. We build a velocity pyramid which captures coarse dynamic information of appearances. This method is also efficient compared to spatial-temporal interest point approaches because it does not need tracking or local maximum exploration.

3 Velocity Pyramid

An illustration of the procedure to construct the velocity pyramid subregions is given in Fig. 1. First, extract appearance features and motion vectors. Second, quantize motion vectors into motion bins. Third, calculate an encoded appearance histogram for each motion bin. Lastly, concatenate the encoded histograms to form an input vector for a classifier.

Appearance features. The pyramid representation can be applied to any kind of low level visual features, including densely of sparsely sampled features, e.g. SIFT, HOG, STIP, etc. Here we use a single type of features for simplicity of explanation. For one frame which consists of n feature samples, the set of low level features can be represented as

$$X = \{\mathbf{x}_i\}_{i=1}^{n}, \tag{1}$$

where $\mathbf{x}_i$ is the ith sample.

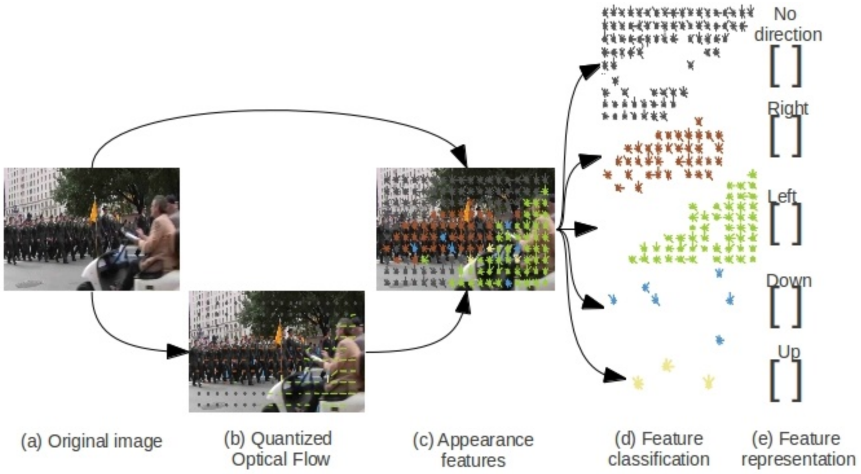
No direction
Right
Left
Down
Up
(a) Original image
(b) Quantized Optical Flow
(c) Appearance features
(d) Feature classification
(e) Feature representation

Fig. 1. Subregion generation in velocity pyramid. (a) Original image which shows people who are parading right meet a car passing left. (b) Optical flows calculated from two adjacent frames. The flow vectors are quantized into 4 directions, and each color represents one direction. Gray dots have 0 motions. (c) Appearance features. In this figure, we use dense oriented gradient based features. (d) Partition result of (c) according to the quantization result in (b). (e) Feature representation, e.g. histograms from the Bag-of-Words model.

Motion vectors. Motion information is captured by optical flow computed by $Farnebäck$ algorithm [20]. We calculate velocity vectors for the same coordinates as in X. So the set of optical flows can be expressed as

$$V = \{\mathbf{v}_i\}_{i=1}^{n}, \tag{2}$$

where $\mathbf{v}_i$ is a velocity vector of the ith sample.

Motion Quantization. In order to relate appearance information with motion information, we assign each feature vector $\mathbf{x}$ to a certain motion orientation bin. For each non-zero velocity vector $\mathbf{v}$, let us introduce a P-dimensional orientation vector $\mathbf{o}$, in which each element is a binary variable. The appearance features that have motion are classified into P categories and in each category one of the dimensions $o(p)$ is equal to 1 and other dimensions are equal to 0. In other words, $o(p) \in \{0, 1\}$ and $\sum_p o(p) = 1$. The value of $\mathbf{o}$ is determined by quatizing a motion vector

$$o(p) = 1 \quad \text{if } \frac{2\pi}{P}p \leq \theta < \frac{2\pi}{P}(p+1), \;\; p \in \{0, ..., P-1\}, \tag{3}$$

where θ is the orientation of optical flow vector $\mathbf{v}$ ranging from 0 to 2π. This motion bin vector is calculated for each sample. Consequently, for each frame

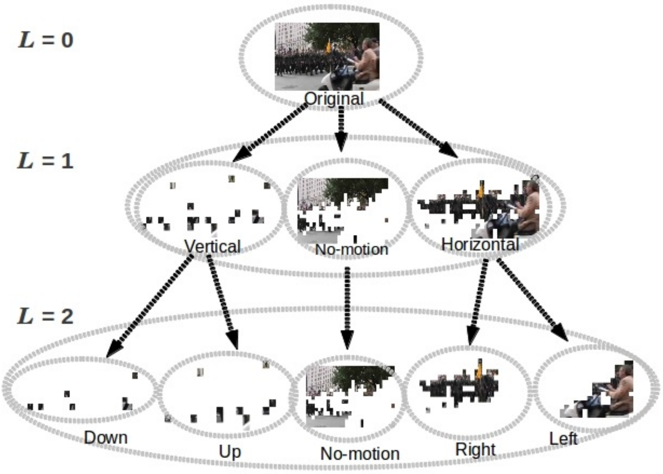

Fig. 2. Velocity Pyramid. In $L = 1$ and $L = 2$, the horizontal part is mainly comprised of a passing parade and a car; the vertical part mainly comes from people's legs; and no motion part mainly corresponds to backgrounds.

we have a set of orientation vectors which can be explained as

$$O = \{\mathbf{o}_i\}_{i=1}^{n}, \tag{4}$$

Feature classification. Features are classified according to their orientation quantization result

$$X_p = \{\mathbf{x}_i | \mathbf{v}_i \neq (0, 0) \text{ and } o_i(p) = 1\} \ (p \in \{0, ..., P - 1\}), \tag{5}$$

$$X_P = \{\mathbf{x}_i | \mathbf{v}_i = (0, 0)\}. \tag{6}$$

where P is the number of quantized orientation bins, and $X_p,\ p \in \{0, 1, ..., P\}$ is a feature set. X_P represents features from an extra zero bin which is more likely to come from background features. The total number of pyramid components is $P + 1$, and the relationship among sets satisfies $X = X_0 \cup X_1 \cup ... \cup X_P$.

Feature encoding. For each set of features $X_p,\ p \in \{0, 1, ..., P\}$, the corresponding histogram of encoded features is calculated by the encoding method in Subsection 4.2. Each histogram is a component of the velocity pyramid. And a final representation of the video is the concatenation of all the histograms.

Pyramid level. Let us define L as the level of velocity pyramid, so the number of non-zero motion components is given by $P = 2^L$. $L = 0$ represents the original appearance feature; $L = 1$ divides these features into horizontal and vertical components; $L = 2$ includes 4 equally quantized orientations including left, right, up, down; etc. We illustrate velocity pyramid in Fig. 2.

4 System construction

4.1 Spatial and velocity pyramid

We integrate spatial pyramid with velocity pyramid in order to capture a coarse geometrical and dynamic information simultaneously. In spatial pyramid, pyramid components are from tiles made by dividing images into finer subregions, and histograms of local features are calculated and matched for the resulted subregions.

We make the spatial pyramid by dividing video clips into several sub-volumes. Tiling is an important factor in spatial techniques. In [7], 2×2 grids shows good result. They also evaluate several other spatio-temporal grids for action recognition, and find that a partition of horizontal 3×1 grids is the optimal for capturing layout of natural scenes. We also use these partitions when implementing spatial pyramid in the system. The same set of features as used in velocity pyramid are encoded into a histogram for each volume separately. The resulted histograms for all volumes are then concatenated into a single vector as the representation of the video.

4.2 GMM supervectors

Gaussian mixture model (GMM) and SVMs was proposed in the context of speaker verification, and applied successfully to multimedia event detection [12]. It outperforms Bag-of-Words, because it realizes soft assignment by considering covariance information.

Given a set of features $X = \{x_i\}_{i=1}^{n}$, the probability distribution function of X conditioned on a Gaussian Mixture Model is given by

$$p(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{7}$$

where x represents a d dimensional feature vector, K is number of Gaussian mixtures, $\theta = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ are parameters for Gaussian functions, w_k, μ_k, Σ_k are the weight, mean, and covariance of the kth Gaussian probability distribution function $\mathcal{N}(\cdot|\mu_k, \Sigma_k)$.

It is difficult to precisely estimate the parameter θ for one video since its number of feature samples is quite small. In this case, Maximum a Posteriori (MAP) adaptation technique is utilized, because it performs well with a small amount of data [11]. In MAP adaptation the priori knowledge comes form an universal background model (UBM), a GMM whose parameters are estimated by EM algorithm from all training data. After obtaining the UBM, GMM parameters for each video are estimated by MAP adaptation in the following way

$$\hat{\mu_k} = \frac{\tau \mu_k^{(U)} + \sum_{i=1}^{n} c_{ik} x_i}{\tau + \sum_{i=1}^{n} c_{ik}}, \tag{8}$$

$$c_{ik} = \frac{w_k^{(U)} \mathcal{N}(x_i|\mu_k^{(U)}, \Sigma_k^{(U)})}{\sum_{k=1}^{K} w_k^{(U)} \mathcal{N}(x_i|\mu_k^{(U)}, \Sigma_k^{(U)})}, \tag{9}$$

where $\theta^{(U)} = \{w_k^{(U)}, \mu_k^{(U)}, \Sigma_k^{(U)}\}_{k=1}^K$ is a set of parameters for Gaussian components in UBM. c_{ik} is the confidence of Gaussian mixture k for observing feature point x_i. τ is a pre-defined hyper parameter. Note that here only mean vectors for each video are adapted. Next, the signature of a video clip is represented as a concatenation of the K adapted mean vectors

$$\Phi(X) = (\tilde{\mu_1}^T \tilde{\mu_2}^T ... \tilde{\mu_K}^T)^T, \tilde{\mu_k} = \sqrt{w_k^{(U)}} (\Sigma_k^{(U)})^{-\frac{1}{2}} \hat{\mu_k}. \tag{10}$$

In the formula above, each mean vector is normalized by the corresponding weight $w_k^{(U)}$ and covariance $\Sigma_k^{(U)}$.

4.3 SVM detection

GMM supervectors are the inputs to Support Vector Machines (SVMs) with RBF kernel $k(X, X^{'})$ for detection:

$$k(X, X^{'}) = \exp(-\gamma \sum_{p=0}^{P} \|\Phi(X_p) - \Phi(X_p^{'})\|_2^2), \tag{11}$$

where γ is set to be the inverted averaged distance between GMM supervectors. The detection confidence of SVM is given by

$$f(X) = \sum_{l=1}^{L} a^{(l)} k(X, X^{(l)}) + b. \tag{12}$$

where $X^{(l)}$ is a set of features from a training video, L is the number of support vectors, $a^{(l)}$ and b are parameters obtained during SVM training.

4.4 Fusion of spatial and velocity pyramid

Spatial and temporal information are integrated in a late fusion manner. Suppose $f_E^{(sp)}$ and $f_E^{(vp)}$ are detection scores of spatial and velocity pyramid, respectively. The final confidence of one video for event E is:

$$S_E = a * f_E^{(sp)}(X) + (1 - a) * f_E^{(vp)}(X). \tag{13}$$

where a $(0 \leq a \leq 1)$, is the fusion weight, which is determined by cross validation for each E during training.

5 Experiment

In this part, we will first introduce the dataset and metric for evaluation, then show experiment result with respect to: effectiveness of spatial-velocity pyramid and the influence of pyramid levels.

5.1 Dataset

The dataset we used is a part of HAVIC data collected by Linguistic Data Consortium, including MED10 data set, MED11 development and test data set [1]. The videos are user-generated videos posted to various Internet video hosting sites. These datasets are for system development and evaluation in the Multimedia Event Detection (MED) task of TRECVID workshop, aiming at permitting users to define their own complex events and quickly and accurately searching large collection of multimedia clips. In MED, an event is defined in an **event kit** including event name, definition, explication, description, and example video clips.

The MED10 dataset is a collection of 3,468 videos (∼115h), of which 1,744 videos are for training and 1,724 for testing. There are three events: *Assembling_shelter*, *Batting_in_run*, and *Making_cake*. Approximately 50 positive clips are provided for each event to train its event model. The MED11 dataset is a collection of 44,904 videos (∼1406h) including 10 events, in which 13,083 videos are for training, and 31,821 for testing. Each event has 80-230 positive samples in its event kit. These datasets are challenging due to the following reasons: user-generated videos diverse in resolution, length, and quality; unconstrained videos usually have unavoidable camera motions (e.g. *Getting_a_vehicle_unstuck*), clustered background (e.g. *Parade*), various viewing angles, etc.

5.2 Evaluation criterion

The evaluation criterion is Normalized Detection Cost (NDC) which is used in the TRECVID MED task. This criterion is the linear combination of two kinds of error rates: missed detection rate (P_{MD}) and false alarming rate (P_{FA}). When applying a certain threshold T to the detection scores, the calculation of NDC, P_{MD}, and P_{FA} are defined by the formulas below

$$NDC(T) = w_{MD}P_{MD}(T) + w_{FA}P_{FA}, \tag{14}$$

$$P_{MD}(T) = N_{MD}(T)/N_{pos}, \tag{15}$$

$$P_{FA}(T) = N_{FA}(T)/N_{neg}, \tag{16}$$

where w_{MD} and w_{FA} are the weighting factors for the two error rates respectively. In MED task, $w_{MD} = 1.0$ and $w_{FA} = 12.4875$. N_{MD} is the number of videos that are real positives but have a confidence score lower than the detection threshold. Oppositely, N_{FA} is the number of videos that are real negatives, but assigned a higher confidence score than the detection threshold. N_{pos} and N_{neg} are total numbers of positive and negative videos in test set, respectively.

We use a posterior way to tune the detection threshold T to find the minimum NDC (MNDC). We will report MNDC for each event separately as well as the mean MNDC across all events,

$$MNDC = \min_{T} NDC(T). \tag{17}$$

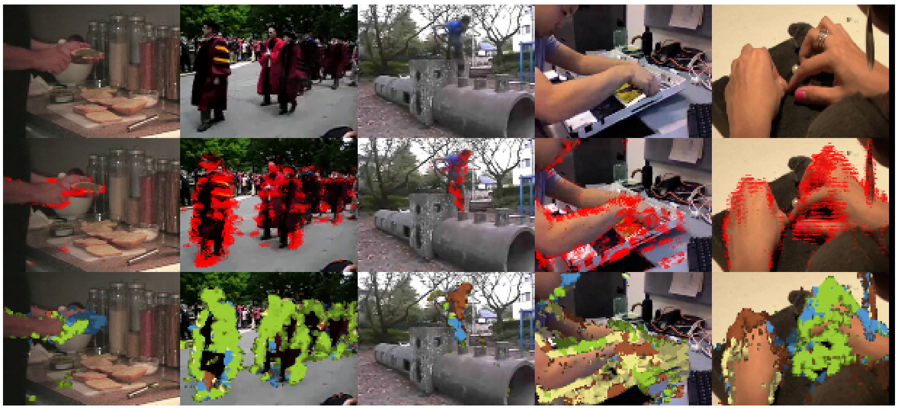

Fig. 3. Clip examples with significant improvements by velocity pyramid than without it. This figure lists 5 events in columns. For each column, the first row is original frame, the second row shows optical flow frame, and the third row is the result of flow quantization. In the third row, color indicates flow orientations, and saturation indicates flow magnitudes.

5.3 Performance of Velocity Pyramid

In this section, we compared the performance of the original representation without pyramid, spatial pyramid, and velocity pyramid. We use dense HOG feature [6] since dense features have shown better results than features from sparse interest points in several applications.They are sampled from 4×4 pixels grid, counted in a patch of 20×20 image pixels divided by a 2×2 window. For each window we generate an 8-bin histogram, summing up to a 32-dimensional HOG feature. PCA is applied to it without reducing its dimension. The spatial pyramid follows a pattern of *original*, 2×2, and 3×1, totally 8 spatial components. All pyramid techniques utilize the same set of HOG features.

Evaluation on MED10 dataset For MED10 data, we use a single level of velocity pyramid $L = 2$, totally 6 velocity components including the original one. The result is shown in Table 1. MNDC for spatial pyramid, 0.635, is better than the original HOG features, 0.661. Velocity pyramid outperforms spatial pyramid for 2 out of 3 events. Furthermore, the combination of spatial and velocity pyramid achieves the best MNDC, 0.607. For a motion intensive event *Batting_in_run*, velocity pyramid is especially effective. For a complex event *Making_cake*, which is comprised by multiple objects and activities, spatial pyramid and velocity pyramid collaborate well to obtain large gain from original HOG features.

Evaluation on MED11 dataset MED11 dataset containing 10 events is a more challenging dataset. Since we get a better result when combining $L = 1$ and $L = 2$ (see the next subsection), we apply this setting to the MED11 data.

Event	HOG original	HOG SP	HOG VP	HOG SP&VP
Assembling_shelter	0.768	0.772	0.776	0.751
Batting_in_run	0.453	0.446	0.434	0.442
Making_cake	0.761	0.688	0.642	0.628
Mean	0.661	0.635	0.617	0.607

Table 1. MNDC of HOG original, spatial pyramid, velocity pyramid for 3 events in MED10. SP = spatial pyramid, VP = velocity pyramid.

Event	HOG original	HOG SP	HOG VP	HOG SP&VP
Birthday_party	0.860	0.749	0.762	0.739
Changing_a_vehicle_tire	0.698	0.600	0.598	0.573
Flash_mob_gathering	0.412	0.364	0.366	0.362
Getting_a_vehicle_unstuck	0.556	0.523	0.597	0.512
Grooming_an_animal	0.774	0.712	0.746	0.705
Making_a_sandwich	0.856	0.753	0.768	0.761
Parade	0.698	0.607	0.599	0.594
Parkour	0.574	0.484	0.498	0.486
Repairing_an_appliance	0.645	0.598	0.518	0.519
Working_on_a_sewing_project	0.810	0.783	0.752	0.746
Mean	0.688	0.617	0.620	0.600

Table 2. MNDC of HOG original, spatial pyramid, velocity pyramid for 10 events in MED11. SP = spatial pyramid, VP = velocity pyramid.

The evaluation result is shown in Table 2. The spatial-velocity pyramid outperforms spatial pyramid in 8 out of 10 events, which further verifies the effectiveness of the method. The velocity pyramid method obtained a competitive result with spatial pyramid; for 4 out of 10 events it outperforms spatial pyramid. Specifically, we observed a significant improvement on *Repairing_an_appliance* and *Working_on_a_sewing_project*. In these two events, motions are clearer than the others. This may provide a large gain in the detection result. These two events also have targets whose motions are widely spread across one frame. In this case, velocity pyramid performs better than spatial pyramid. Meanwhile, velocity pyramid is not effective for some events. These include *Parkour* where the motion area is small, and *Getting_a_vehicle_unstuck* which has hand-held camera motions that hides real object motion. However, the overall performance is improved by utilizing spatial-temporal information. Fig. 3 shows some examples in which velocity pyramid has significant improvements than without motion information.

The best performance for MED task in TRECVID 2012 was achieved by AXES team. They reported a MNDC of 0.411 on MED11 dataset [3], while our best performance on the same dataset is a MNDC of 0.495. The higher performance of AXES team may owe to the robustness of Motion Boundary Histogram (MBH) against camera motion.

To reduce both computation and storage costs, we extract features every 60 frames (2 seconds). In our experiment, velocity pyramid's computation cost is

Event	$L=0$	$L=1$	$L=2$	$L=1{,}2$
Assembling_shelter	0.768	**0.695**	0.776	0.744
Batting_in_run	0.453	0.448	0.434	**0.410**
Making_cake	0.761	0.684	0.642	**0.628**
Mean	0.661	0.609	0.617	**0.594**

Table 3. Influence of pyramid levels on MNDC. $L = 0$ is the detection result by the original appearance features; $L = 1$ means horizontal, vertical and 0-bin; $L = 2$ means 4 equally quantized orientation plus 0-bin; $L = 1{,}2$ means an early fusion of supervectors from different levels. Note that in $L = 1$ and 2, supervector from $L = 0$ is also used. The values in bold are the best MNDC score for each event.

20% of STIP, whose cost is reported to be much less than Motion SIFT [19]. In addition, the metadata size of velocity pyramid is 15% of dense trajectory.

5.4 Influence of pyramid levels

The effects of different pyramid levels L are evaluated in this subsection. The result is shown in table 3. When $L = 1$, which means the components only include 0-bin, horizontal and vertical, the result is surprisingly good. By this division, we can have a rough separation of foreground and background. The combination of different pyramid levels of $L = 1{,}2$ is better than a single pyramid level $L = 1$ or $L = 2$.

6 Conclusion

In this paper, we propose velocity pyramid as an image representation for multimedia event detection. While spatial pyramid divides appearance in a 2D spatial domain, velocity pyramid models appearance in the motion. The resulted velocity pyramid together with a representation by Gaussian Mixture Models is applied to the challenging MED dataset and shows effectiveness for detecting events. In the case of MED11 dataset, MNDC of 0.620 is obtained by velocity pyramid. Further, the MNDC score is reduced to 0.600 when velocity pyramid is combined with spatial pyramid. Future work includes the cancelling of camera motions in unconstrained videos, such as using the camera motion canceled feature MBH instead of HOG. In addition, since currently only orientation information is taken into consideration, we plan to use not only orientation information, but also flow magnitude information in velocity pyramid.

Acknowledgments. Thanks for Canon Incorporation for providing with computation resources and technical supports.

References

1. 2013 TRECVID Multimedia Event Detection Track, `http://www.nist.gov/itl/iad/mig/med13.cfm`

2. Y. G. Jiang, X. Zeng, G. Ye, et al.: Columbia-UCF TRECVID2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching. In Proc. of TRECVID Workshop (2010).
3. R. Aly, K. McGuinness, et al.: AXES at TRECVid 2012. In Proc. of TRECVID Workshop (2012).
4. L. Jiang, Alexander G. Hauptmann, G. Xiang: Leveraging High-level and Low-level Features for Multimedia Event Detection. In: ACM Multimedia 12, pp. 449–458 (2012).
5. A. Torralba, A. Oliva: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV, vol. 42(3), pp. 145–175 (2001).
6. N. Dalal, B. Triggs, C. Schmid: Human Detection Using Oriented Histograms of Flow and Appearance. In: Proc. ECCV, pp. 428–441, Austria (2006).
7. I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld: Learning Realistic Human Actions from Movies. In Proc. CVPR, pp. 1–8 (2008).
8. S. Lazebnik, C. Schmid, J. Ponce: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In Proc. CVPR, pp. 2169–2178 (2006).
9. C. Sun, R. Nevatia: Large-scale Web Video Event Classification by use of Fisher Vectors. 2013 IEEE Workshop on Application of Computer Vision, pp. 15–22 (2013).
10. V. Viitaniemi, J. Laaksonen: Spatial extensions to bag of visual words. In: Proc. CIVR, ACM (2009).
11. N. Inoue, K. Shinoda: A Fast and Accurate Video Semantic-Indexing System Using Fast MAP Adaptation and GMM Supervectors. IEEE Transactions on Multimedia, vol.14, no.4-2, pp. 1196–1205 (2012).
12. Y. Kamishima, N. Inoue, K. Shinoda, S. Sato: Multimedia Event Detection Using GMM Supervectors and SVMs. In Proc. ICIP, pp. 3089–3092, Florida (2012).
13. S. Yu, Z. Xu, D. Ding, W. Sze: Informedia E-Lamp@TRECVID 2012. In Proc. of TRECVID Workshop (2012).
14. H. Cheng, J. Liu, S. Ali, O. Javed: SRI-Sarnoff AURORA System at TRECVID 2012 Multimedia Event Detection and Recounting. In Proc. of TRECVID Workshop (2012).
15. I. Laptev: On space-time interest points. IJCV, vol. 64, pp. 107–123 (2005).
16. H. Wang, A. Klser, C. Schmid, C. L. Liu. Action recognition by dense trajectories. In Proc. CVPR, pp. 3169–3176 (2011).
17. F. Wang, Y. G. Jiang, C. W. Ngo: Video Event Detection Using Motion Relativity and Visual Relatedness. In Proc. ACM Multimedia, pp. 239–248 (2008).
18. M. Chen, A. Hauptmann: MoSIFT: Recognizing Human Actions in Surveillance Videos. CMU-CS-09-161, Carnegie Mellon University (2009).
19. E. bermejo, O. Deniz, G. Bueno, R. Sukthankar: Violence Detection in Video using Computer Vision Techniques. In Proc. Computer Analysis of Images and Patterns, LNCS 6855, pp. 332-339 (2011).
20. G. Farnebäck: Two-frame motion estimation based on polynomial expansion. In Scandinavian Conference on Image Analysis (2003).