Automatic Construction of an Action Video Shot Database using Web Videos

Do Hang Nga Keiji Yanai

Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chogugaoka, Chofu-shi Tokyo 182-8585 Japan
dohang@mm.cs.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract

There are a huge number of videos with text tags on the Web nowadays. In this paper, we propose a method of automatically extracting from Web videos video shots corresponding to specific actions with just only providing action keywords such as "walking" and "eating".

The proposed method consists of three steps: (1) tag-based video selection, (2) segmenting videos into shots and extracting features from the shots, and (3) visual-feature-based video shot selection with tag-based scores taken into account. Firstly, we gather video IDs and tag lists for 1000 Web videos corresponding to given keywords via Web API, and we calculate tag relevance scores for each video using a tag-co-occurrence dictionary which is constructed in advance. Secondly, we fetch the top 200 videos from the Web in the descending order of the tag relevance scores, and segment each downloaded video into several shots. From each shot we extract spatio-temporal features, global motion features and appearance features, and convert them into the bag-of-features representation. Finally, we apply the VisualRank method to select the video shots which describe the actions corresponding to the given keywords best after calculating a similarity matrix between video shots. In the experiments, we achieved the 49.5% precision at 100 shots over six kinds of human actions by just providing keywords without any supervision. In addition, we made large-scale experiments on 100 kinds of action keywords.

1. Introduction

A huge number of videos have been stored at video sharing sites on the Web such as YouTube and DailyMotion, and many videos are being uploaded to them every second. When people uploads their videos, they usually attach to the videos text keywords called as "tags" which enable other people to search for the uploaded videos by keyword-based search. However, in general, tags are attached to a whole video sequence of each video. Therefore, if attached tags correspond to only a specific part of the video, it is unknown which part of the video corresponds to the tags. For example, some videos which has "eating" as a tag might include the scenes of entering restaurants, ordering meals, and drinking coffee after eating. People who want to watch only "eating something" scenes have to skip the scenes of no interest, and search for eating scenes manually. This is a very troublesome and time-consuming task.

Then, in this paper, we propose a new method to detect most relevant video shots to given keywords from a large number of tagged Web videos, which requires no supervision and just only providing keywords at the beginning. As keywords, we mainly focus on the words related to human action such as "eating" and "running". To this end, we use a state-of-the-art spatial-temporal feature [24] to represent each video shot. Note that video shots mean small fragments of a video obtained by dividing the given video at the points of scene change or camera change.

If video shots corresponding to any "action" verbs can be obtained automatically, we can build training data on human action recognition for unconstrained videos easily. So far, constructing of action training data is very expensive, since collecting only video sequences corresponding to a specific action is very time-consuming, which is totally different from the situation of finding still images corresponding to a specific object. In fact, the largest action dataset used commonly so far includes only 14 kinds of categories [27]. On the other hand, we can gather video shots associated with unlimited kinds of actions by using the proposed method, although brief cleaning by hand is still needed to use them as training data for action recognition. Our final objective is automatic construction of an action video shot database which is helpful for the research community on action recognition.

In the proposed method, firstly, we rank the 1000 videos which have the given tags based on tag co-occurrence evaluation after obtaining video IDs and their tag lists via Web API. Secondly, we download the top 200 video in terms of tag co-occurrence scores and segment all the downloaded videos into video shots, and thirdly apply a graph-based ranking method, VisualRank [13], to rank video shots by taking account of visual features of video shots and tag co-occurrence scores so that shots corresponding to the given keywords are ranked higher.

To summarize our contribution in this paper, it consists of three-fold: (1) fully-unsupervised construction of an action video shot database, (2) two-step video shot selection consisting of tag-based video selection from large number of tagged videos, and visual-feature-based shot selection with the state-of-the-art spatio-temporal feature, and (3) a large-scale experiments on 100 kinds of actions with video metadata analysis on 100,000 YouTube videos and spatio-temporal feature analysis on 20,000 YouTube videos.

In the rest of this paper, we describe related work in Section 2. Then in Section 3, we explain the overview of the proposed method, and in Section 4 we explain the de-

tail of each processing including tag-based ranking, visual-feature-based ranking and the visual feature representation. Section 5 describes the experimental results. Finally we conclude this paper in Section 6.

2. Related Work

In this section, we refer to some related works on action recognition, Web image mining, and tag ranking methods.

Action recognition: For these five years, spatio-temporal (ST) features and their bag-of-features (BoF) representation have drawn attention for human action recognition and content-based video analysis, since by using them action recognition problem can be regarded as being almost the same problem as object recognition except feature extraction methods. Until two or three years ago, most of the works on action recognition focused on controlled video data such as KTH dataset [26] and Weizmann dataset [2] the videos of which are taken by a fixed camera with uniform backgrounds. Recently some works dealt with uncontrolled video such as "in-the-wild" YouTube dataset [19] and Hollywood action dataset [21], since classification rates on KTH and Weizmann has reached nearly perfect recognition rate, 95.5% and 100%, respectively [17]. In 2010, more works focused uncontrolled video categorization for YouTube videos [29, 28] and Kodak consumer video dataset [7]. Most of these works aimed to categorize whole videos into one of the pre-defined categories, while our objective is to search a large number of Web videos for part of videos associated with the given keywords.

In a few works, unsupervised methods were attempted for action recognition. Niebles *et al*. [23] categorized action videos in KTH datasets and their original ice-skating video data using the PLSA model. Niebles *et al*. [22] also proposed a method to extract human action sequences from unconstrained Web videos. Cinbis *et al*. [4] proposed a method to learn action models automatically from Web images gathered via Web image search engines, and recognize action for the same video dataset as [22]. Although Cinbis *et al*.'s work is the most similar to our work, they used Web images as a training source and only static features as an action descriptor, while we use Web videos and spatio-temporal features. In addition, in both works by Niebles *et al*. and Cinbis *et al*. a people detector based on HOG (Histogram of Oriented Gradient) [6] were used to extract a region of a human body, which restricts kinds of actions, while our method does not limit applicable actions to only human actions and it might be able to collect non-human actions such as "airplane-flying" and "tornado". As another similar work, Ballan *et al*. [1] proposed a method to add tags to video shots by using Web images obtained from Flickr as training samples. Meanwhile, Laptev *et al*. [14, 21, 8] proposed methods to associate movies and movie scripts automatically. These methods enable us to build an action shot database in a unsupervised manner, although target videos are limited to only the movies the scripts of which are available.

Web image mining: Regarding still images, many works on image gathering from the Web to build an image database automatically has been proposed so far [30, 10, 9, 31, 16, 25] Most of these works employed object recognition methods to select relevant images to given keywords from "raw" images collected from the Web using Web image search engines. Now we can import this idea to action video recognition domain by using the BoF representation of video shots. This is our initial motivation of this work, and our work can be regarded as being video shot version of these automatic Web image gathering.

Tag ranking: In this paper, we perform tag analysis to compute tag-based relevance scores. Having tags is common characteristic of consumer generated media (CGM) data on the Web. To have uploaded contents searched for by many people, adding tags to the uploaded contents is essential. Therefore, in general, images and videos on the Web have more than two tags, or sometimes more than ten tags. Yang *et al*. [32] proposed a method to evaluate a tag relevance score on each tag based on tag co-occurrence statistics which is called as "Web 2.0 Dictionary" in the paper. Since this method requires only tag analysis and no visual feature analysis, we use this to select Web videos to download. As a similar method which does not require visual features, Dong *et al*. [18] proposed a method to evaluate tag relevance score by combining the probabilistic relevance score estimation and random walk-based refinement. Although this two-step method is similar to ours, they use only tag information, while you use tag and visual features.

3. Overview of the Methods

In this paper, we propose a new method of automatically extracting from tagged Web videos video shots corresponding to specific actions with just only inputting action keywords such as "walking" and "eating". The proposed method consists of three processing steps (Figure 1): (1) tag-based video selection, (2) segmenting videos into shots and extracting features from all the shots, and (3) visual-feature-based video shot selection with tag-based scores taken into account.

In the first step, we evaluate a relevance score of each video to the given keywords before downloading videos from the Web, and select more relevant videos to be downloaded from a large number of videos which have the given keywords as the tags, since we evaluate the score by using only tag co-occurrence statistics without visual features. Note that a list of tags and video IDs corresponding to the given query words can be obtained via video search Web API officially provided by Web video sharing sites.

After downloading the videos, we divide each of them into video shots and extract visual features from each of all the shots. In this paper, as visual features extracted from each video shot, we use the spatio-temporal (ST) features proposed by Noguchi *et al*. [24], global motion features, Gabor appearance features, and their fusion.

In the third step, we rank video shots by applying graph-based ranking method, VisualRank [13], with a visual-feature-based similarity matrix and a bias damping vector based on tag-based video relevance scores. Finally we can obtain video shots corresponding to the given keywords in

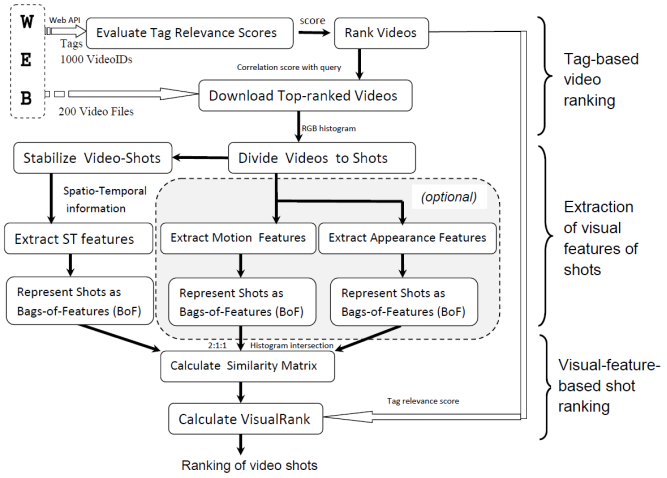

Figure 1. Overview of the proposed method.

the upper rank of the video shot ranking. Note that *video shots* are ranked in the third step, while *whole videos* are ranked in the first step.

4. Methods

In this section, we describe the detail on tag-based video ranking, feature extraction from shots and visual-feature-based video shot ranking.

4.1. Tag-based Video Ranking

We can easily obtain Web videos associated with the given keywords by using Web API. In case of YouTube, they provide Web API to search their video database for the videos tagged with the given query words. However, tags are sometimes only weakly related or unrelated to the corresponding videos, since tags are assigned subjectively by the uploaders. The objective of this step is to select the more query-related videos to download.

Firstly, we send the given keywords to the Web API of Web video sharing sites, and we obtain sets of video IDs and tags. Using co-occurrence of tags, we evaluate relevance of the video to the given keyword. To this end, we use the "Web 2.0 Dictionary" method proposed by Yang *et al*. [32]. "Web 2.0 Dictionary" corresponds to statistics on tag co-occurrence, which we need to construct in advance by gathering a large number of video tags from the Web.

Assume that $N(t)$ is the number of the videos tagged with word t among all the Web videos, and $\mathcal{T}$ is a set of all the words other than t over all the the Web videos. The correlation of parent word t and its child word $t_i \in \mathcal{T}$ is defined as

$$w(t, t_i) = \frac{F(t, t_i)}{N(t)} \tag{1}$$

where $F(t, t_i)$ is the number of videos tagged with both word t and word t_i at the same time. When $\mathcal{T}_V$ represents a set of all the words other than t in video V, we can estimate relevance of video V for word t, $P(V|t)$, by substituting $\mathcal{T}_V$ for V and $w(t, t_i)$ for $P(t_i|t)$ as follows:

$$\begin{aligned} P(V|t) &\propto P(\mathcal{T}_V|t) \\ &= \prod_{t_i \in \mathcal{T}_V} P(t_i|t) \\ &= \prod_{t_i \in \mathcal{T}_V} w(t, t_i) \end{aligned} \tag{2}$$

This is the original method to calculate relevance of an image/video to the give keyword as described in [32]. This is based on an idea that other tags than the query tag are supporters of the query tag, and the query tag can be regarded as being more relevant to the video when the query tag is supported by many supporter tags strongly related to the query word.

However, due to multiplying of all the correlation values between the query tag and the rest of tags within one video, the value of Eq.(2) becomes smaller as the number of tags increases. To prevent this, we modify it so that the number of co-occurrence words to be used for calculation is limited to m at most, and define the relevance score $Sc_t(V)$ using average log likelihood as follows:

$$\begin{aligned} S(V|t) &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 w(t, t_i) \\ &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} (\log_2 F(t, t_i) - \log_2 N(t)) \\ &= \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) - log_2 N(t) \end{aligned} \tag{3}$$

$$Sc_t(V) = \frac{1}{n} \sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) \tag{4}$$

where $\mathcal{T}'$ contains at most the top m word t_i in terms of the descending order of $w(t, t_i)$, and n $(n \leq m)$ represents $|\mathcal{T}'|$. Since the second term of Eq.(3) is always the same in the video set over the same action keyword, we omit it and define the relevance score $Sc_t(V)$ as shown in Eq.(4). In the experiment, we set m as 10, and select more relevant 200 videos to the given keyword from the 1000 videos returned by the Web API. This tag-based selection in the first step is important to feed promising videos to the second step which contains much more expensive feature extraction process.

Note that in case of compound keywords such as "drink coffee", we regard compound keywords as just one word and $N(t)$ as the number of the videos including all of the element word of the compound keyword in their tag sets. $w(t, t_i)$ is regarded as the number of videos having all the words of t and t_i even if t_i is also a compound word. For a video having no co-occurrence tags, we ignore such video in the experiments since we cannot calculate the relevance score.

In the experiments, as seed words, we prepared 150 sets of verbs and nouns which are related to actions such as "ride bicycle" and "launch shuttle". We gathered 1000 video tags for each seed word, and extracted all the tags. As a result, we obtained 12,471 tags which appear more than five times among all the collected tags. For each of 12,471 words,

we gathered 1000 video tags again, and constructed "Web 2.0 Dictionary" by counting tag co-frequencies according to Eq.(1).

4.2. Visual-feature-based Shot Ranking

After downloading the top 200 videos from the Web in the descending order of the relevance score estimated by the tag-based ranking method, we segment the downloaded videos into shots by a simple shot boundary detection method by thresholding color-histogram distances between adjacent frames.

Before extracting visual features, we select video shots from all the shots extracted from 200 videos, since the number of all the shots sometimes exceeds 10,000 and the total time of all the shots exceeds fifteen hours. To make computational cost feasible, in the experiment, we set the upper limit number of shots extracted from one video, and selected only the 2000 shots according to the following heuristic which intends to balance selecting more shots from the higher-ranked videos against selecting various shots from as many videos as possible.

$$N_{upper}(V_i) = c \times Sc(V_i) + f(N(V_i)) \quad (5)$$

$$\text{where } f(x) = \begin{cases} 20 & (x \leq 20) \\ 20 + (x-20)/4 & (20 < x < 100) \\ 40 & (x \leq 100) \end{cases}$$

and, $N_{upper}(V_i)$ and $N(V_i)$ represents the limit number of shots and the number shots extracted from the i-th video, respectively. $Sc(V_i)$ represents a tag-based relevance score of the i-th video. c is a constant which depends on the size of the "Web2.0 dictionary". In the experiment, we set c as 10. Basically we took into account both the number of shots detected by shot boundary detection and the tag relevance score of the video. We select $N_{upper}(V_i)$ shots at most from the i-th video at even intervals, and aggregate 2000 shots in the descending order of the tag relevance score $Sc(V)$.

After selecting shots to feed into visual-feature-based ranking, we extract visual features including spatio-temporal (ST) features, global motion features and appearance features from all the shots. The detail on the visual features will be explained in the next subsection.

As a method on visual-feature-based shot ranking, we employ the VisualRank method [13], which is an image ranking method based on the widely known Web page raking method, PageRank [3]. PageRank calculates ranking of Web pages using hyper-link structure of the Web. The rank values are estimated as the steady state distribution of the random-walk Markov-chain probabilistic model. In the iterative processing, each page gives out ranking points to its hyperlink destinations. Therefore, the ranking point of the page linked by more pages having much ranking points becomes higher. VisualRank uses a similarity matrix of images instead of hyper-link structure. Eq.(6) represents an equation to compute VisualRank.

$$\boldsymbol{r} = \alpha S \boldsymbol{r} + (1-\alpha)\boldsymbol{p} \quad (0 \leq \alpha \leq 1) \quad (6)$$

where S is the column-normalized similarity matrix of images, $\boldsymbol{p}$ is a damping vector, and $\boldsymbol{r}$ is the ranking vector each element of which represents a ranking score of each image. α plays a role to control the extent of effect of $\boldsymbol{p}$. Commonly, α is set as 0.85. The final value of $\boldsymbol{r}$ is estimated by updating $\boldsymbol{r}$ iteratively with Eq.(6). Because S is column-normalized and the sum of elements of $\boldsymbol{p}$ is 1, the sum of elements of ranking vector $\boldsymbol{r}$ also stays 1. Note that we assume that the elements of S and $\boldsymbol{r}$ correspond to the video shots in the descending order of the tag-based scores.

Although $\boldsymbol{p}$ is set as a uniform vector in VisualRank as well as normal PageRank, it is known that $\boldsymbol{p}$ can plays a bias vector which affects the final value of $\boldsymbol{r}$. Haveliwala [11] proposed to let topic-preferences reflect PageRank scores by giving larger values on the elements corresponding to the Web page related to the given topic. Basically, a bias vector can adjust ranking scores of images so that the rank scores of the biased images become higher.

In the similar way, in this paper, we propose to use a non-uniform damping vector according to the tag-based relevance score estimated in the first step in place of a uniform damping vector. We define two kinds of bias vectors as follows:

$$p_i^{(1)} = \begin{cases} 1/k & (i \leq k) \\ 0 & (i > k) \end{cases} \quad (7)$$

$$p_i^{(2)} = \begin{cases} Sc(V_i)/C & (i \leq k) \\ 0 & (i > k) \end{cases} \quad (8)$$

$$\text{where } C = \sum_{j=1}^{k} Sc(V_j)$$

$Sc(j)$ represents the tag relevance score of the video from which shot j was extracted.

The bias vector (1) represented in Eq.(7) is defined by giving uniform bias values to the elements corresponding to the top k shots regarding tag-based video scores, while the bias vector (2) represented in Eq.(8) is defined by setting the bias vector as the normalized values proportional to the tag-based scores within the top k shots. In fact, the bias vector (1) is similar to that Jing *et al*. [13] assigned bias values to only the top k images in terms of image ranking of commercial image search engine outputs.

4.3. Feature Extraction from Video Shots

To calculate the ranking scores of video shots using Eq.(6) via iterative computation, we need to prepare a visual similarity matrix S between video shots in addition to a bias vector $\boldsymbol{p}$. In this subsection, we describe the features to be extracted from video shots and how to compute a similarity matrix.

In this paper, as spatio-temporal (ST) features, we use the method we proposed in the previous paper [24], which has achieved the best 80.4% classification rate regarding "in-the-wild" YouTube dataset [19] by integrating them with global motion features and Gabor appearance features by Multiple Kernel Learning (MKL), while Liu *et al*., Cinbis *et al*. and Le *et al*. achieved 71.2% [17], 75.2% [5] and 75.8% [15], respectively. In this paper, we use this ST feature primarily, and global motion features and Gabor appearance features as optional features for feature fusion. We chose these three features so that their characteristics are

Figure 2. Steps to extract the ST feature. (1) detected SURF points, (2) detected SURF points with motion, and (3) obtained Delaunay triangles. (Cited from [24])

different from each other. All these features are not used as they are, but are vector-quantized and converted to bag-of-features (BoF) vectors regarding each shot.

4.3.1 Spatio-Temporal Feature

Following the method described in our previous paper [24], firstly, we detect interest points and extract feature vectors employing the SURF method [12], and then we select moving interest points employing the Lucas-Kanade method [20]. In this method, only moving interest points are considered as ST interest points and static interest points are discarded, because it is expected that a ST feature represents how objects in a video are moving. After detecting moving interest points, we apply Delaunay triangulation to form triples of interest points where both local appearance and motion features are extracted. In addition, we track each interest point for consecutive five frames, and describe flow directions of interest points and change of the size of the triangles. This enables us to extract ST features not from one point but from a triangle surface patch, which makes the feature more robust and informative. The ST features are extracted every five frames. This method is relatively faster than the other ST features such as cuboid-based features, since it employs SURF [12] and the Lucas-Kanade method [20], both of which are known as very fast detectors. Totally the dimension of the ST feature vector is 256. Figure 2 shows an example of the processing steps to extract the ST features.

4.3.2 Motion Feature

Although the proposed ST feature contains motion information, it represents only local motion. As a holistic motion feature, we build motion histograms over a frame image. This feature is expected to have different discriminative power from the ST feature. We extract motion features at grid points with every 8 pixels using the Lucas-Kanade method [20]. Extracted motion features from each grid are voted to histogram of 7 direction and 8 motion magnitude.

4.3.3 Appearance Feature

We use Gabor texture histograms as an appearance feature. A Gabor texture feature represents texture patterns of local regions with several scales and orientations. In this paper, we use 24 Gabor filters with four kinds of scales and six kinds of orientations. Before applying the Gabor filters, we divide a frame image extracted from video shots into 20×20 blocks. We apply the 24 Gabor filters to each block, then average filter responses within the block, and obtain a 24-dim Gabor feature vector for each block. Totally, we extract 400 24-dim Gabor vectors from each frame image.

4.3.4 Vector Quantization of Features for a Shot

We extract global motion and appearance features from every 4 frames within each video shot, while we extract ST features from every 5 frames, and we vector-quantize all of them and convert them into the bag-of-features (BoF) representation within each shot. While the standard BoF represents the distribution of local features within one image, the BoF employed in this paper represents the distribution of features within one shot which consists of several frame images. We call this BoF regarding one video shot as bag-of-frames (BoFr).

In the experiment, we set the size of the codebook of the 256-dim ST features, the 24-dim appearance features and the 56-dim motion features as 5000, 5000 and 3000, respectively.

4.3.5 Computation of Similarity Matrix

To obtain a similarity matrix S between video shots for calculation of the ranking scores of video shots using Eq.(6) via iterative computation, we use histogram intersection as follows:

$$s(H_i, H_j) = \sum_{l=1}^{|H|} min(h_{i,l}, h_{i,l}) \tag{9}$$

where H_i, $h_{i,l}$ and $|H|$ represents the BoF vector of the i-th shots, its l-th element and the dimension number of the BoF vector, respectively.

In addition to single features, we made experiments on fusion of the three kinds of features by linear combination. A combined similarity is obtained as follows:

$$S_{combined} = w_{ST} \times S_{ST} + w_{mot} \times S_{mot} + w_{app} \times S_{app} \tag{10}$$

where S_{ST}, S_{mot} and S_{app} represents the similarity matrix of ST features, motion features and appearance features, respectively.

In case of supervised action recognition, optimal fusion weights can be estimated by machine learning methods such as Multiple Kernel Learning and AdaBoost. However, in our case, we have no training data, since this work aims unsupervised action database construction. Therefore, it difficult to use any optimization methods for estimating weights. Instead, in the experiment, we set three weights heuristically as follows:

$$w_{ST} = \frac{1}{2},\ w_{mot} = \frac{1}{4},\ w_{app} = \frac{1}{4} \tag{11}$$

To estimate optimal weights even under unsupervised way is one of our future works.

4.3.6 Treatment for Camera Motion

Most existing works on action recognition do not consider camera motions, since most of them assume a fixed camera.

However, it is important to cope with camera motion in case of Web videos, and then some recent works on action recognition for unconstrained Web videos coped with it. Cinbis *et al*. [5] employed a homography-based camera motion compensation approach at the task of action recognition on Web videos. Unfortunately, since they did not compare their results with the results without camera motion compensation, the effectiveness of camera motion compensation for action recognition on unconstrained Web videos is unclear.

Then, in this paper, we examine and compare two cases: with camera motion removal, and without it. To compensate camera motion, we make use of a homography-based camera motion compensation approach, which is basically the same as [5]. Although compensation of camera motion is possible, accurate compensation is difficult for Web videos. This is because Web videos contain various kinds of intentional and unintentional camera motions and their resolution is usually low. Especially, for our objective which is automatic building of an action shot database, the shots stored in the database should have no camera motion. The situation is different from the case of action recognition where they must classify all the given videos which might include camera motion. Therefore, in this paper, we adopt a simple strategy that we discard the shots where camera motion is detected as a method with no camera motion compensation, which is the same as [19]. In the actual implementation, we detect camera motion before extracting features as preprocessing.

To detect camera motion, we calculate motion features based on the Lucas-Kanade method at every 8-pixel grid. If the region where motion is detected is larger than a predefined threshold, we consider camera motion is detected.

5. Experimental Results

To examine effectiveness of the proposed method, we made experiments with various conditions using six kinds of human action keywords: "batting", "eating ramen", "jumping trampoline", "running marathon", "shooting football" and "walking street". Some examples of the frame images of six actions are shown in Figure 3. All the verbs included in these six keywords are the same as ones used in [24], since we will compare our results with Noguchi *et al*.'s results brought by the state-of-the-art supervised method employing MKL-based feature fusion.

In addition, we made two additional experiments: collecting video shots of non-human actions, and building 100 kinds of action database.

In the experiments, we obtained rankings of 2000 shots for each actions. For evaluation of ranking results, in general, average precision is widely used. However, we mainly use the precision rate at rank 100 rather than average precision, since commonly used dataset on human action such as KTH dataset [26] and "in-the-wild" YouTube dataset [19] has about the 100 video shots per action [1] and the objective of our work is automatic construction of an action video shot database which is helpful for the research community on action recognition.

[1] KTH dataset has 599 shots for 6 actions, and "in-the-wild" dataset has 1168 shots for 11 actions.

Figure 3. Examples of six kinds of the actions.

In all the experiments, we used YouTube.com as data source. We collected video metadata including video IDs and tags using YouTube Data API.

5.1. Experiments with various settings

As shown in Table 1, we made 10 kinds of experiments by changing the conditions. The condition of each experiment means as follows:

RND Download 200 videos for the given keywords, and select 100 shots randomly.

TAG Download the top 200 videos among 1000 videos in terms of tag-based ranking, and select 100 shots randomly.

Exp.1 Download 200 videos for the given keywords without tag-based ranking, select 2000 shots randomly, and re-ranked them by visual-feature-based ranking using only ST features with a uniform damping vector.

Exp.2 Add tag-based video shot selection to Exp.1.

Exp.3 Use a bias damping vector obtained by (1) Eq.(7) or (2) Eq.(8) instead of a uniform damping vector in Exp.2.

Exp.4 Add camera motion compensation to Exp.3(1).

Exp.5 Use motion features instead of ST features in Exp.3(1).

Exp.6 Use appearance features instead of ST features in Exp.3(1).

Exp.7 Use combined features of three kinds of features instead of ST features in Exp.3(1).

Exp.8 Add camera motion compensation to Exp.7.

The results of the experiments explained above are shown in the last columns of Table 1. From these results, RND brought the worst result, which selected just randomly 100 shots. TAG improved RND by 9.3% by introducing tag-based ranking to select videos to download. Exp.1 outperformed TAG, which used visual-feature-based ranking. Exp.2 improved Exp.1 by 7.3% by combining tag-based

Table 1. The conditions of the experiments and their results.

Exp. no	tag-based ranking	biased damp. vec.	camera motion compensation	feature	mean prec@100
RND	randomly-selected 100 shots				14.2 %
TAG	√	-	-	-	23.5 %
1	-	-	-	ST	33.7 %
2	√	-	-	ST	41.0 %
3(1)	√	√(1)	-	ST	**47.3 %**
3(2)	√	√(2)	-	ST	44.8 %
4	√	√	√	ST	39.8 %
5	√	√	-	motion	31.8 %
6	√	√	-	appear.	39.7 %
7	√	√	-	**fusion**	**49.5** %
8	√	√	√	**fusion**	41.2 %

Table 2. Precision@100 of six actions (%).

Exp. no	batting	eating	jumping	running	shoot	walking	AVG.
RND	12	13	17	23	4	16	14.2 %
TAG	32	14	45	23	20	7	23.5 %
1	56	37	69	20	5	23	33.7 %
2	77	30	75	23	17	24	41.0 %
3(1)	66	42	82	35	33	29	47.3 %
3(2)	73	28	90	38	27	30	44.8 %
4	38	31	78	45	12	**36**	39.8 %
5	69	10	67	12	32	20	31.8 %
6	35	**57**	64	**46**	7	29	39.7 %
7	69	39	**87**	30	**36**	**36**	**49.5 %**
8	61	37	74	38	16	21	41.2 %
9 (0.75)	68	32	80	29	26	29	44.0 %
9 (0.80)	**82**	34	83	29	30	33	48.5 %
9 (0.85), 7	69	39	87	30	36	36	**49.5 %**
9 (0.90)	77	31	79	26	28	35	46.0 %
9'(0.95)	74	31	81	28	25	32	45.2 %
MKL [24]	83	78	98	87	82	52	80.0 %

ranking and visual-feature-based ranking.

In Exp.2, we used a uniform damping vector for VisualRank computation. In Exp.3, we tried using a non-uniform bias damping vector in the two ways: Exp.3(1) gave uniform weights to the top 1000 shots, and Exp.3(2) assigned the weights in proportion to the tag-based relevance scores. From the results of Exp.3(1) and Exp.3(2), it turned out that the method (1) was superior to the method (2) as well as Exp.2, although the difference is not so large. Therefore, in all the the experiments after Exp.3, we used the method (1) to set a bias damping vector.

In Exp.4, we introduced camera motion compensation. However, Exp.4 was degraded from Exp.3. This never means that camera motion removal is useless. This just indicates that the default strategy that shots with camera motion are discarded is superior to the alternative strategy that shots with camera motions are used after camera motion removal. Since many good-conditioned video shots with no camera motion can be extracted from the downloaded videos, we do not need to use less-conditioned video shots in which camera motion were removed artificially.

In Exp.5 and Exp.6, we used global motion features and appearance features instead of ST features, which achieved 31.9% and 39.7%, respectively. This shows that ST features, which achieved 47.3%, were much superior to the two other features in this task.

Exp.7 achieved the best result, 49.5%, which used all the proposed method including feature fusion except camera motion compensation. Compared with Exp.3(1), feature fusion improved the result slightly from 47.3% to 49.5%. In Exp.8, we added camera motion compensation to Exp.7. As a result, Exp.8 was degraded in the similar way as Exp.4.

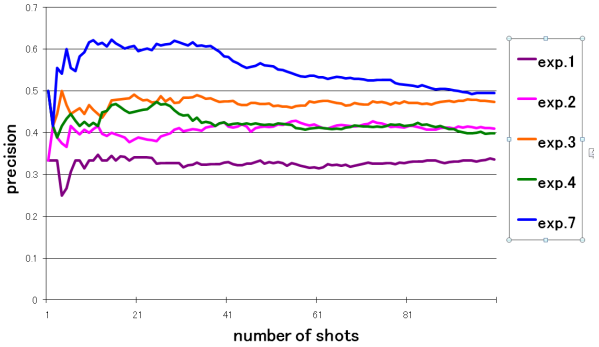

Figure 4. Mean precision at the n-th shot over six actions.

We show all the results of six actions in Table 2. In Exp.4, the results for "walking" and "running" were improved compared to Exp.3(1). Since the videos related to "walking" and "running" are usually recorded by moving cameras, the number of good-conditioned shots without camera motion is limited. Therefore, in case of discarding the shots with camera motion, too many shots were discarded. This is the reason why using even less-conditioned shots after motion removal is better strategy for "walking" and "running".

In Table 2, we add Exp.9 which means the experiments in case of changing the value of α in Eq.(6) from 0.75 to 0.95 with the step of 0.05. The conditions except α is the same as Exp.7 which achieved the best results, 49.5%. The value in the parenthesis in the table represents α. Note that Exp.9 (0.85) is equivalent to Exp.7. The results on Exp.9 indicate that 0.85 is the best value for α. In addition, we show the result by the state-of-the-art supervised learning methods [24] in the bottom row in Table 2, which shows that the difference is still larger in terms of precision except for "batting".

In Figure 4, we show the mean precision at the n-th shot over six actions. From this graph, Exp.7 achieved the best precision even at the lower rank.

5.2. Additional Experiments on Non-human Actions and 100 Actions

In addition, we made two additional experiments: collecting video shots with non-human action keywords such as "airplane flying" and building a large-scale action video shot database with 100 kinds of action keywords. In all the additional experiments, we used only ST features in the same way as Exp.3(1), since extraction of three kinds of features for fusion requires heavy computation.

The results of 6 kinds of non-human actions including artifact actions and natural phenomena are shown in Table 3. The results were not as good as the six human actions, since we used only ST features which was suitable for representing human actions. Seeking suitable features for non-human action is one of our future works.

We made large-scale experiments on 100 kinds of actions

Table 3. Precision@100 of non-human actions (%)

aircraft +landing	tornado	blooming +flower	airplane +flying	earthquake	shuttle +launching	**AVG.**
30	39	44	14	7	18	**25.3**

Table 4. Precision@100 of 100 human actions (%)

soccer+dribble	100	play+drum	40	climb+tree	24
fold+origami	96	skate	37	ride+horse	24
crochet+hat	95	swim+crawl	36	roll+makizushi	24
arrange+flower	94	cut+hair	35	sew+button	24
paint+picture	88	run+marathon	35	fry+tempura	23
boxing	86	count+money	33	slap+face	20
jump+parachute	82	paint+wall	33	read+book	19
jump+trampoline	82	shoot+football	33	squat	19
do+exercise	79	draw+eyebrows	32	row+dumbell	16
do+aerobics	78	fieldhockey+dribble	32	wash+clothes	15
do+yoga	77	hit+golfball	32	wash+dishes	15
surf+wave	75	lunge	32	comb+hair	14
shoot+arrow	73	play+piano	32	drink+coffee	14
massage+leg	72	row+boat	32	swim+breaststroke	13
fix+tire	67	sing	32	cry	12
batting	66	chat+friend	31	eat+sushi	12
basketball+dribble	64	clean+floor	31	serve+teniss	11
blow-dry+hair	64	cut+onion	31	tying+tie	11
knit+sweater	64	shave+mustache	31	boil+egg	9
ride+bicycle	62	pick+lock	30	head+ball	9
curl+bicep	58	plaster+wall	30	swim+backstroke	9
shoot+ball	58	blow+candle	29	take+medicine	8
tie+shoelace	57	wash+face	29	serve+volleyball	7
laugh	50	walking+street	29	swim+butterfly	7
dive+sea	49	brush+teeth	28	bake+bread	6
harvest+rice	49	catch+fish	28	cook+rice	6
ski	49	drive+car	28	grill+fish	5
iron+clothes	47	plant+flower	28	jog	5
twist+crunch	47	play+guitar	28	slice+apple	5
dance+flamenco	45	lift+weight	27	peel+apple	5
dance+hiphop	43	raise+leg	27	bowl+ball	4
eat+ramen	42	hang+wallpaper	26	smile	4
dance+tango	41	jump+rope	26	kiss	2
play+trumpet	41			**AVG. (68-100)**	**12.2**
AVG. (1-34)	**65.9**	**AVG. (35-67)**	**31.0**	**AVG. (ALL)**	**36.6**

with video metadata analysis on 100,000 YouTube videos and spatio-temporal feature analysis on 20,000 YouTube videos. We show the results for 100 kinds of actions in Table 4 including 6 kinds of actions used in the previous experiments. The mean of the precision at 100 shots over 100 actions was 36.6%, and the precision of each action varies from 2 to 100. This shows that the result depends on the kinds of actions and selection of action keywords sent to Web API as query words greatly. Especially, single action keywords such as "smile" and "cry" were too ambiguous to obtain good candidate videos. Although the current performance is not enough to built action shot database for most of actions in the full automatic manner, for some actions the proposed method worked very well.

All the results including a video summary of each of 100 actions and direct links to the original videos on the YouTube can be seen at our project Web page: http://mm.cs.uec.ac.jp/webvideo/ .

6. Conclusions

In this paper, we presented a method of automatically extracting from Web videos video shots corresponding to specific actions with just only providing action keywords. In the experiments, we achieved the 49.5% precision at 100 top-ranked shots over six kinds of human actions and the 36.6% precision for 100 kinds of human actions without any supervision. Although the obtained results were not enough, we believe that the direction we proposed in the paper is promising.

As future works, we plan to improve the way of setting bias damping vectors in VisualRank calculation to achieve better results, and to seek new kinds of visual features which work even for non-human actions.

References

[1] L. Ballan, M. Bertini, A. D. Bimbo, M. Meoni, and G. Serra. Tag suggestion and localization in user-generated videos based on social knowledge. In *Proc. ACM MM WS on Social Media*, pp. 3–7, 2010. 2

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005. 2

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998. 4

[4] N. I. Cinbins, R. G. Cinbins, and S. Sclaroff. Learning action from the web. In *ICCV*, pp. 995–1002, 2009. 2

[5] N. I. Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, pp. 494–507, 2010. 4, 6

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, vol. 1, pp. 886–893, 2005. 2

[7] L. Duan, D. Xu, I. W. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, 2010. 2

[8] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *CVPR*, pp. 1491–1498, 2009. 2

[9] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, pp. 1816–1823, 2005. 2

[10] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, pp. 242–255, 2004. 2

[11] T. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE trans. on KDE*, 15(4):784–796, 2003. 4

[12] B. Herbert, E. Andreas, T. Tinne, and G. Luc. Surf: Speeded up robust features. *CVIU*, pp. 346–359, 2008. 5

[13] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *PAMI*, 30(11):1870–1890, 2008. 1, 2, 4

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[15] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. 4

[16] L. Li and L. Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. In *CVPR*, 2007. 2

[17] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009. 2, 4

[18] D. Liu, X. Hua, L. Yang, M. Wang, and H. Zhang. Tag ranking. In *WWW*, pp. 351–360, 2009. 2

[19] J. Liu, J. Luo, and M. Shah. Recognizing realistic action from videos. In *CVPR*, 2009. 2, 4, 6

[20] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pp. 674–679, 1981. 5

[21] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *ICCV*, 2009. 2

[22] J. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting moving people from internet videos. In *ECCV*, pp. 527–540, 2008. 2

[23] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006. 2

[24] A. Noguchi and K. Yanai. A surf-based spatio-temporal feature for feature-fusion-based action recognition. In *ECCV WS on Human Motion: Understanding, Modeling, Capture and Animation*, 2010. 1, 2, 4, 5, 6, 7

[25] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007. 2

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, pp. 32–36, 2004. 2, 6

[27] J. Sun, X. Wu, S. Yan, L. F. Cheong, T. S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009. 1

[28] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik. Finding meaning on YouTube: Tag recommendation and category discovery. In *CVPR*, pp. 3447–3454, 2010. 2

[29] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. YouTubeCat: Learning to categorize wild web videos. In *CVPR*, 2010. 2

[30] K. Yanai. Generic image classification using visual knowledge on the web. In *ACM MM*, pp. 67–76, 2003. 2

[31] K. Yanai and K. Barnard. Probabilistic Web image gathering. In *ACM MIR*, pp. 57–64, 2005. 2

[32] Q. Yang, X. Chen, and G. Wang. Web 2.0 dictionary. In *CIVR*, pp. 591–600, 2008. 2, 3