The University of Electro-Communications,
Tokyo

Doctoral Dissertation

To Better Exploration of Action Recognition in Videos

Author:
Hang Nga Do

Supervisors:
Assoc. Prof. Yanai Keiji
Prof. Onai Rikio

A dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Department or Informatics,
Graduate School of Information and Engineering

March 2015

To Better Exploration of Action Recognition in Videos

Approved by supervisory committee:

Chairperson: Associate Professor Keiji Yanai
Member: Professor Rikio Onai
Member: Associate Professor Hiroki Takahashi
Member: Associate Professor Hayaru Shono
Member: Associate Professor Naoki Hashimoto

ビデオ映像に対する人間動作の認識

ドー　ハン　ガー

概要

本論文では、ビデオ映像に対する多様人間動作の高精度な認識の実現のために、様々な人間の動作に対応した映像を Web から自動収集することによって大規模映像データベースを構築するためのフレームワーク、及びそのために有用な動作認識のための特徴抽出手法を提案する．

第 1 章では、動作認識の研究の現状および大規模な動作データベースの重要性とその構築における問題点を述べ、その問題を解決するための本論文で提案する一連の研究について述べる．

第 2 章では、特定動作の対応したショットを Web 上に存在する大量の動画から自動的に抽出するフレームワークを提案する．このフレームワークでは、まず与えられた動作キーワードに対応した Web 動画のリストおよびメタデータを Web 動画共有サイトから大量に取得し、メタデータに含まれる各 Web 動画のタグ情報を分析することによって、より深くキーワードと関連していると推定される動画を選択する．次に選択したビデオをショットに分割し、それらのショットの視覚的な関係によってショットランキングを行う．ショットランキングの上位には、より深く動作と対応したショットがランキングされることが期待される．さらに、ランキングの精度を高めるために、画像を Web 画像検索エンジンから取得してショットランキングに導入する方法も提案する．実験では 100 種類の人間動作と 12 種類の非人間動作に対し提案フレームワークを適用し、多くの関連ショットが得られた．

第 3 章では、VisualTextualRank という新しいランキング手法を提案する．VisualTextualRank は VisualRank の拡張版であり、視覚特徴だけではなくテキスト特徴も有効活用するランキング手法である．第 2 章でのショットランキングにおいていは既存の VisualRank と呼ばれる手法を利用していたが、第 3 章では VisualRank に代わりに VisualTextualRank を適用し、実験にて VisualTextualRank の有効性を検証した．その結果、100 種類の人間動作の大部分において認識精度の向上が実現できた．

第 4 章では、第 2 章で紹介した提案フレームワークの精度をさらに改善することを目的とし、第 2 章と第 3 章で使った既存の視覚特徴抽出法の改良手法を提案する．特徴点のデンスサンプリングと選択法を導入し、より多くの代表的な特徴を得る．また、新しい時空間特徴も提案し、その特徴の従来の特徴に対する相補性を検証する．実験では動作分類および動作ショット抽出での提案手法の有効性を示した．動作分類実験は一般に広く使われている大規模の動作認識評価用データセットで行い、映像の動作認識研究における最先端の結果に匹敵する結果が得られた．また、既存の最高性能を誇る特徴と提案特徴を統合す

ることによって、さらに高い精度を得ることができた．さらに、提案手法を第 2 章で提案したフレームワークに適用して動作ショット抽出実験を行った結果、抽出精度を大幅に向上させることができることを確認した．

第 5 章では、人間の手の動きと関連した動作を注目し、手を使って行われる動作の認識精度を向上することを目的とし、手検出・追跡のシステムを提案する．実験では、チャレンジ性が高いビデオデータセットにおいて、提案の手の動作の検出法の有効性が確認できた．また、提案システムを詳細動作分類にも応用した．検出された手領域から特徴を抽出した場合とフレーム全体から特徴を抽出した場合に精度向上が得られた．よって、提案手法によって抽出された特徴が詳細動作をよく表現できる特徴であることが示された．

第 6 章では、本論文の結論をまとめるとともに、今後の改良点を述べる．また、本論文の得られた成果を使った大規模の動作の視覚的な分析の研究の展望についても考察する．

THE UNIVERSITY OF ELECTRO-COMMUNICATIONS, TOKYO

Abstract

Department or Informatics,
Graduate School of Information and Engineering

Doctor of Philosophy

To Better Exploration of Action Recognition in Videos

by Hang Nga Do

Our overall purpose in this dissertation is automatic construction of a large-scale action database with Web data, which could be helpful for the better exploration of action recognition. We conducted large-scale experiments on 100 human actions and 12 non-human actions and obtained promissing results. This disseration is constructed with 6 chapters. In the followings, we briefly introduce the content of each chapter.

In Chapter 1, recent approaches on action recognition as well as the necessity of building a large-scale action database and its difficulties are described. Then our works to solve the problem are concisely explained.

In Chapter 2, the first work which introduces a framework of extracting automatically relevant video shots of specific actions from Web videos is described in details. This framework at first, selects relevant videos among thousands of Web videos for a given action using tag co-occurance and then, divides selected videos into video shots. Video shots are then ranked based on their visual linkage. The top ranked video shots are supposed to be the most related shots of the action. Moreover, our method of adopting Web images to shot ranking is also introduced. Finally, large-scale experiments on 100 human actions and 12 non-human actions and their results are described.

In Chapter 3, the second work which aims to further improve shot ranking of the above framework by proposing a novel ranking method is introduced. Our proposed ranking method, which is called VisualTextualRank, is an extension of a conventional method, VisualRank, which is applied to shot ranking in Chapter 2. VisualTextualRank effectively employs both textual information and visual information extracted from the data. Our experiment results showed that using our method instead of the conventional ranking method could obtain more relevant shots.

In Chapter 4, the third work which aims to obtain more informative and representative features of videos is described. Based on a conventional method of extracting spatio-temporal features which was adopted in Chapter 2 and Chapter 3, we propose to extract spatio-temporal features with triangulation of dense SURF keypoints. Shape features of the triangles along with visual features and motion features of their points are taken into account to form our features. By applying our method of feature extraction to the framework introduced in Chapter 2, we show that more relevant video shots can be retrieved at the top. Furthermore, the effectiveness of our method is also validated on action classification for UCF-101 and UCF-50 which are well-known large-scale data sets. The experiment results demonstrate that our features are comparable and complementary to the state-of-the-art.

In Chapter 5, the final work which focuses on recognition of hand motion based actions is introduced. We propose a system of hand detection and tracking for unconstrained videos and extract hand movement based features from detected and tracked hand regions. These features are supposed to help improve results for hand motion based actions. To evaluate the performance of our system on hand detection, we use VideoPose2.0 dataset which is a challenging dataset with uncontrolled videos. To validate the effectiveness of our features, we conduct experiments on fine-grained action recognition with "playing instruments" group in UCF-101 data set. The experiment results show the efficiency of our system.

In Chapter 6, our works with their major points and findings are summarized. We also consider the potential of applying the results obtained by our works to further researches.

Acknowledgements

This research dissertation would not have been possible without the support of many people. I wish to express my gratitude to my supervisor, Associate Professor Keiji Yanai who was abundantly helpful and offered invaluable assistance, support and guidance. Special thanks also to all my graduate friends and seniors, especially Noguchi Akitsugu, for sharing the literature and invaluable assistance.

I would also like to convey thanks to Tonen International Scholarship Foundation and Japan Society for the Promotion of Science for providing the financial means.

Finally, I wish to express my love and gratitude to my beloved families; for their understanding & endless love, through the duration of my studies.

Contents

List of Figures

List of Tables

Abbreviations

SVM	**S**upport **V**ector **M**achines
STF	**S**patio-**T**emporal **F**eature
HOG	**H**istogram of **O**riented **G**radients
HOF	**H**istogram of **O**ptical **F**low
MBH	**M**otion **B**oundary **H**istogram
BoV	**B**ag **o**f **V**isual words
FV	**F**isher **V**ector
DNN	**D**eep **N**eural **N**etworks
SURF	**S**peeded **U**p **R**obust **F**eatures
VR	**V**isual**R**ank
VTR	**V**isual**T**extual**R**ank
API	**A**pplication **P**rogramming **I**nterface

Chapter 1

Introduction

1.1 Research Background

An action can be considered as a sequence of primitive movements generated by a human agent that fulfil a function or purpose, such as jumping, walking, or kicking a ball. Action recognition is the process of naming actions, usually in the simple form of an action verb, i.e. to determine the action label that best describes an action instance, even when performed by different agents with large variations in viewpoints, manner or surrounding conditions such as background, illumination and so on.

Since the 1980s, action recognition research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as medicine, robotics, human-computer interaction or sociology, among others. Action recognition is a fundamental key of video analysis based applications such as video surveillance [45, 48, 62, 65, 93] and video retrieval [49]. Especially, in recent years, the continuous development of video production and archiving has led to the great need for automatic video annotation tools. If it is possible to automatically label which actions have been performing in a video with high precision, it will cost much less human effort for video summarization, video surveillance and so on. In fact, action recognition covers wide range of research fields including motion analysis [1, 14, 54, 94], dynamic scene understanding [13], human behavior understanding [90], human action classification [95],

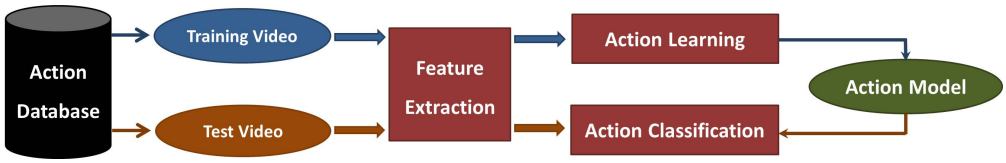

FIGURE 1.1: A typical process flow for action recognition which comprises independent stages of feature extraction, learning and classification.

human activity recognition [122, 101] or video event recognition [89, 118, 115, 34, 19] and so on. In this dissertation, we concentrate on human action classification.

A typical process flow for action classification comprises independent stages of feature extraction, learning and classification (See Figure 1.1 for the illustration). First, feature extraction is performed on all videos in the action database including training videos and test videos. Here feature extraction consists of the extraction of visual and/or motion cues from the videos that are discriminative with respect to the actions, and the encoding of videos using extracted features. Next, an action model is learned using training data. Finally, the learned model is applied to classify new feature observations (test data). In the same way as many other areas of pattern recognition, supervised learning models, Support Vector Machines (SVMs) have become the most widely used models for action recognition. In this dissertation, we focus on construction of action database and extraction of video features.

An action database consists of a number of videos showing particular actions. Typically, an action database is constructed manually: annotators must localize the pre-defined actions in the video source by watching the videos carefully. This has been known as a terribly time-consuming work. In action recognition, only a primary action is considered as a target in both training videos and test videos. Even with only one action, the task is still challenging due to the variability of human actions. The actions can look different when they are seen from different views or operated by different people. They even can be manipulated in many disparate ways. Thus, to obtain good recognition performance, training data should capture actions in many different conditions. In other words, action database should be large and able to reflect as much as possible the diversity of actions.

Feature extraction is the main vision task in action recognition. In this step, as visual and motion cues of videos, Spatio-Temporal (ST) features which describe both spatial and temporal description of movement have become the most exploited ones due to their verified efficiency and practicality. Three years ago Wang et al. proposed a method to

extract multiple features (HOG, HOF, MBH) aligned with trajectories of dense sampled points [129]. This method has become the state-of-the-art and the most popular ST feature extraction approach. Their dense trajectory based features have been widely used in many areas such as generic action recognition [55, 127, 43, 51, 111], activity recognition [101, 83, 149], and video event recognition [72, 115, 34].

With extracted features, a video can be encoded as a vector by applying an encoding technique. Up to several years ago, the most popular encoding technique was BoV (Bag of Visual words) model. BoV model learns a codebook offline by clustering a large set of descriptors with k-means and assigns each descriptor of an image to the closest entry in the codebook. Nevertheless, the BoV model suffers from some limitations, one of which is the loss of some discriminative information in both spatial and temporal dimensions. Several years ago, Fisher Vector (FV) encoding technique was applied to image classification task and shown to extend the BoV representation [91]. FV has many advantages with respect to the traditional BoV: it can be computed from much smaller vocabularies and therefore at a lower computational cost; it performs well even with simple linear classifiers; it can be compressed with a minimal loss of accuracy using product quantization. The approach of using FV of ST features has been exploited by many researches on human action recognition and content-based video analysis and shown to be very effective and easy to implement [130, 4, 81]. In this dissertation, BoV is applied to represent videos in Chapter 2 and Chapter 3; FV is used in Chapter 4 and Chapter 5.

Deep Neural Networks (DNNs) have recently shown outstanding performance on image classification and detection tasks [66, 117]. DNNs have been expected to replace engineered features, such as SURF, HOG or HOF, for a wide variety of problems including action recognition. However, up to now, none of DNNs based action recognition approaches have successfully verified significant performance improvements of DNNs over dense trajectory based features with FV [60, 113].

1.2 Objective

In this dissertation, our overall purpose is the automatic construction of a large-scale action database with Web data, which could be helpful for the better exploration of

action recognition. While image datasets contain thousands of object categories [102] with million of images, action datasets lag far behind. The largest dataset up to date, UCF101, has only 101 action categories with about 13000 video shots [61]. The main reason is the requirement of tremendous human effort on building a large-scale action database (as referred above). In this dissertation, we proposed to extract from Web data relevant video shots of specific actions by an unsupervised method. Our final objective is to construct a large-scale action shot database with minimal manual supervision. To this end, we conducted four following works.

The first work introduces a system of extracting automatically from Web videos relevant video shots of specific actions. Our main idea is at first, selecting relevant videos among thousands of Web videos for specified action and then, extracting the most related shots from selected videos. For video selection, we use tag co-occurance frequencies. For the extraction of corresponding shots, we apply an unsupervised ranking method called VisualRank (VR) [57].

The second work develops a novel ranking method, VisualTextualRank (VTR), which improves VR by effectively employing both textual information and visual information extracted from the data. We applied VTR to our above mentioned system. Based on our experiment results, we could demonstrate that our ranking method can improve the performance of video shot retrieval over the conventional ranking method VisualRank.

The third work aims to design a novel method of extracting low-level Spatio-Temporal Features (STFs) based on triangulation of dense SURF keypoints which have dominant and reliable movements. Our spatio-temporal features investigate triangles which are produced by applying Delaunay triangulation to those informative points. Shape features of the triangles along with visual features and motion features of their points are taken into account to form our features. By apply our method of feature extraction to the system of extracting automatically relevant Web video shots of specific actions introduced in the first work, more relevant video shots can be retrieved at the top. We further conducted experiments on several action recognition benchmarks to show the effectiveness of our proposed features on recognition task.

The final work focuses on recognition of hand movement based actions. We designed a system of hand detection and tracking for unconstrained videos. We applied hand

movement based features extracted from detected and tracked hands to classify fine-grained actions and obtained promising results. Our features can be expected to be helpful for improvement of extraction of Web video shots for hand motion related actions.

We made large-scale experiments on more than 100 action keywords and obtained promising results. According to our works, automatic construction of a large-scale database for various actions can be accomplished without any difficulty. Our main contributions can be summarized as follows:

(1) An automatic system of extracting relevant video shots of specific actions from the Web which enables us to construct large-scale action video shot database

(2) A novel ranking method which analyzes simultaneously visual links among video shots along with textual links between videos and their tags

(3) A novel method of extracting Spatio-Temporal Features based on triangulation of dense SURF keypoints

(4) A system of hand detection and tracking for uncontrolled videos and the idea of implementing it to perform fine-grained action recognition

1.3 Structure of This Dissertation

The remainder of this dissertation is organized as follows. Our method of constructing automatically action video shots using Web data is described in Chapter 2. The detail of our ranking method, VisualTextualRank, is explained in Chapter 3. Our proposed method of extracting spatio-temporal features is described in Chapter 4. Our system of hand detection and tracking as well as our implementation of the system to the problem of fine-grained action recognition are described in Chapter 5. Finally, conclusions and future works are presented in Chapter 6.

Chapter 2

Automatic Construction of Large-scale Video Shot Database using Web data

2.1 Introduction and Related Work

2.1.1 Introduction

In the first stage of action recognition, researches focused only on small-scale and constrained data. The most popular benchmark action datasets around a decade ago were KTH [108] (6 actions), Weizmann [7] (10 actions) or IXMAS dataset [135] (13 actions). Classification rates on these datasets have reached nearly perfect rates. According to a survey of action recognition systems [136] published 3 years ago, 12 out of the 21 tested systems performed better than 90% on the KTH dataset and 3 out of 16 tested systems scored a perfect 100% recognition rate on the Weizmann dataset. However, in fact, these databases do not capture the richness and complexity of real-world actions. A typical video clip in these datasets contains only a single actor with no occlusion and very limited clutter. These datasets are staged, and limited in terms of illumination and camera position variation. Figure 2.1 and Figure 2.2 show some example frames from KTH dataset and Weizmann dataset, respectively. Due to these fairly controlled conditions, these datasets have been considered as being inappropriate for the purpose of learning

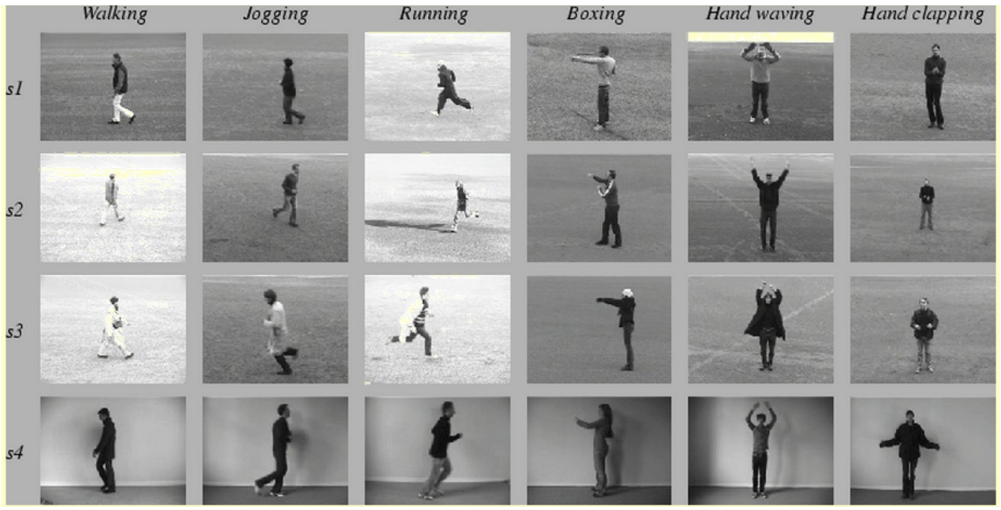

FIGURE 2.1: Thumbnails of actions in KTH dataset [108]. A typical video clip in this dataset contains only a single actor with no occlusion and very limited clutter. The dataset is staged with homogeneous indoor/outdoor backgrounds.

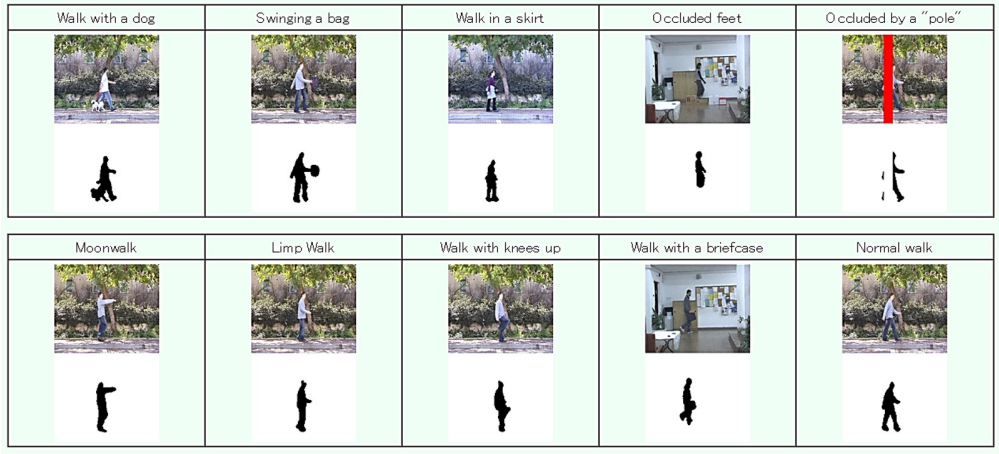

FIGURE 2.2: Thumbnails of actions in Weizmann dataset [7]. This dataset is staged with homogeneous outdoor backgrounds and provides irregular versions (with dog, occluded, with bag, etc.) for robustness experiment for verb "walk".

realistic actions. In other words, action study using these datasets is not supposed to be able to support for real-world vision applications. This emphasizes the importance of building realistic video data with human actions for the training and evaluation of new methods.

In order to increase the applicability of database based action learning, recently some datasets which consist of uncontrolled data have been proposed [99, 77, 53, 96, 61]. See Figure 2.3 for the thumbnails of actions in UCF Sport Action dataset [99] which is the first among datasets constructed using uncontrolled data. As video sources, most

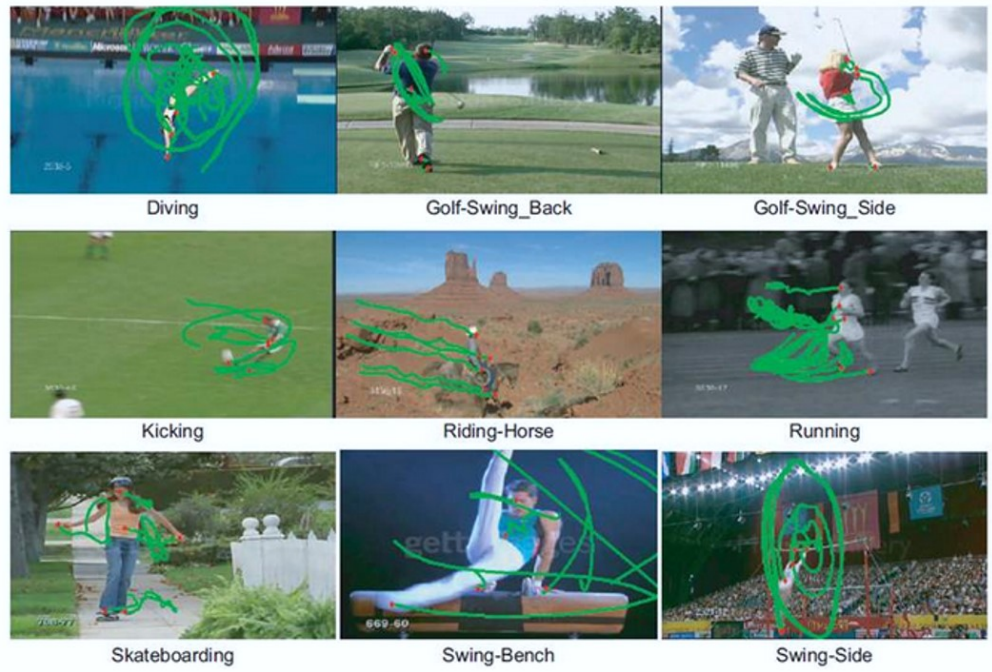

FIGURE 2.3: Thumbnails of actions in UCF Sport Action dataset [99]. Actions in this dataset are collected from various sports in broadcast television channels.

recently released datasets exploit Web data. Web videos with large diversity in terms of actions and large variations in actor appearance, viewpoint, cluttered background, illumination conditions and so on. Therefore, these datasets can be expected to be more challenging for researchers but more suitable for real-world action learning. Moreover, Web videos are extremely numerous and easy to obtain. With several billion videos currently available on the Internet and dozens of video hours uploaded to video sharing sites such as YouTube, Daily Motion every minute, Web video source has become tremendously huge and unstoppably growing data source. By using Web API like YouTube API, we can obtain a large number of videos of various topics from Web data without any difficulties.

Even though recently released action databases videos seem to somehow reflect the real-world, their scale is still limited. While large-scale static image datasets contain thousands of image categories [102], action datasets lag far behind (the largest dataset, UCF101, has 101 action categories [61]). In the image classification field, the breadth of the semantic space has been shown to have important implications. For many real world vision applications, the ability to handle a large number of object classes becomes

a minimum requirement, e.g. an image search engine or an automatic photo annotator is significantly less useful if it is unable to cover a wide range of object classes. Even with the same object class, small number of images can not represent the large variations in object appearance, viewpoint and surrounding conditions. Two important impacts of database scale on recognition performance which have been validated in the image recognition field are: first, a technique that achieves good accuracies on datasets with small number of categories may actually performs not so well on large numbers of categories [24]; and second, for classes with rich training data, simple non-parametric methods can obtain reasonable performance (the bigger the data, the higher the recognition rates) [120]. Action recognition is predicted to share the same tendencies with image recognition. Consequently, there is a rising need to build action datasets which are larger in both number of categories and quantity of video shots for each category. Note that video shots here refer to small fragments of a video obtained by separating it at each point of a scene or camera change. A video shot is supposed to represent for a single action or scene.

However, construction of a large-scale action dataset is a terribly troublesome and time-consuming task due to the noise of video sources which requires much more effort to remove in comparison to the case of images. To collect data for a specific action using a video sharing site such as YouTube, annotators first input action keyword and then manually find relevant video parts from retrieved videos. This retrieval generally depends on text based search. Search engine of the site finds in its database and returns videos with words which are called as tags and considered as being related to the given keyword. Since tags are attached subjectively by the video uploaders which are general users, it is common that tags are sometimes irrelevant to the keyword. Thus the tag based search results may include many unrelated videos (See Figure 2.4 for an example). Moreover, in general, tags are annotated to the whole video sequence, not to specific scenes. Therefore, it cannot be determined which tag corresponds to which part of the video. For example, some videos tagged “eat” might include not only the eating scene but also such other scenes as entering restaurants, ordering foods, or drinking something (See Figure 2.5). People who want to search for eating scenes have to manually skip the scenes of no interest while carefully watching the whole video.

In this chapter, we proposed a method to automatically extract from tagged Web videos relevant video shots of specific actions using metadata as well as visual context of these

FIGURE 2.4: Some videos obtained by searching with "wash hand" keyword on YouTube. We expected videos which contain scenes of people washing their hands like the top one. However, many search results like the other ones have no scenes of interest even though they have "wash hand" as their tags. The second video is a part of a comedy with title as "Employees Must Wash Hands...Before Murder". The bottom video is a video clip of the song named as "Wash your hands too".

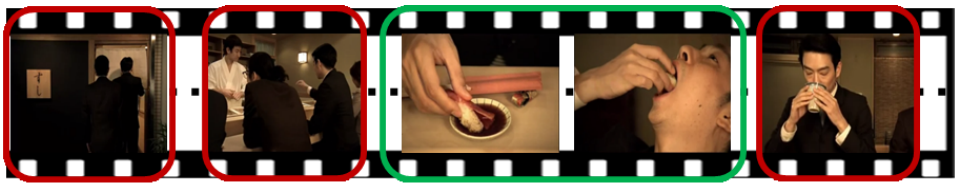

FIGURE 2.5: A video obtained by searching with "eat sushi" keyword on YouTube. It contains scenes of interest (scenes with green bounding box) describing "eat sushi" action as well as irrelevant scenes (scenes with red bounding box) describing actions of entering restaurant, ordering sushi and drinking tea (respectively from the left to the right). Researchers who need only training data for "eat sushi" action must watch the whole video carefully to find its relevant scenes

videos. We reported our work in the journal article [29]. Our unsupervised method requires only the provision of action keywords at the beginning. As for keywords, we mainly focus on words related to human actions. Our list of human action keywords contains sport activities such as "serve volleyball" or "row dumbbell" as well as activities of daily living like "shave mustache" and "tie shoelace". The list also includes some music related activities like "play trumpet" and "dance flamenco" or emotion related activities like "slap face" and "cry" as the consequence of "being angry" and "being sad" respectively. Moreover, we also tried several non-human actions such as "flowers bloom" or "leaves fall". Our main idea is at first, selecting relevant videos among thousands of Web videos for specified action and then, extracting the most related shots from selected videos. The video selection step is based on our assumption that videos tagged with many relevant words have high probability of being relevant videos so they should be

selected. For the extraction of corresponding shots, we apply an unsupervised ranking method called VisualRank [57]. We made large-scale experiments on 100 human action keywords and 12 non-human action keywords. Our system aims to avoid as much as possible the prohibitive cost of manual annotation. The experiment results reflected the effectiveness of our system as we obtained automatically many relevant video shots for many action keywords. Note that here precision is considered as the percentage of relevant shots among top ranked 100 shots (Precision@100).

Furthermore, we proposed to take still Web images corresponding to given actions into account. Our intuition is that the shots with more similarity to related action images have higher probability of being relevant shots, thus they should be biased in shot ranking. In this chapter, we collected images related to the given actions automatically via Web image search engines based only on provided keywords and measure visual resemblances between video shots and selected images. Shots with higher similarity scores will have higher chance to be ranked to the top. Note that these Web images involved processes also do not require any supervision, therefore the automaticity of the whole framework can be preserved. We verified the efficiency of introducing Web images by applying Web images exploited framework on 28 human actions and 8 non-human actions with precision achieved by original framework respectively lower than 20% and 15%. The results demonstrated that exploiting Web action images can significantly improve the performance of the original system.

In the next sections, we first describe the overview of our unsupervised system of extracting relevant video shots from Web videos. We then go to the detail of each proposed methodology in Chapter 3.2. We report the results of the system in Chapter 3.3 and finally, conclude this work in Chapter 3.4.

2.1.2 Related Work

As the first attempt to construct an action training database with minimal manual annotation (and the only one until ours as the best of our knowledge), Laptev et al. [68, 77, 35] proposed methods to automatically associate movie scripts and actions and obtain video shots in movie representing particular classes of human actions. According to their methods, first patterns corresponding to the actions are automatically located in the script by applying OpenNLP toolbox for natural language processing and part

of speech (POS) tagging for identification of nouns, verbs and particles. Then, the temporal localization of human actions and scene descriptions are estimated by matching script text with the corresponding subtitles using dynamic programming. Their first dataset which was built using the above methods, Hollywood [68], released in 2008, provides 8 classes of human actions (AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, and StandUp). The dataset contains 233 video samples with approximately 60% correct labels. Their methods actually can help reduce human effort on construction of realistic action database. However, the targeted videos are only the movies with available scripts and the trainable actions are limited to only actions appeared in movies. On the other hand, our proposed system can be applied to extract data for various types of actions which are distributed over much more immense video source.

Here we refer to some recent work which show that action recognition exploiting still images is possible [106, 144, 119, 134]. Moreover, many works on automatic construction of image database exploiting images gathered from the Web have been carried out so far [140, 40, 39, 141, 70, 107]. As in another related work, Ballan et al. [5] proposed a method to add tags to video shots by using Web images obtained from Flickr as training samples. Meanwhile, Cinbis et al. [21] proposed a method to learn action models automatically from Web images gathered via Web image search engines, and recognize actions for the same video dataset as [84]. Although Cinbis et al. 's work is the most similar to our work, they exploit only Web images and static features as a training source, while Web videos and spatio-temporal features are also adopted in our work.

2.2 Overview of Proposed System

The objective of the proposed system is explained explicitly in Figure 2.6. From abundant Web videos of an action keyword, we exploit their visual features as well as textual information to obtain only relevant video shots of that keyword. Figure 2.7 illustrates the overview of the proposed system. Our system consists of four following processing steps (the third step is optional):

1. Video selection and video-tag relevance calculation

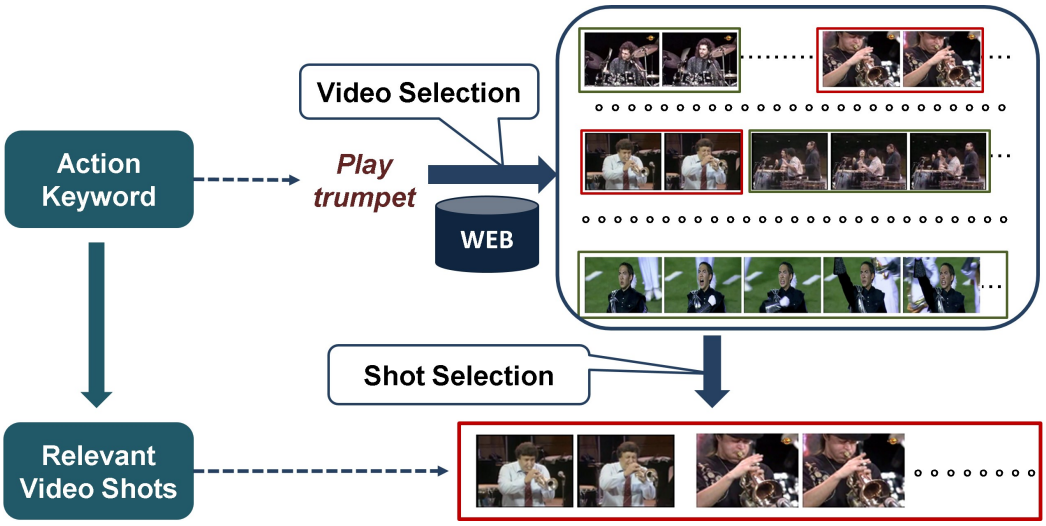

FIGURE 2.6: Illustration of our objective. When we search for videos of a given action keyword (as "play trumpet" in this example) using a video search engine like YouTube, we can obtain bunch of videos including relevant ones as well as irrelevant ones. Even the relevant ones may contain unrelated parts. In this example, playing trumpet is just one section of an instrumental performance. Thus the videos may consist of many irrelevant sections such as playing drum and playing piano. Our objective is to extract only relevant video parts of the given keyword (parts which are surrounded by red bounding box) *in an unsupervised manner*.

2. Shot segmentation and shot similarity measurement

3. Image selection and shot-image similarity calculation (option)

4. Shot ranking

In the first step, video IDs and tags for at most 1000 Web videos of search results for the action keyword are collected via Web API. The co-occurrence frequencies among tags are exploited to build a database of tag relevance information. Then videos are ranked in the descending order of their tag relevance scores with the keyword. Only the top ranked videos are downloaded since they are considered as action related videos. Meanwhile, the relevance scores of the videos to their tags are also calculated by similar way to calculate relevance of videos to the keyword.

In the second step, the downloaded videos are segmented into video shots using color information. Spatio-temporal features are extracted from all shots and used to calculate similarity matrix of shots.

The third step is an option. In this step, firstly, hundreds of top results of image search for given action keywords are downloaded using Bing API. Then, Web action images

are automatically selected based on human detection method. Finally, similarity scores between shots and images are measured according to their static features. Note that human detected images are selected and images with no human detected are discarded only in case of human actions. In case of non-human actions, images directly retrieved by Bing API are adopted. The third step can be performed in one of two modes: for shots and images, (1) SURF features are extracted, and shot-to-image similarities are measured using feature matching. (2) Simple but efficient pose features which simulate the orientation of human body parts are extracted, and shot-to-image similarities are measured by comparing their pose features.

Both modes can be applied to human actions while the first mode is restricted to non-human actions only. Note that as for shots, we do not extract pose features from all of their frames but only one frame at every second since normally there is no significant change in one second. This also helps to reduce the cost of calculation.

In the final step, we rank video shots by VisualRank [57] which originally is an image ranking method with a visual-feature-based similarity matrix and a bias damping vector based on tag-based video relevance scores. In the end, we can obtain video shots corresponding to the given keywords in the upper rank of the video shot ranking results.

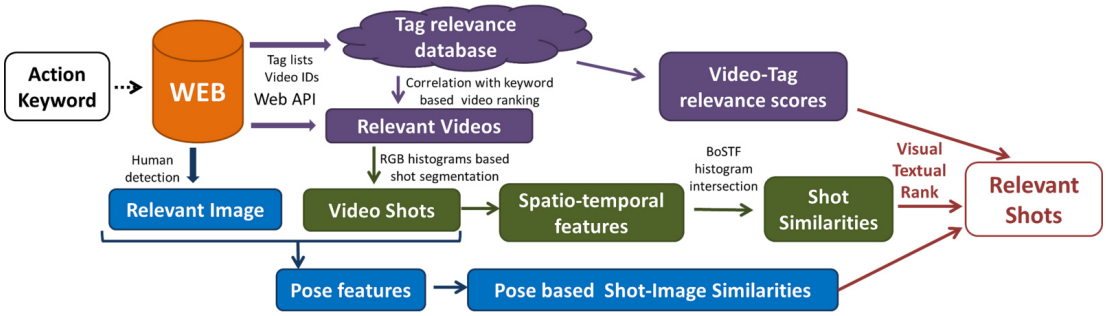

FIGURE 2.7: Overview of unsupervised system of extracting corresponding video shots for specific actions from Web videos. We modified our previous system[26] by introducing Web images and our proposed ranking method (VisualTextualRank) to enhance shot ranking process. For the detail of VisualTextualRank, please refer to Chapter 3.

2.3 Methodologies

2.3.1 Tag-based Video Selection

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

First, the given keywords are sent to the Web API to collect sets of video IDs and tags. Then, the relevance scores of Web videos to the given keyword are calculated according to co-occurrence relationships between their tags. To this end, we apply the "Web 2.0 Dictionary" method proposed by Yang et al. [143] with some modifications in relevance measurement. "Web 2.0 Dictionary" corresponds to statistics on tag co-occurrence, which we need to construct in advance using a large number of tags gathered from the Web. This method is based on an idea that tags other than the query are supporters of the query, and the query can be regarded as being relevant to a video whose tags are its strong supporters.

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 tagged to all the Web videos. The correlation of parent word t and its child word $t_i \in \mathcal{T}$ is defined as

$$w(t, t_i) = \frac{F(t, t_i)}{N(t)} \tag{2.1}$$

where $F(t, t_i)$ is the number of videos tagged with both word t and word t_i at the same time. Let $\mathcal{T}_V$ represent a set of tags for video V excluding t, we estimate relevance score 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:

$$\begin{aligned} P(V|t) &\propto P(\mathcal{T}_V|t) \\ &= \prod_{t_i \in \mathcal{T}_V} P(t_i|t) \\ &= \prod_{t_i \in \mathcal{T}_V} w(t, t_i) \end{aligned} \tag{2.2}$$

The above equations to calculate relevance of an image video to the given keyword are obtained by applying [143]. However if we multiply all the correlation values between the query tag and the rest of the tags within one video, the value of Equation 2.2 becomes smaller as the number of tags increases. To prevent this, we modify Equation 2.2 so that the number of co-occurrence words used for calculation is limited to m at most, and define the relevance score $Sc_t(V)$ using average log likelihood as follows:

$$\begin{aligned} S(V|t) &= \frac{1}{n}\sum_{t_i \in \mathcal{T}'} \log_2 w(t, t_i) \\ &= \frac{1}{n}\sum_{t_i \in \mathcal{T}'} (\log_2 F(t, t_i) - \log_2 N(t)) \\ &= \frac{1}{n}\sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) - log_2 N(t) \qquad (2.3) \\ Sc_t(V) &= \frac{1}{n}\sum_{t_i \in \mathcal{T}'} \log_2 F(t, t_i) \qquad (2.4) \end{aligned}$$

where $\mathcal{T}'$ contains at most the top m word t_i in the descending order of $w(t, t_i)$, and n ($n \leq m$) represents $|\mathcal{T}'|$. Since the second term of Equation 2.3 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 Equation 2.4. In the experiment, we set m as 10, and select the most 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 allow only promising videos to go to the next step which requires more costly processes such as feature extraction and similarity calculation.

Note that in case of compound keywords such as "drink coffee", we regard $N(t)$ as the number of the videos including all of the element word of the compound keyword in their tag sets and $w(t, t_i)$ as the number of videos having all the words of t and t_i even if t_i is also a compound word. We ignore videos which do not have any co-occurrence tag since we can not calculate their relevance scores.

In the experiments, as seed words, we prepared 150 sets of verbs and nouns which are related to such actions as "ride bicycle" or "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 "our Web 2.0 Dictionary" by counting tag co-frequencies according to Equation 2.1.

2.3.2 Relevant Shot Extraction

With obtained videos, we segment them into video shots (Chapter 2.3.3.1) and apply VisualRank to the shots with our assumption that the top ranked shots are the relevant ones. VisualRank [57] is an image ranking method based on the widely known Web page raking method, PageRank [10]. 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, a page linked to more pages have much ranking point becomes higher. VisualRank uses a similarity matrix of images instead of hyper-link structure. Equation 2.5 represents how to compute VisualRank.

$$\boldsymbol{r} = \alpha S\boldsymbol{r} + (1-\alpha)\boldsymbol{p} \quad (0 \leq \alpha \leq 1) \tag{2.5}$$

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 Equation 2.5. 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 $\boldsymbol{r}$ corresponds 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 $\boldsymbol{r}$. Heavenward [46] 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. We experimented both following definition ways of $\boldsymbol{p}$:

$$\boldsymbol{p}_i^{(1)} = [1/n] \tag{2.6}$$

$$\boldsymbol{p}_i^{(2)} = \frac{\exp(\mathrm{SI}(S_i))}{\sum_{j=1}^{n}\exp(\mathrm{SI}(S_j))} \tag{2.7}$$

The uniform damping vector presented in Equation 2.6 is used when we do not employ optional third step. The nonuniform damping vector presented in Equation 2.7 is used when we take Web images into account. So that in this case video shots has similar visual characteristics with corresponding images will be biased during ranking computation. $\boldsymbol{p}_i^{(2)}$ is proportional to corresponding shot-image similarity score $\mathrm{SI}(S_i)$. For the computation of shot-image similarity scores, please refer to Chapter 2.3.4.

2.3.3 Shot-Shot Similarity Matrix Calculation

In this subsection, we describe how to estimate the similarity matrix which appears in Equation 2.7. In our work, this similarity matrix holds ST feature based similarity scores between shots. We first divide each downloaded video into several shots and extract ST features from all the shots. We then represent each shot as a Bag-of-Spatio-Temporal-Features (BoSTF) histogram and calculate similarity between shots as their histogram intersection.

2.3.3.1 Shot Segmentation and Selection

After downloading the most relevant 200 videos to the given keyword regarding tag relevance scores, we segment downloaded videos into video shots based on their RGB histograms. We simply calculate 64 dimensional RGB histogram for each frame and record one segmentation point between two consecutive frames if their histogram intersection is larger than our predefined threshold. As the result, we obtain 10 shots per video on average. However, there are some shots whose duration is too short or too long. It is hard for us to recognize what happens in a shot which lasts too short. In contrast, excessively long shots are supposed to be uninformative since there is no significant change in them. We consider a shot as too short one if its duration is smaller than one second, or too long one if it lasts more than one minute. Thus we select only shots which last longer than one second and shorter than one minute. To make computational cost feasible, in the experiment, we set the upper limit number of shots to go to the next step as 2000. If shot number exceeds 2000, we select only 2000 shots according to the heuristic manner summarized by Equation 2.8 which intends to balance selecting more shots from the higher-ranked videos against selecting various shots from as many videos as possible.

$$
\begin{aligned}
N_{upper}(V_i) &= c \times Sc(V_i) + f(N(V_i)) \qquad (2.8)\\
\text{where} f(x) &= \begin{cases} 20 & (20 \leq x) \\ 20 + (x-20)/4 & (20 < x < 100) \\ 40 & (x \geq 100) \end{cases}
\end{aligned}
$$

$N_{upper}(V_i)$ 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 features from the selected shots as described in the next subsection.

2.3.3.2 Feature Extraction

Following the method described in Noguchi and Yanai's work [85], firstly, interest points are detected using the SURF method [47], and then moving interest points are selected applying the Lucas-Kanade method [75]. Since ST features are supposed to represent movements of objects, only moving interest points are considered as ST interest points and static interest points are discarded. After detection of ST interest points, triples of interest points which hold both local appearance and motion features are formed applying Delaunay triangulation. Then changes of flow directions of interest points as well as the sizes of the triangles are tracked within five consecutive frames. This tracking enables us to extract ST features not from only one point but from a triangle surface patch. Thus the features are expected to be more robust and informative. The ST features are extracted from every five frames. This method of ST feature extraction is relatively faster than the other methods such as cuboid based method, since it employs

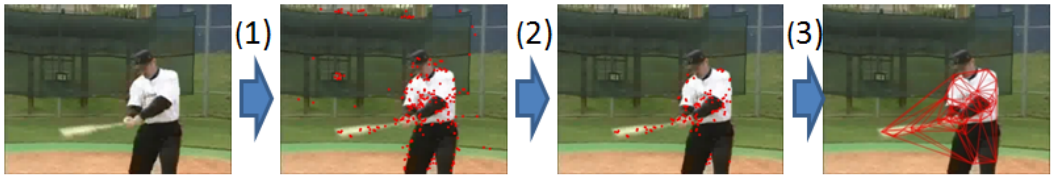

FIGURE 2.8: Steps to extract the ST feature. (1) detecting SURF points, (2) detecting SURF points with motion, and (3) applying Delaunay triangles. (Cited from [85])

SURF detector [47] and Lucas-Kanade detector [75] which are comparatively fast detectors. Figure 2.8 shows an example of the process for extracting the ST features from a video shot of action "batting".

2.3.3.3 Calculation of Shot-to-Shot Similarity Matrix

To apply VisualRank ranking method to the shots, we need to compute the similarities among all the shots to find out the shots sharing the most visual characteristics with others. To this end, we first vector-quantize them and convert them into BoV vectors. While the standard BoV represents the distribution of local features within one image, the BoV employed in this chapter represents the distribution of features within one shot which consists of several frame images. We call our BoV as Bag-of-Frames (BoFr). In the experiment, we set the size of the codebook as 5000.

The similarity between two shots is measured as their histogram intersection:

$$s(H_i, H_j) = \sum_{l=1}^{|H|} min(h_{i,l}, h_{i,l}) \tag{2.9}$$

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

2.3.4 Calculation of Shot-Image Similarity based Damping Vector

Remind that employing Web images is an optional step based on our intuition that the shots which are more similar to corresponding action images have higher probability of being relevant shots. So the idea here is to select action images from Web images, calculate the similarities between shots and images, and then bias the shots with high similarities in the shot ranking step.

2.3.4.1 Image Selection

When an action keyword is queried on a Web image search engine, thousands of images might be returned. However, in general, even top results may be not relevant images of the queried action due to the wide variety of keyword's meaning as well as the action itself, especially in the case of human action. Here we want to filter the returned results of Web image search engine so that the fewer irrelevant images the better. On the other hand, we also want to preserve the automaticity of our framework, thus manual selection is not preferred here. We postulate two assumptions: (1) the set of retrieved images contains relevant images of the queried action and (2) humans or body parts should be seen in human action images.

It is reasonable to consider that in case of human actions, images which contain humans are more likely related images than images in which humans do not appear. Based on these assumptions, we select a collection of action images by applying a human detection method [8, 145] on Web images. For non-human actions, we simply select the first images returned by Web search engine and evaluate shot-image similarities by local feature matching (See Chapter 2.3.4.3). Note that in the first proposed mode of shot-image similarity calculation, we only care if images contain humans or not and compute similarities between human detected images and shots based on SURF matching. On the other hand, the second mode requires more detailed analysis of human movements and adopts human pose estimation method (See Chapter 2.3.4.2 and Chapter 2.3.4.4).

In the first mode, we use Poselets method [8] to detect humans. Poselets are demonstrated as effective body part detectors trained by 3D human annotations. We apply Poselets detector tools which are officially offered by the authors[1] on the set of retrieved Web images using default parameters. Figure 2.9 illustrates some examples of selected Web images using Poselets-based human detection.

Note that as shown in our previous work [27], the appropriate number of images to use in shot-similarity calculation step should be 20 to 30. Here we use 30 first human detected images.

[1] http://www.cs.berkeley.edu/%7Elbourdev/poselets/

FIGURE 2.9: The top six Web images after Poselets-based image filtering.

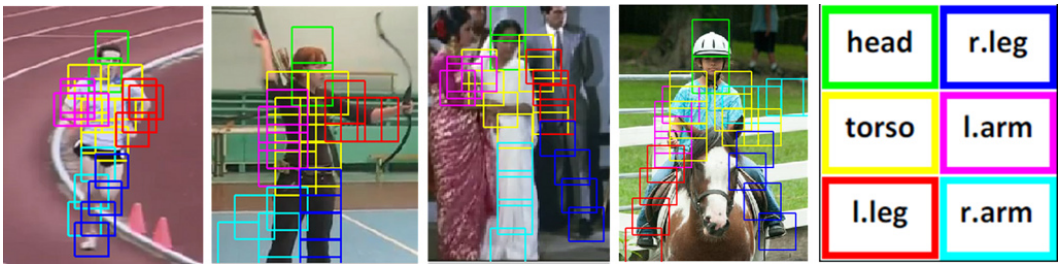

FIGURE 2.10: Examples of pose detection results by full body model

2.3.4.2 Pose Feature Extraction

In case of human action recognition, not only low-level features such as SURF and our proposed spatio-temporal feature but high-level features like human pose should be also adopted. Even though actions may depend on actors or situations which they are taken, the basic poses for humans to perform them in general are similar. Based on this idea, we extract features of human poses detected in shots and images, and compare poses using these features. We suppose that the similarity calculation based on pose comparison can achieve better performance than local-feature-matching-based calculation.

As for the characteristics of a pose, we pay attention to relations of body parts' orientation or in other words, to their connection. We apply pose estimation models proposed by Yang et.al [145] which are flexible mixture models for capturing contextual co-occurrence and spatial relations between body parts. For each pose, their full body model[2] detects 26 human body elements where 2 elements correspond to head, 4 elements relate to each limb and 8 elements point out torso (See Figure 2.10).

[2] http://phoenix.ics.uci.edu/software/pose/

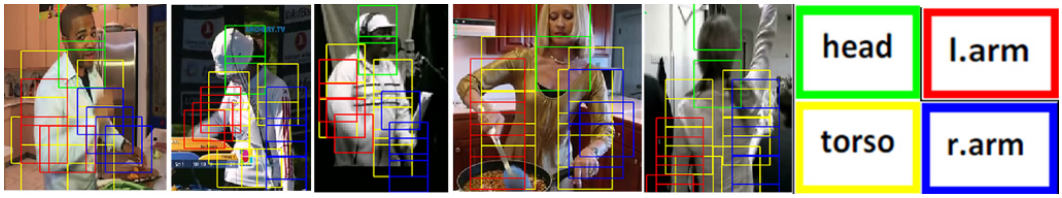

FIGURE 2.11: Examples of pose detection results by upper body model

Since our action category list contains some actions like "play piano" or "eat" which are most frequently taken when only upper bodies of actors appear, we also employ upper body model. In case of upper body pose estimation, the upper body model detects 2 elements of head, 4 elements of each of 2 arms, and 8 elements of torso (See Figure 2.11).

From each detected pose, we simply extract inner orientation and correlation orientation of its parts as its features. Inner orientation here is defined as direction of a body part such as an arm or a torso. Correlation orientation here refers to spatial relations between a pair of body parts such as a head and a leg. Following is how we calculate inner orientation and correlation orientation.

$$\begin{aligned} O_{in}(P) &= [dx_1 \;\; dy_1 \quad . \;\; . \;\; . \quad dx_{n-1} \;\; dy_{n-1}], \\ &\quad \text{where} \;\; dx_i = x_i - x_{i+1}, \; dy_i = y_i - y_{i+1} \qquad (2.10) \\ O_{co}(P_i, P_j) &= [X_{P_i} - X_{P_j} \;\; Y_{P_i} - Y_{P_j}] \qquad (2.11) \end{aligned}$$

where $O_{in}(P)$ means inner orientation of part P and $O_{co}(P_i, P_j)$ refers to correlation orientation between part P_i and part P_j. (x_i, y_i) represents position of element i of part P which has n elements. (X_P, Y_P) is defined as center position of part P. Finally, for each detected pose, we obtain a 70 dimensional feature. Note that for an image or a shot frame, first we apply the full body model. If the full body model fails to detect human pose, we then try the upper body model. If the upper body model succeeds to detect an upper pose, we calculate its orientation except for leg related orientation which will be regarded as 0. This enables us to compare poses even in case that they are detected by different body models.

2.3.4.3 Local Feature Matching Based Shot-to-Image Similarity Calculation

For shot-image similarity calculation, we first extract SURF local features [47] from all action images of selected set and each one frame per five consecutive frames of

all the shots. For each shot, we count matching points between SURF local features extracted from each frame and each Web image by thresholding Euclidean distances between SURF feature vectors. The similarity $\mathrm{SI}(S_i)$ between a shot S_i which has M frame images ($F_j(j = 1..M)$) and an image set $\mathcal{I}$ which has N images ($I_k(k = 1..N)$) is calculated by the following equations:

$$\mathrm{SI}(S_i) = \sum_{k=1}^{N} \max_{j=1} \mathrm{SI}(F_j|I_k), \tag{2.12}$$

$$\text{where } \mathrm{SI}(F_j|I_k) = \frac{2 * \mathrm{MatchPoint}(F_j, I_k)}{(\mathrm{Point}(F_j) + \mathrm{Point}(I_k))}, \tag{2.13}$$

$\mathrm{MatchPoint}(F_j, I_k)$, $\mathrm{Point}(F_j)$ and $\mathrm{Point}(I_k)$ represent the number of matched points between a frame image F_j and a Web image I_k, the number of extracted SURF features from F_j and the number of extracted SURF features from I_k, respectively.

2.3.4.4 Pose Comparison Based Shot-to-Image Similarity Calculation

Like the above mode of shot-image similarity calculation, the similarity between a shot and a set of images is regarded as the similarity of its frame with the highest similarity score, and the similarity between a frame and a set of images is equal to normalized total similarity of that frame to all images in the set. Here we simply define pose comparison based similarity between a frame and an image as Euclidean distance between the poses. However, in case of comparison between the upper body pose and the full body pose, we disregard leg associated elements. That means we only compare upper parts of the poses in this case. Moreover, since calculation of distance between two full poses will result in higher value than other cases due to extra leg related distance, we normalize it as follows:

$$\mathrm{SI}'(\mathrm{F|I}) = \mathrm{SI}(\mathrm{F|I}) * \frac{\text{number of elements unrelated to legs}}{\text{total number of elements}} \tag{2.14}$$

In this calculation of ours, the number of orientation elements unrelated to legs and total number of orientation elements equal to 40 and 70, respectively.

2.4 Experiments and Results

To examine effectiveness of the proposed system, we conducted various experiments under different conditions with 100 kinds of human action keywords and 12 kinds of non-human action keywords. We first explain our evaluation method and describe briefly about our experimental settings. Each experiment and its results will be expressed in detail in the next subsections.

2.4.1 Experimental Settings and Evaluation Method

In our experiments, we used YouTube videos as our data source. We collected video metadata including video IDs and tags using YouTube Data API. To examine the effectiveness of our proposed method, we make large-scale experiments on 100 human action categories and 12 non-human action categories with video metadata analysis on 112,000 YouTube videos and spatio-temporal feature analysis on 22,400 YouTube videos. In each experiment, we obtained rankings of 2000 shots in average for each action, since as we mentioned above, we downloaded 200 videos for each action and each video is segmented into 10 shots in average. For the evaluation of recognition results, average precision is widely used. However, here we use the precision rate over top ranked 100 shots since we expect that they are qualified to be used for action database construction while commonly used datasets such as KTH dataset [108] and "in-the-wild" YouTube dataset [71] have approximately 100 video shots per action[3]. That means in each experiment, we simply count the number of relevant shots among 100 top ranked shots NR and the precision achieved in that experiment is computed as $NR/100$.

We carried out 3 experiments with following settings. We reported the results of the first experiment in our conference paper [26]. The results of the other two can be found in [27].

(1) Exp.1: Original Framework (without optional step)

(2) Exp.2: Framework adopts Web images with local feature matching based shot-similarity calculation method

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

(3) Exp.3: Framework adopts Web images with pose comparison based shot-similarity calculation method

The objective of Exp.1 is to verify the performance of the original framework without optional step which takes Web images into account. On the other hand, Exp.2 and Exp.3 show the effectiveness of adopting Web images.

2.4.2 Performance of the Original Framework

The purpose of the first experiment is to validate our original framework when Web action images are not taken into account. We call this experiment as Exp.1. This means in Exp.1, shot selection step involves only spatio-temporal features and biases the top k shots regarding tag relevance scores (Equation 2.6). We conduct Exp.1 on our full action category set which consists of 100 human action categories and 12 non-human action categories. The results for human actions and non-human actions are summarized in Table 2.1 and Table 2.2, respectively.

As shown in Table 2.1, the mean of the precision at 100 shots over 100 human actions was 36.6%, and the precision varies from 2 to 100 depending on each action category. Top 34 actions regarding precision obtained 66 relevant shots among top ranked 100 shots in average and 14 actions achieved precision higher than 70%. Figure 2.12 shows some example results of some of successful action categories. However, the original framework failed to extract relevant shots for some actions (Figure 2.13). In the case of "boil egg", some shots are actually related to "egg" but few of them describe exactly "boil egg" action. In cases of actions like "smile", the action itself is too ambiguous to recognize. "Smile" is one of facial expressions which are mostly researched by emotion recognition works. Our proposed original framework cannot distinguish "smile" and other facial actions. As for action keywords like "jog", we could not select relevant videos of theirs due to tag noise as well as the variety in meaning of the keywords. Downloaded videos of "jog" mainly consist of videos about TV shows, movies or even motorbikes called as "jog".

As for non-human actions, we obtained 14.9% as average precision. While some categories like "flower blooming" or "tornado" obtained quite a number of relevant shots at the top, some categories such as "leaves falling" and "waterfall" detected just very few

Table 2.1: Precision@100 of 100 human actions (%)

soccer+dribble	100
fold+origami	96
crochet+hat	95
arrange+flower	94
paint+picture	88
boxing	86
jump+parachute	82
jump+trampoline	82
do+exercise	79
do+aerobics	78
do+yoga	77
surf+wave	75
shoot+arrow	73
massage+leg	72
fix+tire	67
batting	66
basketball+dribble	64
blow-dry+hair	64
knit+sweater	64
ride+bicycle	62
curl+bicep	58
shoot+ball	58
tie+shoelace	57
laugh	50
dive+sea	49
harvest+rice	49
ski	49
iron+clothes	47
twist+crunch	47
dance+flamenco	45
dance+hiphop	43
eat+ramen	42
dance+tango	41
play+trumpet	41
AVG. (1-34)	**65.9**

play+drum	40
skate	37
swim+crawl	36
cut+hair	35
run+marathon	35
count+money	33
paint+wall	33
shoot+football	33
draw+eyebrows	32
fieldhockey+dribble	32
hit+golfball	32
lunge	32
play+piano	32
row+boat	32
sing	32
chat+friend	31
clean+floor	31
cut+onion	31
shave+mustache	31
pick+lock	30
plaster+wall	30
blow+candle	29
wash+face	29
walking+street	29
brush+teeth	28
catch+fish	28
drive+car	28
plant+flower	28
play+guitar	28
lift+weight	27
raise+leg	27
hang+wallpaper	26
jump+rope	26
AVG. (35-67)	**31.0**

climb+tree	24
ride+horse	24
roll+makizushi	24
sew+button	24
fry+tempura	23
slap+face	20
read+book	19
squat	19
row+dumbell	16
wash+clothes	15
wash+dishes	15
comb+hair	14
drink+coffee	14
swim+breaststroke	13
cry	12
eat+sushi	12
serve+teniss	11
tying+tie	11
boil+egg	9
head+ball	9
swim+backstroke	9
take+medicine	8
serve+volleyball	7
swim+butterfly	7
bake+bread	6
cook+rice	6
grill+fish	5
jog	5
slice+apple	5
peel+apple	5
bowl+ball	4
smile	4
kiss	2
AVG. (68-100)	**12.2**
AVG. (ALL)	**36.6**

Table 2.2: Precision@100 of 12 non-human actions (%)

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

leaves +falling	snow +falling	typhoon	heavy +rain	waterfall	explosion	**AVG.**
3	14	4	0	5	0	**14.9**

relevant shots (Figure 2.14). In fact, for "leaves falling" or "waterfall" categories, most of collected videos are unrelated to the actions. The main reason is that tag noise led to the failure in relevant video selection.

2.4.3 Effectiveness of Exploiting Web Images

To examine the efficiency of introducing Web action images, we validate our modified system including the optional step on 28 human action categories and 8 non-human action categories which showed the lowest precision in the first experiment. Note that the local feature matching based framework (Exp.2) can run on both human actions and non-human actions while pose comparison based mode works (Exp.3) can only run on human actions. We show results of these experiments for human actions and non-human actions in Table 2.3 and Table 2.4 respectively. For human actions dataset, we want to evaluate the effectiveness of adopting Web action images and compare two modes of shot-similarity calculation.

As shown in Table 2.3, introducing Web images helps to enhance the performance for human actions by 6.2% and 8.8% in average in case of exploiting local feature matching mode and pose matching mode respectively. For non-human actions, experimental results (Table 2.4) demonstrate that by introducing Web images into shot ranking, we can improve the precision from 4.4% to 18.6% in average. That means even in case where the tag noise led to the selection of irrelevant videos, our proposed method still can extract from those videos a number of action related video shots. Figure 2.15 and Figure 2.16 respectively shows some relevant shots which were detected by taking Web images into account in case of human actions and non-human actions.

We realized that local feature matching based method improved the performance in average but degraded it in cases of several categories such as "slap face", "wash clothes" and "comb hair". On the other hand, exploiting shot-to-image similarity measurement based on pose comparison not only obtained the highest precision in average but also outperformed Web images unexploited framework for most actions except for "swim" related ones. In case of "swim", human pose estimation failed to detect humans in water, hence obtained shots are mostly human detected shots such as medal rewarding, interviewing. (Figure 2.17). These results match with our expectation that in general,

TABLE 2.3: Results of 28 human action categories depending on how to exploit Web images. Exp.1: Web images unexploited, Exp.2: Web images + local feature matching exploited, Exp.3: Web images + Pose matching exploited

Action	Exp.1	Exp.2	Exp.3
slap+face	20	13	36
read+book	19	23	22
squat	19	32	37
row+dumbbell	16	24	33
wash+clothes	15	10	31
wash+dishes	15	25	40
comb+hair	14	12	20
drink+coffee	14	9	19
swim+breaststroke	13	31	11
cry	12	5	5
eat+sushi	12	11	15
serve+tennis	11	15	24
tie+necktie	11	23	24
boil+egg	9	6	14
head+ball	9	7	7
swim+backstroke	9	14	3
take+medicine	8	7	8
serve+volleyball	7	31	35
swim+butterfly	7	31	14
bake+bread	6	18	18
cook+rice	6	15	16
grill+fish	5	26	26
jog	5	21	10
pick+apple	5	9	2
slice+apple	5	2	13
bowl+ball	4	15	17
smile	4	18	26
kiss	2	3	3
Average	10.1	16.3	18.9

TABLE 2.4: Results of 8 non-human action categories of experiment on validating effectiveness of Web image introduction. Exp.1: Web image unexploited, Exp.2: Web image exploited (local feature matching based similarity calculation)

Action	Exp.1	Exp.2
explosion	0	5
falling+leaves	3	16
snow+falling	14	22
typhoon	4	29
airplane+flying	2	32
earthquake	7	25
heavy+rain	0	3
waterfall	5	17
Average	4.4	18.6

for human action learning, human poses hold very informative clues that should be exploited (Figure 2.18) and applying human pose matching to measure similarities between human action images can achieve better results than using low-level features only.

To confirm this hypothesis, we further conducted more experiments on other human action categories using pose matching between video shots and images introduced framework. We selected randomly 17 human action categories from actions which showed precision higher than 20% but lower than 35% in image unexploited framework. As expected, the performance was remarkably improved as it rose from 26.8% to 36.8% in average and the full system outperforms Web images unexploited system in most of categories. The results are summarized in Table 2.5 and result examples are shown in Figure 2.19.

TABLE 2.5: Results of 17 human action categories of experiment on validating effectiveness of proposed pose matching method

Action	Exp.1	Exp.3
blow+candle	29	35
clean+floor	31	38
jump+rope	26	39
roll+makizushi	24	26
sew+button	24	40
drive+car	28	35
ride+horse	24	35
catch+fish	28	45
play+guitar	28	38
shave+mustache	31	28
chat+friend	31	38
draw+eyebrows	32	35
play+piano	32	27
plaster+wall	30	38
brush+teeth	28	34
row+boat	32	28
wash+face	29	30
Average	28.6	36.8

2.5 Conclusions

In this chapter, we proposed a method of automatically extracting from Web videos video shots corresponding to specific actions by only providing action keywords. To the best of our knowledge, we are the first to aim at automatic construction of such a large-scale database for action recognition. The empirical results showed that using the proposed framework, we could obtain remarkable number of relevant video shots for many of experimented action classes. Nevertheless, the performance of proposed

framework depends on the action categories and the efficiency of exploiting images. For example, precision rates of the best 24 and 35 actions exceeded 50% and 40%, respectively, by using the original framework which does not exploit Web images. On the contrary, introducing Web image based shot bias into shot ranking degraded performance of the system on some categories. However, for other categories, exploiting action images helped enhance significantly performance of the system. Particularly, exploiting proposed shot-image pose matching method improved precision rates of most of experimented human categories.

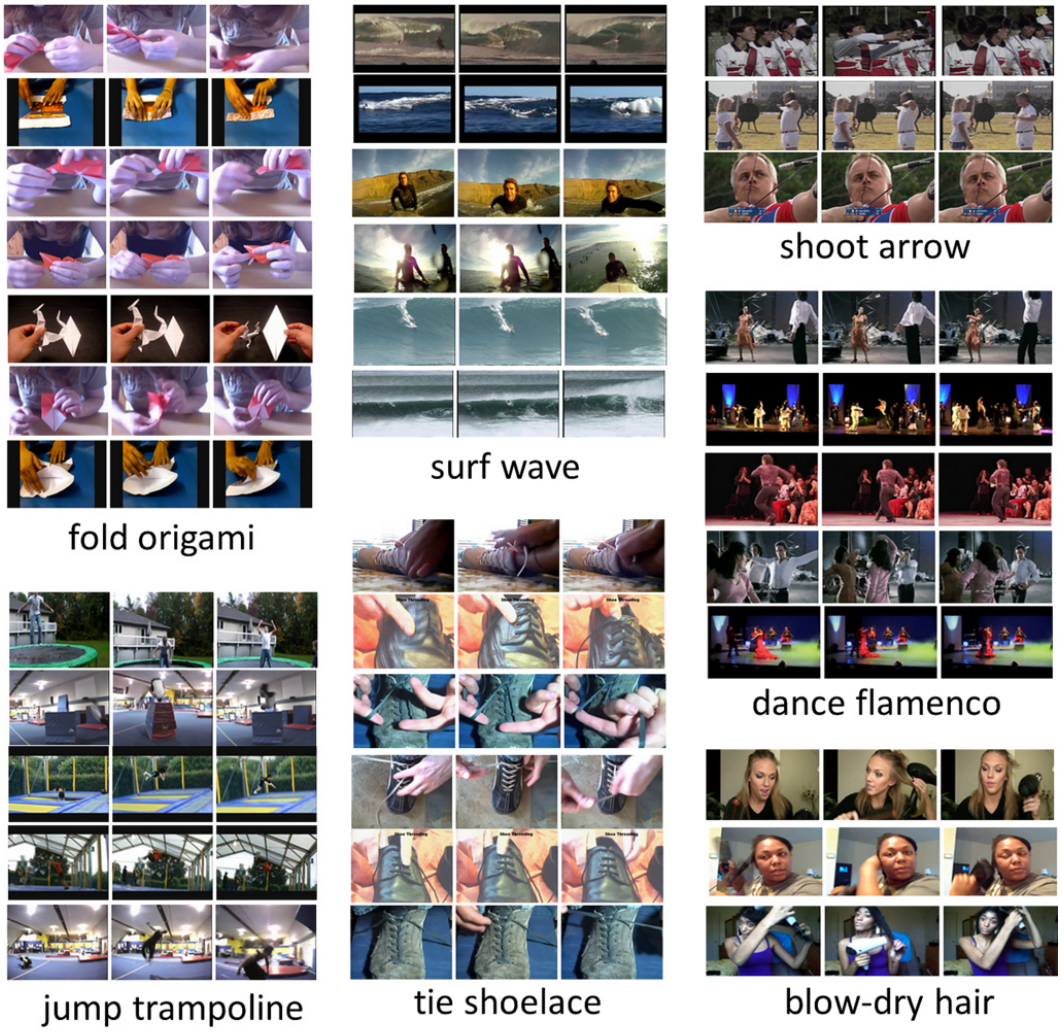

FIGURE 2.12: Relevant shots obtained in top 10 ranked shots of some categories which achieved high precision. Many of relevant shots are boosted to the top in these cases.

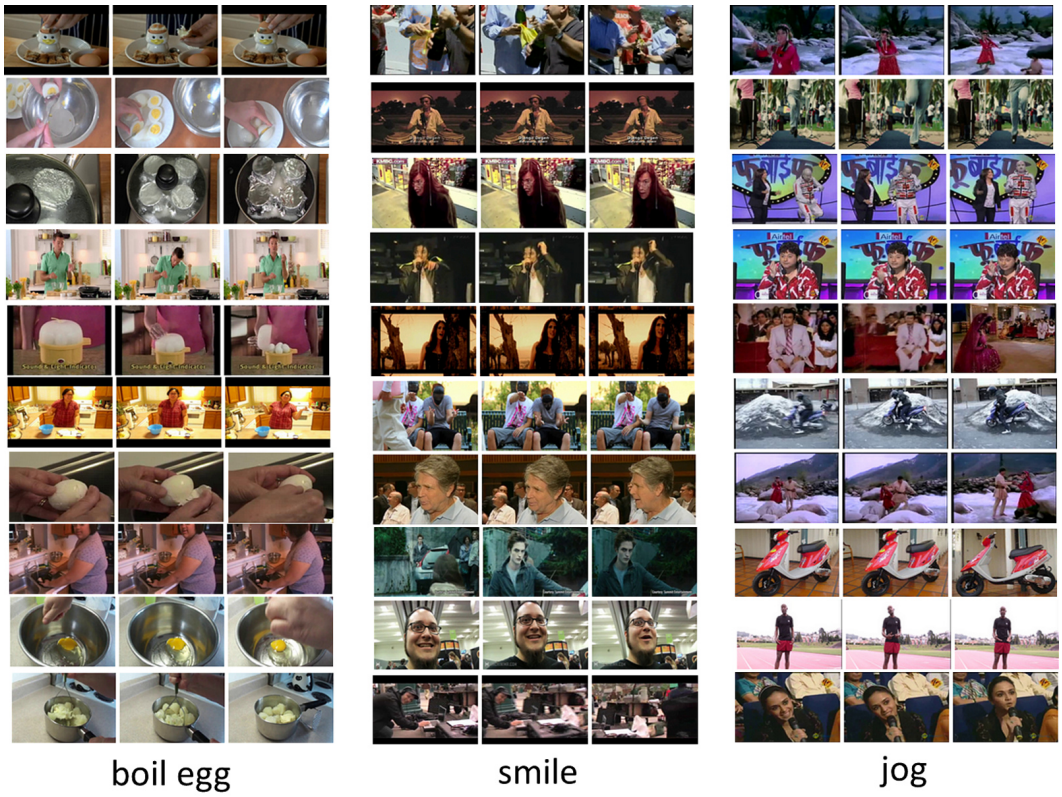

FIGURE 2.13: 10 shots among top 30 ranked shots of some low precision categories. As for "boil egg", "eggs" app ear in many shots but few shots describe exactly "boil egg" action. Especially, single action keywords such a s "smile" or "jog" are too ambiguous to obtain good candidate videos.

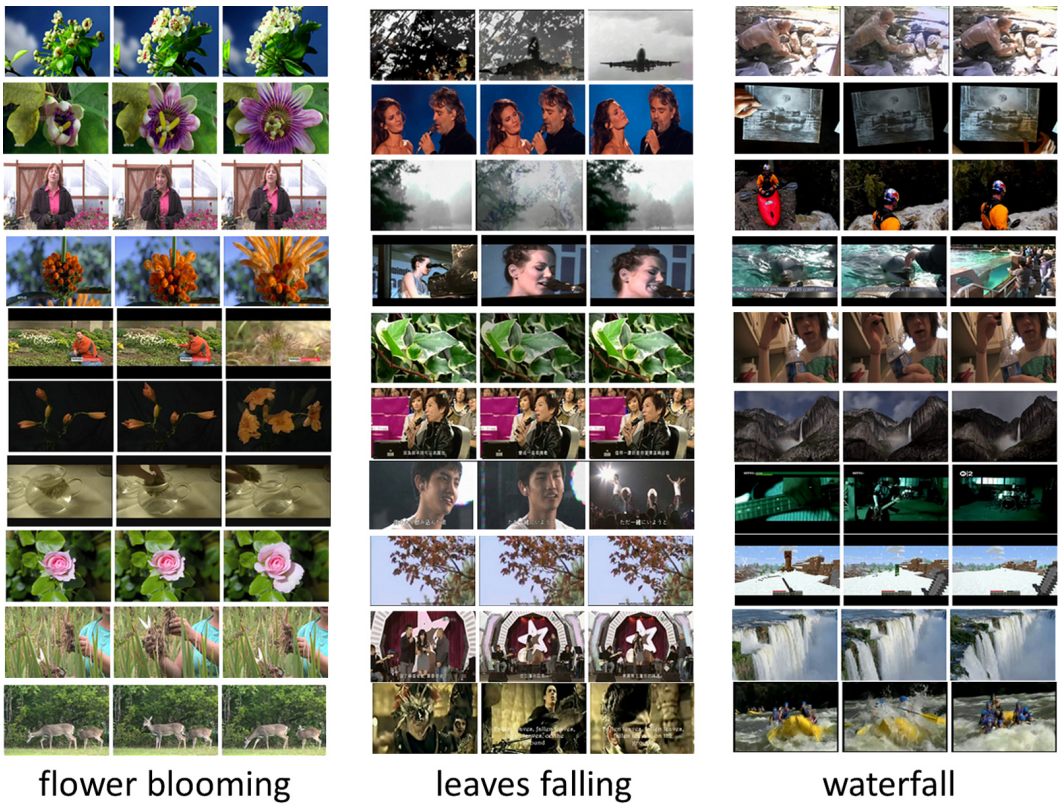

FIGURE 2.14: 10 shots among top ranked 50 shots. Nearly half of shots for "flower blooming" are expected shots. In the cases like "leaves falling" or "waterfall" tag noise caused selection of irrelevant videos. Particularly, "leaves falling" became tag of many music related clips so most of downloaded videos are not related to "leaves falling" scene but to music.

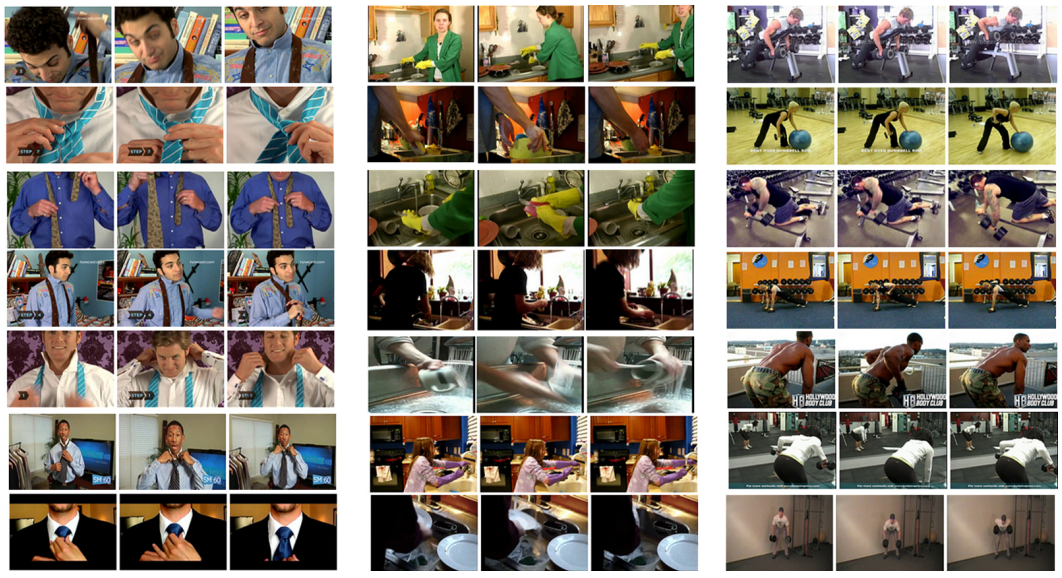

FIGURE 2.15: Some relevant shot that framework without optional step failed to detect were obtained by introducing Web images for human actions.

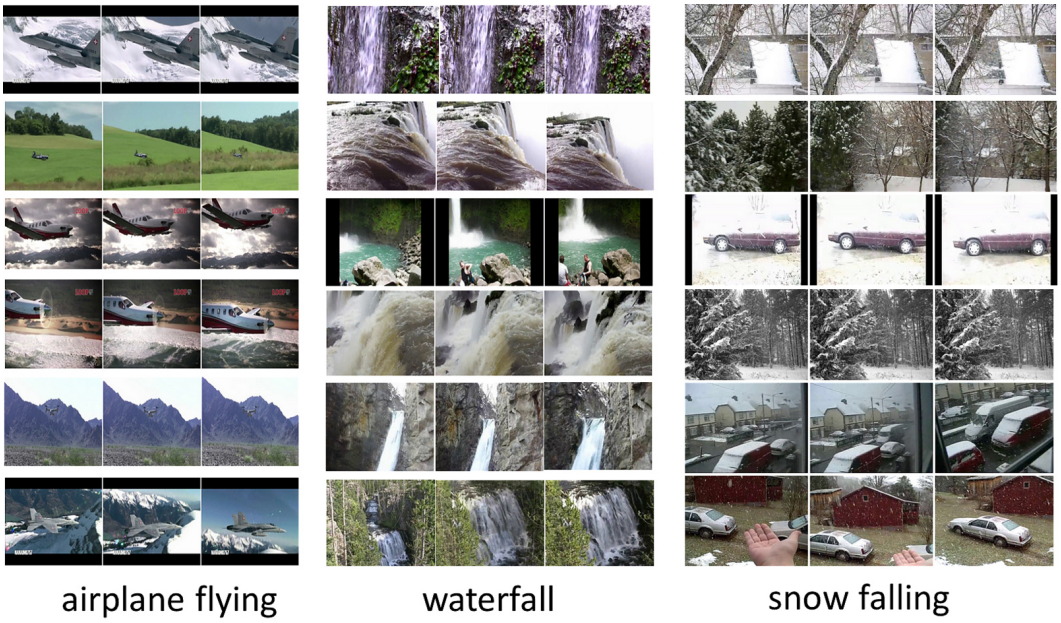

FIGURE 2.16: Some relevant shot that framework without optional step failed to detect were obtained by introducing Web images for non-human actions.

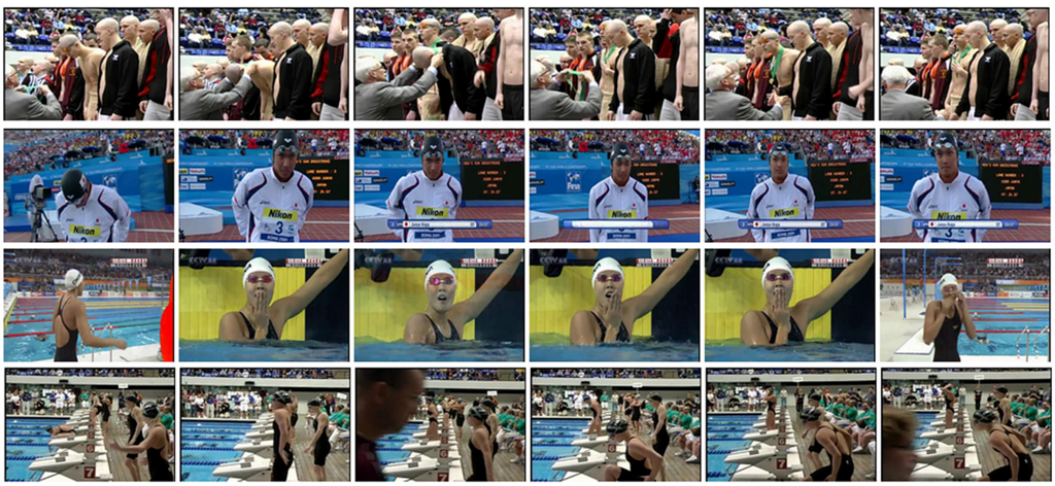

FIGURE 2.17: Top results of "swim backstroke". Since humans could not be detected while swimming in the water so only human detected scenes like medal rewarding, interviewing, result notifying or warming-up (respectively from top to bottom) were obtained. This is one of few cases that human pose comparison based method does not work well.

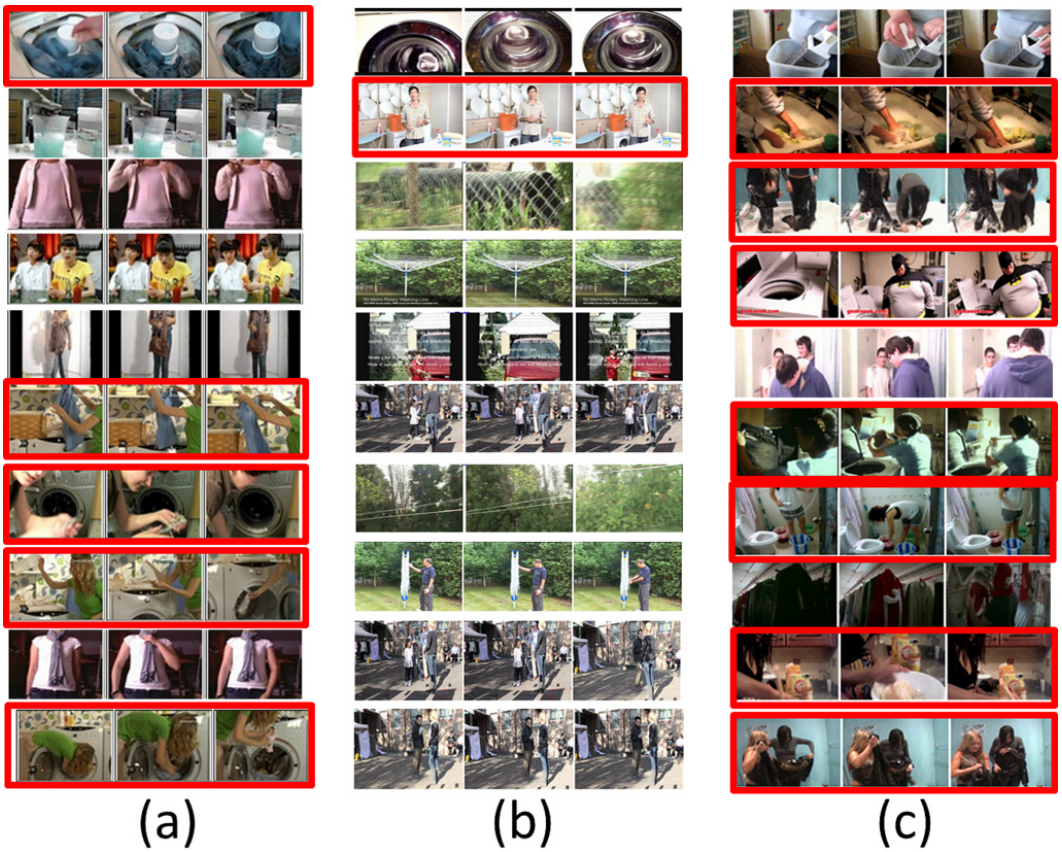

FIGURE 2.18: Top 10 ranked results for "wash clothes" by (a) Web images unexploited framework, (b) local feature matching exploited framework, (c) Pose matching exploited framework. Relevant shots are bounded with red boxes. As shown here, while local feature matching based method ranked less relevant shots to the top, pose comparison based framework biased to the shots which have human poses of washing clothes so it performed better.

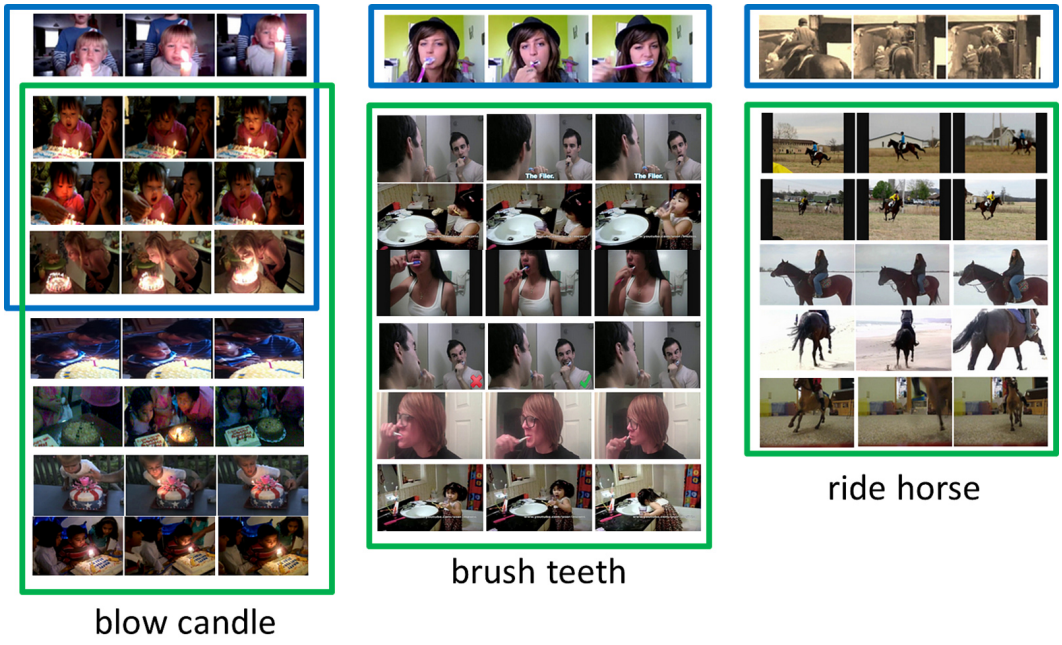

FIGURE 2.19: Relevant shots among top 15 ranked shots for "blow candle", "brush teeth" and "ride horse". Relevant shots which were extracted by Web images unexploited framework and Web image exploited framework with pose matching based shot-image similarity calculation are enclosed by blue and green bounding box respectively. These results demonstrate that exploiting Web images helps to boost more relevant shots to the top.

Chapter 3

VisualTextualRank: An Extension of VisualRank to Large-Scale Video Shot Extraction exploiting Tag Co-occurrence

3.1 Introduction and Related Work

In the previous chapter, we applied VisualRank to the extraction of relevant shots. According to VisualRank, shots found to share the most visual characteristics with the group at large shall be determined as the most relevant ones and brought to the top of search results. With VisualRank, we succeeded in retrieving relevant video shots for many action categories. However, the problem is that, by applying VisualRank, solely visual relationships between shots are explored, thus we obtained at the top many video shots which have similar appearances. This causes the loss of variety in the results. Particularly in some cases, most of those top ranked video shots do not correspond to the given action keywords even though they are visually related.

FIGURE 3.1: An example of Web video retrieval result. This figure shows two video shots together with tag lists of their videos which are retrieved by YouTube with "blow candle" keyword. We can see that some relevant words such as "birthday" and "cake" are tagged to both videos. Thus we can presume that these two video shots are semantically related to each other and relevant to "blow candle" even though they are not visually similar.

Since human actions are too diverse, their corresponding video shots are not always visually similar even if they are semantically related. The change in camera view or the way how people perform the action may cause visual differences. Our intuition is that, two video shots which belong to two videos tagged with related keywords may represent the same action even if they do not hold the same visual features (See Figure 3.1). Hence, shot ranking should additionally consider tag information. Besides, tags are supposed to be more efficiently adopted if their relevance is evaluated considering not only their intra-relationships but also their correlation with video shots. For example, if we find that a video shot is important, or in other words, related to the given action keyword, so that the tags of the video are highly likely to be important as well. And the converse is also true: if a tag was found to be relevant to the keyword, it is highly probable that the videos annotated with it are also relevant.

Then, in this chapter, we present VisualTextualRank as an extension of VisualRank [57], a novel method of co-ranking tags and media data to extract automatically relevant data of given keywords. Our work improves VisualRank by effectively employing both textual information and visual information extracted from the data. We applied our proposed ranking method to our system of extracting automatically relevant video shots of specific actions from Web videos which is represented in the previous chapter. Based on our experimental results, we demonstrate that our ranking method can improve the performance of video shot retrieval over the conventional ranking method VisualRank. Our contribution is a co-analysis of visual links among video shots along with textual link

between videos and their tags and its application to the learning of semantic similarities of video shots. This work of ours is reported in our conference paper [28] and journal article [32].

In this chapter, we performed tag analysis to compute tag-based relevance scores of videos to given keyword as well as their tags. As efforts on tag ranking considering their relevance, Yang et al. [143] proposed a method to evaluate tag relevance score on each tag based on tag co-occurrence statistics. Dong et al. [25] proposed a method to evaluate tag relevance score by combining the probabilistic relevance score estimation and random walk-based refinement. Especially, Liu et al. [74] presented a Web video topic discovery and tracking method via a bipartite graph which represents the correlation between videos and their tags. Actually, their idea is the motivation of this work. However, they tried to find relevant videos of a topic, while our objective is to detect relevant video shots of a keyword. The main difference between their work and our work in terms of methods is that they used only textual information, while we use both textual and visual features. In this chapter, we propose a novel ranking method, VisualTextualRank, which is based on random walks over a bipartite graph to integrate visual information of video shots and tag information of Web videos effectively. We apply our method to our system of automatic extraction of Web video shots of specific actions which is described in the previous chapter. The experiment results demonstrate that using our ranking method instead of VR can obtain more relevant video shots at the top of ranking results.

The remainder of this chapter is organized as follows: In Chapter 3.2, we represent in detail the proposed ranking method which extends [57] and [74]. We then describe our conducted experiments and discuss about their results in Chapter 3.3. Conclusions are presented in Chapter 3.4.

3.2 Proposed Approach

In this chapter, we aim to enhance our system of automatically extracting from tagged Web videos video shots corresponding to specific actions described in the previous chapter by employing our proposed shot ranking method, VisualTextualRank, instead of VisualRank ranking method [57]. The basic ideas of VTR are: the relevant tags are used to annotate relevant videos; the relevant video shots are from videos annotated

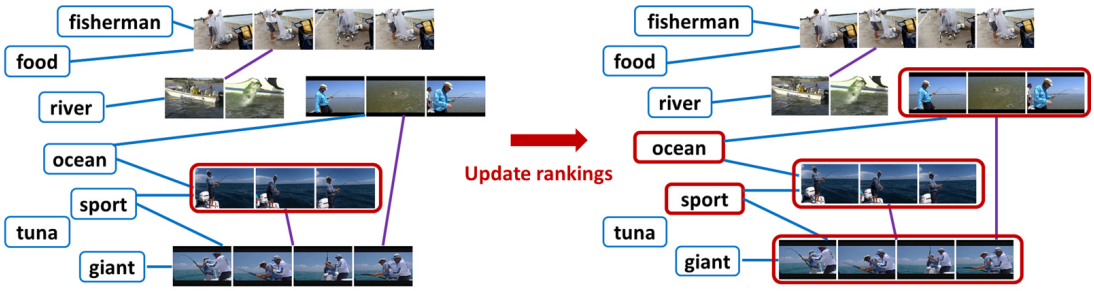

FIGURE 3.2: Illustration of VisualTextualRank by an example of "catch fish" action. Blue links represent relevance between video shots and tags. Purple links refer to visual relationships between shots. Objects marked with red bounding box are considered as being important. Assume that at first we found one important shots as shown at the left of this figure. It will cast its vote for shots and tags which are strongly linked with it. And then at the next step of ranking process, those shots and tags again cast their votes for objects which are tight connected with them. Finally, we can obtain relevant objects of "catch fish" as seen at the left of this figure.

with relevant tags and visually similar to each other. Thus VTR co-ranks tags and video shots so that at each iterative ranking step, ranks of shots are refined using their visual similarities as well as their relevance with corresponding tags, and then, ranks of tags are updated based on their relevance with video shots in conjunction with refined ranking scores of video shots. Figure 3.2 sketches the idea of VTR.

VTR is an extension of VisualRank [57] with ideas motivated by [74]. In [74], tags and videos are also co-ranked using their correlation to refine their relevance with a specific topic. However, unlike our work, in [74], relevance of the whole video, not every scene in it, is evaluated and visual features of videos are totally ignored. On the other hand, VisualRank exploits only a visual linkage between images and does not take textual information into account. In VisualRank, the rank of the image which looks similar to many images with high rank becomes higher after iterative processing. Our proposed VTR employs both visual and textual features of Web videos to explore the mutual reinforcement across video shots and tags.

The proposed co-ranking method can be represented by following iterative processes:

$$\boldsymbol{s}_k = \alpha S_M^* S_C^* \boldsymbol{t}_k + (1-\alpha)\boldsymbol{p} \tag{3.1}$$

$$\boldsymbol{t}_{k+1} = (S_C')^* \boldsymbol{s}_k \tag{3.2}$$

s and t are vectors which represent ranking scores of shots and tags, respectively. The ranks of shots or tags are decided according to the descending order of their ranking scores. Let the number of shots be n_s and the number of tags be n_t, the dimension of S will be $n_s \times 1$ and the dimension of T will be $n_t \times 1$. S_M refers to shot-shot similarity matrix where $S_{M_{i,j}}$ means visual similarity score between shot i and shot j; S_M^* is its column-normalized matrix with size as $n_s \times n_s$. S_C represents shot-tag similarity matrix where $S_{C_{i,k}}$ measures textual relevance score between the video of shot i and tag k; S_C^* is its $n_s \times n_t$ column-normalized matrix. S'_C refers to the transposed matrix of S_C which represents tag-shot similarity matrix and S'^*_C is its column-normalized matrix. Note that since the textual features, here refer to tag co-occurrence, are considered as being noisier than content-based features, we rank video shots first and use their refined ranking scores to update ranks of tags.

t is initially defined as a uniform vector. At each ranking step, after ranking scores of video shots are updated based on their visual similarities and their correlation with tags following Equation 4.4, video shots cast their votes for tags through Equation 3.2. Thus relevant shots will cast important votes for tags which are strongly connected with them. And then at the next iterative step, those tags again help boost ranking scores for video shots which are tight linked with them. Gradually, video shots and tags with few important votes will go to the bottom. VisualTextualRank extends VisualRank by employing tags as textual information and keeps reducing the negative effects of noisy tags by using visual feature based relevance update strategy.

Following VR, we also introduce damping factor α and damping vector p into shot ranking. Damping factor α has been found empirically as holding minor impact on global ordering in ranking results [57, 26]. $\alpha \geq 0.8$ is often chosen for practice. Damping vector p can be a uniform vector or a nonuniform vector. For example, we can use a nonuniform damping vector as described in Chapter 2.3.4 with the idea that shots which are visually related to relevant action images should be biased during ranking computation.

3.3 Experiments and Results

3.3.1 Experiment Settings

We evaluated the effectiveness of our system by precision following the previous chapter. Precision is defined as the percentage of relevant video shots in the top ranked 100 shots (Prec@100). As explained in Chapter 2.4.1, the precision rate at rank 100 should be preferred than "recall" or other evaluation methods such as "average precision" since its objective is to automatically construct action video shot database and the top ranked shots should contain enough positive data for training. The more shots which are representative to the action keyword appear at the top, the better.

We conducted 2 experiments on the human action categories described in the previous chapter. The first experiment aims to compare the performance of our video shot retrieval system proposed in case of applying VisualRank like the original framework and VisualTextualRank proposed in this chapter into the shot ranking step. The second experiment introduces shot-image similarity based shot bias into the shot ranking step while using VisualTextualRank to see if the image information could help further improve our system with VisualTextualRank. The shot-image similarity calculation and shot-image similarity based bias damping vector are calculated respectively following Equation 2.14 (Chapter 2.3.4.4) and Equation 2.7 (Chapter 2.3.2). Since according to the experiment results reported in Chapter 2.4.3, in case of exploiting Web images, human pose obtains better performance, here we employed human pose based similarity.

3.3.2 The efficiency of VisualTextualRank

In the first experiment, we compare the performance of VR and VTR at all 100 human action categories. The precision rates are shown in Table 3.1.

Experiment results demonstrated that by adopting our proposed ranking method instead of the conventional ranking method, more relevant shots were brought to the top. In terms of overall performance, VisualTextualRank improved the average precision over VisualRank by approximately 7%. Figure 3.3 shows some examples of detected relevant shots by applying VisualRank and VisualTextualRank.

TABLE 3.1: Experiment results of all 100 action categories by VR and VTR. VR and VTR refer to performance of video shot retrieval system adopting VisualRank and proposed VisualTextualRank respectively.

Action	VR	VTR
soccer+dribble	100	100
fold+origami	96	99
crochet+hat	95	97
arrange+flower	94	96
paint+picture	88	87
boxing	86	84
jump+parachute	82	63
jump+trampoline	82	92
do+exercise	79	61
do+aerobics	78	79
do+yoga	77	70
surf+wave	75	73
shoot+arrow	73	81
massage+leg	72	78
fix+tire	67	77
batting	66	61
basketball+dribble	64	87
blow-dry+hair	64	59
knit+sweater	64	68
ride+bicycle	62	70
curl+bicep	58	59
shoot+ball	58	58
tie+shoelace	57	73
laugh	50	54
dive+sea	49	41
harvest+rice	49	46
ski	49	60
iron+clothes	47	48
twist+crunch	47	32
dance+flamenco	45	53
dance+hiphop	43	68
eat+ramen	42	47
dance+tango	41	41
play+trumpet	41	59

Action	VR	VTR
play+drum	40	45
skate	37	42
swim+crawl	36	49
cut+hair	35	42
run+marathon	35	43
count+money	33	58
paint+wall	33	32
shoot+football	33	29
draw+eyebrows	32	32
fieldhockey+dribble	32	68
hit+golfball	32	70
lunge	32	27
play+piano	32	34
row+boat	32	23
sing	32	65
chat+friend	31	52
clean+floor	31	38
cut+onion	31	24
shave+mustache	31	30
pick+lock	30	28
plaster+wall	30	55
blow+candle	29	44
wash+face	29	24
walking+street	29	46
brush+teeth	28	27
catch+fish	28	59
drive+car	28	40
plant+flower	28	24
play+guitar	28	41
lift+weight	27	51
raise+leg	27	40
hang+wallpaper	26	46
jump+rope	26	49
climb+tree	24	24

Action	VR	VTR
ride+horse	24	15
roll+makizushi	24	36
sew+button	24	46
fry+tempura	23	12
slap+face	20	45
read+book	19	21
squat	19	34
row+dumbell	16	30
wash+clothes	15	29
wash+dishes	15	39
comb+hair	14	26
drink+coffee	14	16
swim+breaststroke	13	18
cry	12	12
eat+sushi	12	23
serve+teniss	11	27
tie+necktie	11	28
boil+egg	9	11
head+ball	9	16
swim+backstroke	9	9
take+medicine	8	7
serve+volleyball	7	40
swim+butterfly	7	9
bake+bread	6	8
cook+rice	6	11
grill+fish	5	13
jog	5	6
slice+apple	5	16
peel+apple	5	14
bowl+ball	4	4
smile	4	6
kiss	2	3
AVG	**36.6**	**43.5**

Table 3.1 shows that VTR enhanced video shot retrieval system on most of the categories. Particularly, precision was boosted greatly (more than 10%) in many cases such as “fix+tire”, “tie+shoelace”, “ski”, “hit+golfball”, “dance+hiphop”, “plaster+wall”, “blow+candle”, “jump+rope”, “catch+fish”, “swim+crawl”, “play+guitar”, “play+trumpet”, “wash+dishes”, “slap+face”. Only a few categories such as “do+yoga” and “dive+sea” obtained less relevant shots at the top. This can be explained that in such cases, textual information was too noisy so that irrelevant tags and their related shots (which are supposed to be irrelevant as well) were boosted to the top. This problem is very common among approaches which employ Web data. So far Web data has been known

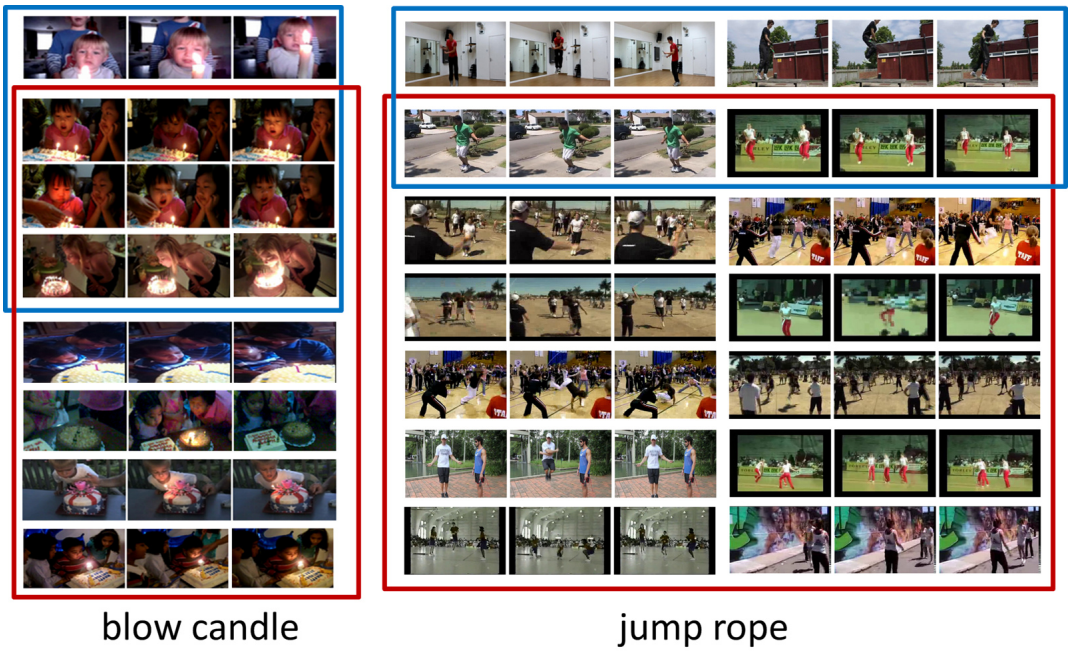

FIGURE 3.3: Relevant shots among top ranked 15 shots. Bounded by blue boxes and red boxes are respectively results obtained by applying VisualRank and our proposed VisualTexutualRank. The results show that our ranking method helps to boost more relevant shot to the top.

as uncontrolled and occasionally extremely noisy data.

Interestingly, we found that, not only precision, VTR also improved VisualRank in terms of variety of ranking results. Since VisualRank employs only visual features, visually similar images are often ranked to the top. In case of shot ranking, applying VisualRank as in the previous chapter tends to boost shots from the same videos to the top since they are generally look similar. On the other hand, VTR additionally exploits the correlation between videos and tags so that not only visually similar video shots but also video shots having strong textual links with relevant shots are highly probable to be ranked high as well. As a result, using VTR can be expected to retrieve shots from more various videos. See Figure 3.4 for an illustrated example.

We define diversity (variety) score of a ranking result as the ratio of the number of identical videos in its top ranked N video shots to N. This definition is based on the fact that the more videos appear at the top, the more diverse the result becomes. The diversity evaluation results are summarized in Table 3.2. Four values of N are taken into consideration: 10, 30, 50, 100. As shown in Table 2, the top ranked shots obtained by adopting VTR are more diverse as they are from more various videos than in case

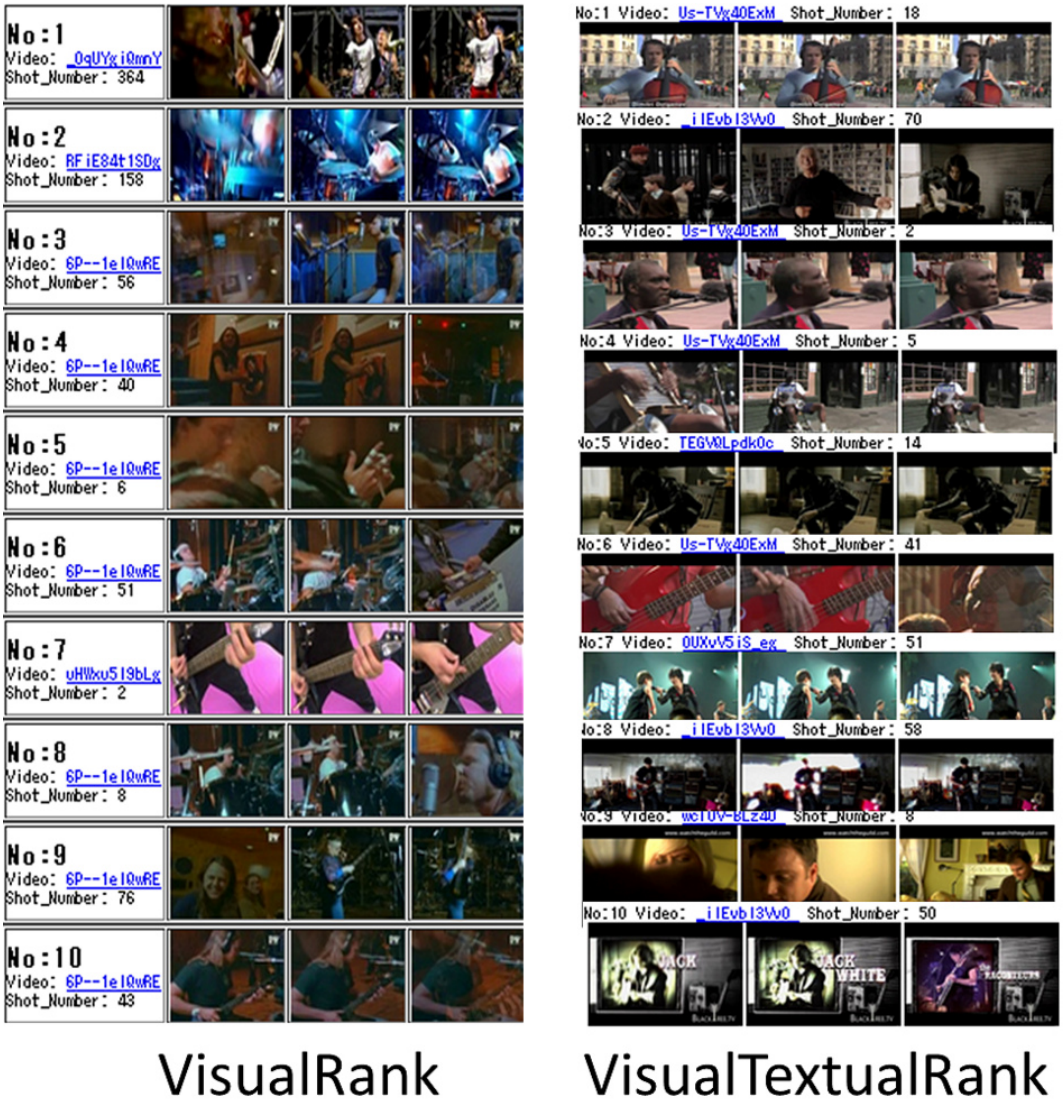

FIGURE 3.4: An example which shows diversity of results obtained by video shot retrieval system with VisualRank (right) and with VisualTextualRank (left). The category here is "play+guitar". In the original framework [26] (Chapter 2.2), more than half of top 10 shots are from the same video with ID "6P–1elQwRE" since they are visually similar. On the other hand, applying VisualTextualRank can select relevant shots from more different videos since it regards not only visually similar shots but also textually related shots.

of using VR. As N increases, the ratios decrease in both cases. However VTR always improves VR in terms of variety by approximately 12-13%.

TABLE 3.2: Evaluation of variety of ranking results. Top10, Top30, Top50 and Top100 here respectively refer to the ratios of the number of videos in top ranked N video shots to N in case of $N = 10$, $N = 30$, $N = 50$ and $N = 100$ (the higher the better). The numbers shown here are the average of results of 100 categories.

Ranking method	Top10	Top30	Top50	Top100
VisualRank	64.2%	46.1%	37.4%	26.8%
VisualTextualRank	**76.1%**	**59.5%**	**50.3%**	**39.1%**

3.3.3 The Performance of The System with VisualTextualRank and Image Exploitation

In this experiment, our framework considers also human pose information during ranking processes and uses damping vector defined in Equation 2.7 (Chapter 2.3.2). Hence this experiment applies our full framework including optional step and VisualTextualRank. Due to computational cost of pose prediction, we chose randomly only 20 categories among 45 failed categories by the original framework (Chapter 2.2) [26] to see how the pose information could help improve our system. As the result of that choosing, the dataset of these two experiments consists of: 7 categories with precision between 20% and 30%, 10 categories with precision between 10% and 20%, and the remainder with precision below 10%.

Here we compare the performance of our framework with 3 different settings for shot ranking step: (1) using VisualRank with uniform damping vector (Equation 2.6, Chapter 2.3.2); (2) using VisualTextualRank with uniform damping vector; (3) using VisualTextualRank with pose similarity based damping vector (Equation 2.7, Chapter 2.3.2). Their performance on tested categories is shown in Table 3.3. As shown in Table 3.3, the framework with VTR and without pose exploitation obtained the best performance on average.

About the effectiveness of introducing human pose feature, we can see that it depends on categories. Experimental results show that this kinds feature helps to improve some categories such as “serve+tennis” or “row+dumbbell” but degrades VTR in some categories such as “blow+candle”, “eat+sushi” and “drive+car”. Our explanation for these results is that in fact, pose features works better when human poses are taken in full body and without large occlusion. This case corresponds to “serve+tennis” and “row+dumbbell”.

TABLE 3.3: Results of 20 human action categories compared between (1) VR (using VisualRank with uniform damping vector); (2) VTR (using VisualTextualRank with uniform damping vector);(3) VTR+pose (using VisualTextualRank with pose similarity based damping vector). All of these categories have precision lower than 30% by the original framework (VR)

Action	VR	VTR	VTR+pose
blow+candle	29	44	35
climb+tree	24	24	24
eat+sushi	12	23	15
jump+rope	26	49	47
catch+fish	28	59	54
read+book	19	21	20
boil+egg	9	11	14
grill+fish	5	13	19
play+guitar	28	41	43
wash+clothes	15	29	31
wash+dishes	15	39	39
drive+car	28	40	34
slap+face	20	45	44
squat	19	34	36
serve+tennis	11	27	30
cook+rice	6	11	15
comb+hair	14	26	27
roll+makizushi	24	36	32
row+dumbbell	16	30	33
tie+necktie	11	28	27
Average	17.9	31.5	30.9

However, in cases like “blow+candle” or “eat+sushi”, in general, only upper bodies appear and they even are obscured by tables (See Figure 3.5). Thus we could not extract pose features properly and employing them in our method led performance of VTR down. This problem of pose features is also discussed in our paper [27].

3.4 Conclusions

In this chapter, we proposed a novel bipartite graph based ranking method, VisualTextualRank, which performs co-ranking of video shots and tags by employing both visual links between video shots along with textual links between videos and their tags. We applied VTR to the system that automatically extracts relevant video shots for specific human actions. The effectiveness of proposed VTR was validated by experiments. VTR could improve the baseline in terms of both precision and variety.

FIGURE 3.5: Effectiveness of introducing pose feature. Top three categories: "serve+tennis" and "row+dumbbell" (whole body seen), "grill+fish" (upper body clearly seen) are examples of ones which Pose-VTR obtained better results. Bottom three categories: "blow+candle", "eat+sushi", "drive+car" (upper body occluded by many objects) are cases when pose cannot be detected and hence pose exploited VTR performed worse than VTR with uniform bias vector.

Chapter 4

Spatio-Temporal Features based on Triangulated Dense SURF Keypoints

4.1 Introduction and Related Work

4.1.1 Introduction

In the previous chapters, to extract spatio-temporal features from videos, we applied the method proposed in [85] which extracts spatio-temporal features based on moving SURF keypoints. If we could extract features from videos more effectively, it would be possible to explore visual characteristics of videos better and obtain more important video shots with our system introduced in Chapter 2. In this chapter, we propose an extension of this method by addressing its problems such as the failured in feature extraction of videos containing camera motion or the holistic decision of motion threshold in the selection of interest points. We propose simple yet effective solutions to solve these problems. That means, similar to [85], our method of feature extraction is also based on SURF interest points with robust movements, nonetheless how we determine those points is different. Moreover, we propose novel features by exploring more aspects of selected points. We use our features to represent videos in our system of extracting Web relevant video shots

of specific actions (Chapter 2) and show that more relevant video shots can be retrieved at the top of ranking results.

In the research field of action recognition, so far low-level spatio-temporal features have been explored to represent videos in many approaches. Some low-level video features are extensions of image descriptors, such as 3D-SIFT [109], HOG3D [63], and Local Trinary Patterns [146]. They are the extensions of well-known and effective image descriptors: SIFT, HOG and Local Binary Patterns, respectively. To determine space-time regions where spatio-temporal features should be extracted, Dollar et al. [33] proposed to apply 2-D Gaussian kernels to the spatial space and 1-D Gabor filters to the temporal direction. They call the obtained regions as cuboids. Laptev et al. [67] proposed an extended Harris detector to extract cuboids. As another method other than using cuboids, extracting local features based on trajectories of interest points also showed good results for action recognition [129, 51, 4]. To track interest points, either tracker based techniques or point matching based techniques have been employed. Matikainen et al. [78] proposed to extract trajectories using a standard KLT tracker, cluster the trajectories, and compute an affine transformation matrix for each cluster center. These approaches have shown very promising results for action recognition. Our method is also based on the trajectories of interest points, nevertheless rather than using the trajectories independently, we encode them according to groups of their points. The groups are formed by employing Delaunay triangulation to the interest points. The idea of applying Delaunay triangulation to cluster interest points was first proposed in Noguchi and Yanai's work [85], which serves as our baseline in this chapter.

In this chapter, we aim to recognize actions in realistic videos with complex background. Empirical results have shown that dense features achieve better performance than non-dense ones, particularly in case of complex videos [129, 52, 86, 138]. Especially, in [129], Wang et al. proposed to extract dense trajectories and compute descriptors such as HOG, HOF and MBH within space-time volumes aligned with the trajectories. Their approach has become one of the state-of-the-art as it outperforms many other approaches in action recognition field. The illustration of their features is shown in Figure 4.1.

Since the success of dense trajectory based features, only a few new local STFs have been investigated [64]. Some recent work on low-level features are only extensions of this method [130, 81, 55, 51, 111]. In this chapter, we propose novel low-level STFs

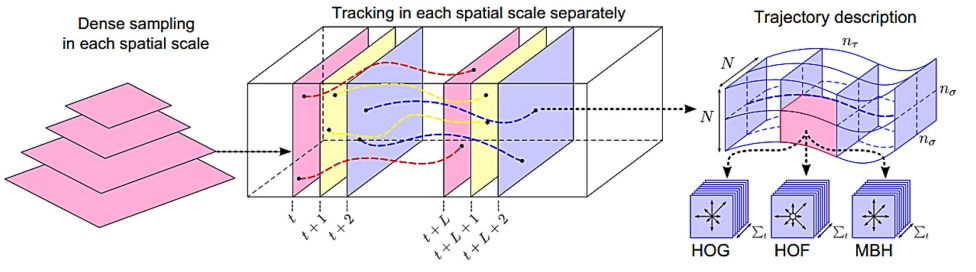

FIGURE 4.1: Overview of STFs extraction method proposed by Wang et al. (cited from [129]). Left: Feature points are sampled densely for multiple spatial scales. Middle: Tracking is performed in the corresponding spatial scale over L frames. Right: Trajectory descriptors are based on its shape represented by relative point coordinates as well as appearance and motion information over a local neighborhood of $N \times N$ pixels along the trajectory.

which are extracted based on triangulation of dense SURF keypoints with dominant and reliable movements. In this case, dominant and reliable points refer to informative points which are supposed to be representative for the actions. Our spatio-temporal features investigate triangles which are produced by applying Delaunay triangulation to those informative points. Shape features of the triangles along with visual features and motion features of their points are taken into account to form our features. We show that concatenating these features with SURF features of interest points can form a powerful representation for videos. Our experiment results conducted on several action recognition benchmarks show that our features are comparable to the state-of-the-art. This work of ours is reported in our conference paper [30].

In this chapter, we use Fisher Vector (FV) to represent videos following many recent work on action classification. FV encoding technique was first applied to image classification task several years ago, shown to extend the traditional BoV representation [91]. The advantage of this technique has been demonstrated that it is not limited to the number of occurrences of each visual word but it also encodes additional information about the distribution of the descriptors. We applied FV model following Sanchez et al. [104]. The methodology of this model is summarized as follows: (1) extracting local features from images, (2) modelling the distribution of those features as mixtures of Gaussian(GMM), training a soft codebook, (3) applying Fisher kernels on the obtained codebook to encode each image as a Fisher Vector.

In our experiments, we apply our method to extract features from videos collected by our system which is introduced in Chapter 2. The experiment results show that

we are able to obtain many more representative shots among the top ranked shots with our method in comparison to the baseline method. Furthermore, to validate the effectiveness of our proposed method of feature extraction on action recognition task, we conduct our experiments on two data sets: UCF50 [97] and UCF101 [61]. These data sets have been known as tough and large scale benchmarks for human action recognition with challenging settings such as large variations in camera motion, object appearance and pose, viewpoint and complicated background and so on. The experiment results demonstrate the efficiency of our improvements over the baseline on this task as well.

The reminder of this chapter is organized as follows: Chapter 4.2 describes the proposed ST feature extraction method in detail. Chapter 4.3 explains about conducted experiments and presents the results. Conclusions are presented in Chapter 4.4.

4.1.2 Related work

One of the most related work to ours is Chen et al.'s work [18]. In [18], Chen et al. proposed a feature which is called as MoSIFT with a quite similar idea to our baseline method. They proposed to apply SIFT algorithm to find candidate points in the spatial domain at first and then detect spatio-temporal interest points with motion constraints. Their descriptor is constructed by concatenating HOG (which describes the spatial appearance of the points) and HOF (which represents the movements of the points). On the other hand, we extract features from interest points not individually but according to their groups. Moreover, we do not design our descriptor by simply combining conventional spatial features and motion features from the detected points but exploring novel aspects of them in both spatial and temporal domains at the same time. Therefore, even though the basic idea of how to detect interest points is common between us and Chen et al., how we implement the idea and how we extract features from detected points are different.

Beside low-level features, mid-level features and high-level features have also been explored in some recent works [50, 150, 133, 44, 42, 103]. In [50], Jain et al. proposed to represent videos by discriminative spatio-temporal patches which are automatically mined from training videos. Their spatio-temporal patches are supposed to correspond to a primitive human action, a semantic object related to the action (such as "weights" in "clean and jerk" action), or a random but informative spatio-temporal patch in the

video. In [150], Zhu et al. proposed a two-layer structure for action recognition to automatically exploit a mid-level action representation which they called as "actons". More specifically, in their method, the first layer builds a low-level representation using classical BoF-STP model, while the second layer automatically exploits actons which are built directly on top of the first layer via their weakly-supervised learning strategy. Even though their proposed representation outperformed many low-level representations, the performance varies on the number of actons per class, and actons must be learned in a supervised manner, thus it is not easy to implement their method compared to low-level approaches.

High-level features such as human-object interactions have also been investigated to represent videos. In [41, 42], Filipovych et al. modeled human-object interactions based on the trajectories and appearance of spatio-temporal interest points. Their approach was applied only to controlled videos taken from the viewpoint of the actor by a static camera against a uniform background. In [44], Gupta et al. employed hand trajectories to model the objects and the human-object motions for classifying interactions between humans and objects. In their work, the motion can be simply extracted based on background subtraction since they worked only on videos with constrained environment (static and fixed background). The works that rely on human-object interactions must encounter many problems related to the diversity of camera viewpoint, object appearance and so on in uncontrolled videos. As a high-level representation which can tackle the problems in uncontrolled videos, Action Bank [103], which is inspired by Object Bank, is a powerful representation of actions. Action Bank represents a video as the collected output of many action detectors that each produces a correlation volume. Each element of action bank is a template-based action detector which captures an individual example action, such as "running-left" or "biking-away". Though Action Bank achieved promising results, its usage of Action Bank is still limited since first, it is not so computationally feasible for large-scale action recognition and second, its performance is not outstanding, in comparison to state-of-the-art low-level features.

Here we refer to some action recognition approaches which consider motion compensation while extracting features. Up to this point, not many approaches like that have been proposed in the literature. Our baseline method [85] does not compensate camera motion but simply ignores the information of the video during the time that camera motion is detected. Thus, this method must suffer from huge information loss and is not

able to extract any feature in case that the whole video contains camera motion. In [22], Cinbis et al. applied video stabilization using homography-based motion compensation approach. They estimated camera motion by calculating the homography between consecutive frames and compensate optical flow of points by removing the estimated camera flow. Similarly, Jain et al. [51] also removed camera motion from the original optical flow, nevertheless they consider affine motion as camera motion. Uemura et al. [123] proposed to combine feature matching with image segmentation to estimate the dominant camera motion, and then separate feature tracks from the background. Wu et al. [110] decomposed Lagrangian particle trajectories into camera-induced and object-induced components for videos acquired by a moving camera. In. [129], Wang et al. did not compensate camera motion in advance but employed motion boundary histograms which already have constant motion removed. uires extra computational cost.

In this chapter, we propose a simple but efficient method of compensating camera motion without making use of additional methods such as image segmentation as in [123] or human detection as in [130]. Our proposed method improves significantly performance of feature extraction over the baseline [85] since it helps not only extract features in case that camera motion exists but also detect more robust interest points.

A standard approach to describe an image for the purpose of classification is to extract from it a set of descriptors, encoding them into a high dimensional vector and pooling them in to an image-level signature. According to the success of the BoV (Bag of Visual words) on image classification, it has also become the most popular model for video representation. Nevertheless, the BoV model suffers from some limitations, one of which is the loss of some discriminative information in both spatial and temporal dimensions. There have been several extensions of this popular model, including the use of better coding techniques based on soft assignment [37, 139, 92, 126] or sparse coding [142, 9, 132] and the use of spatial pyramids to take into account some aspects of the spatial layout of the image [69]. As one of other effective models of human action recognition, a dense representation proposed by Zhen et al.[148] takes into account the motion and structure information simultaneously. In their work, high dimensional features are first extracted and then embedded into a compact and discriminative representation by DLA (Discriminative Locality Alignment) method. On the other hand, instead of using all frames in the video sequence, Liu et al.[73] proposed to learn the

most representative frames called as key frames by AdaBoost algorithm and represent action by the probabilistic distribution and temporal relationships of these frames.

4.2 Proposed Method of Extracting Spatio-Temporal Features

4.2.1 Overview of Proposed Method

In this chapter, we propose to improve the method of extracting ST features [85]. Following [85], we also extract features based on moving SURF points and use Delaunay triangulation to model the spatial relationships between interest points. We address some problems of the baseline method such as the inability to handle camera motion or holistic decision of motion thresholds for selecting points which cause failure in extracting features of some videos. We propose to solve these problems by our simple yet efficient methods of motion compensation and point selection. The overview of proposed method which extracts spatio-temporal features is illustrated in Figure 4.2.

We extract features with temporal step size of N frames. With a set of N consecutive frames, we extract dense SURF keypoints and track them through N frames. In our experiments, we fix N as 5 following the baseline. As trajectories may drift from their precise locations during the tracking process, limiting the tracking process within short duration like this is supposed to be able to overcome this problem. In the case that the number of frames in the given video is not a multiple of N, we simply ignore the last remaining frames. The process of extracting our proposed spatio-temporal feature from a frame set is summarized as follows:

1. Extract dense SURF keypoints of the first frame using Dense SURF [124].
2. Compute optical flows from k^{th} frame ($k = 1, 2, ..., N-1$) to the next frame ($k+1^{th}$ frame) using Large Displacement Optical Flow (LDOF) [11].
3. Estimate camera motion in each frame and compensate motion if camera motion detected (Chapter 4.2.2).
4. Select points which are expected being more informative than the others (Chapter 4.2.3) and form triangles of selected points using Delaunay triangulation.

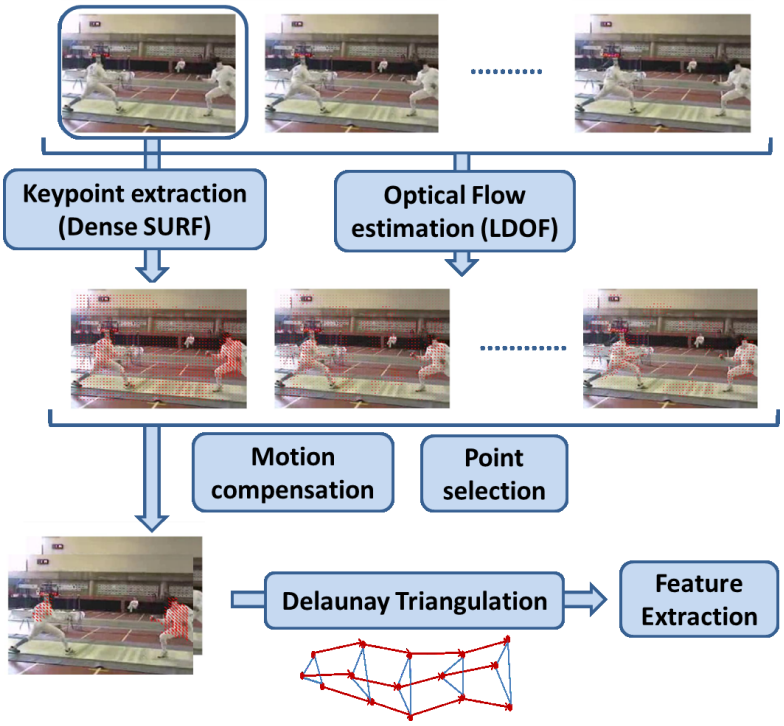

FIGURE 4.2: Overview of proposed method. The example shown here is a part of "Fencing" video shot taken from UCF101 data set. Given a frame set, we first extract dense SURF keypoints from its first frame. We then estimate optical flows of extracted keypoints between each frame and the next one by using LDOF method. The red dots and lines shown in the figures in the second rows respectively represent SURF keypoints and their estimated flows. We then apply our proposed methods of motion compensation and point selection which are based on optical flow information to obtain most informative points as shown in the bottom left of this figure. We use Delaunay Triangulation to group these points and extract our proposed spatio-temporal features from each group.

5. Extract ST features from each triangle based on its shape along with motion features of its points through the frame set (Chapter 4.2.4).

The main improvements of our method over our baseline can be summarized as follows: (1) treatment of camera motion, (2) selection of interest points and, (3) enhancements on descriptorization of ST features. We explain in details these improvements in following subsections.

4.2.2 Detection and Compensation of Camera Motion

According to the baseline method [85], once camera motion has been detected in a frame set, obtained information would be considered as noise, thus no points would be selected. Consequently, no features are extracted if the whole video contains camera

motion. We propose a simple technique to solve this problem. Our proposed method can improve the performance over the baseline since first, our method is able to extract features from videos which contain camera motion and second, our method employs compensated motion which provide more accurate information.

We propose a simple 2-step technique to detect and compensate camera motion. The technique can be summarized as follows:

1. Step 1: Confirm the existence of camera motion based on optical flows of SURF keypoints. If detecting camera motion, determine the direction and magnitude of camera motion before going to the next step.
2. Step 2: Compensate motion by cancelling camera motion from original flows of SURF keypoints.

Detection of camera motion: At the first step, we aim to find out how camera moves at each frame. The camera can stay still (no camera motion) or move in horizontal direction (right or left) and/or vertical direction (up or down). This step is based on our assumption that if most points move toward the same direction, camera moves in that direction. Let denote P^{x^+} and P^{x^-} as number of points with positive and negative optical flows respectively, $P_m^{x^+}$ and $P_m^{x^-}$ as number of moving points which shift to the right and the left respectively, so that we suppose that camera is moving right if Equation 4.1 and Equation 4.2 are satisfied or moving left if Equation 4.3 and Equation 4.4 are satisfied:

$$P_m^{x^+} \geq kP^{x^+} \tag{4.1}$$

$$P_m^{x^+} > P_m^{x^-} \tag{4.2}$$

$$P_m^{x^-} \geq kP^{x^-} \tag{4.3}$$

$$P_m^{x^-} > P_m^{x^+} \tag{4.4}$$

Here, k is a fraction threshold representing minimal required proportion of moving points over all points with the same direction. In our experiments, we set k as $\frac{2}{3}$. A point is considered as a moving point if its absolute optical flow is larger than or equal to 1.The camera is supposed as horizontally stable if none of above condition is satisfied. If the

FIGURE 4.3: An example that shows efficiency of proposed method of reducing camera motion and selecting interest points. The first row presents a frame set of consecutive frames which contains camera motion. In this case, camera is moving (to the right), thus interest points are not detected according to the baseline. The most left figure of the second row shows optical flows of extracted dense SURF keypoints before the camera motion compensated. The middle figure and the right figure of the second row respectively present points determined as moving points by the baseline (with fixed motion threshold) and our method (with flexible threshold). Point selection of the both are performed after compensating camera motion. With our flexible motion threshold, only the most informative and representative points (which belong to the actor) are selected, while with fixed motion threshold, background points are also selected. This example shows that our method is not only able to reduce the effect of camera motion but also to select more representative interest points than the baseline.

camera is detected as being moved, camera motion is calculated as average of absolute optical flows of points which moved to the same direction as camera. Camera motion for vertical direction is estimated in the similar manner.

Compensation of camera motion: If camera motion is detected, low of each SURF keypoint is compensated simply as follows:

$$f_i = f_i - df_{camera} \tag{4.5}$$

Here, f_i refers to flow of point i, f_{camera} refers to camera flow. d equals 1 if camera moved to positive direction or -1 if camera moved to negative direction. f_{camera} is measured separately for all considered directions (forward, backward, up and down) and compensation is operated in each of those directions. By our manner, camera motion can be compensated in most cases except when the camera moves forward/backward. Handling this case of camera motion is one of our future works. See Figure 4.3 for an example result of our motion compensation method.

FIGURE 4.4: Some examples to show that motion threshold should be flexible. In general, sport activities such as surfing, high jump and ice dancing generate larger movement displacements than daily activities such as apply lipstick, typing and shaving beard. Even with the same kind of action, as shown in the right, the further the distance between the camera and the actor, the smaller the movements may look.

4.2.3 Selection of Interest Points

Remember that in our method, not all of the extracted SURF keypoints but only a portion of them are considered as interest points. According to the baseline, selection of interest points is based on their optical flows between the first frame and the middle frame of the frame set. A point is believed as an interest point if its flow is larger than the pre-defined motion threshold. As a result, in case that no points satisfy that condition, no feature can be extracted. Moreover, in the baseline, the motion threshold is determined in a holistic manner and fixed for every frame of every video of every action. However, points which are selected based on a constant motion threshold may not always be representative. Magnitude of movement may vary largely from action to action. For instance, sport activities such as jumping trampoline or swimming are supposed to cause large displacements. On the other hand, daily activities such as drinking or talking in general generate smaller optical flows. Even with the same kind of action, changes in environmental conditions such as distance between camera and actor also can cause distinct movement quantity (see Figure 4.4 for the illustration). We demonstrate that in order to overcome these problems, motion threshold should be flexible.

We propose to determine motion threshold flexibly and select as many reliable moving points as possible. The idea is that the robustness of a point should be compared to its surrounding points at the same time rather than to a fixed threshold. In our method, motion threshold is estimated for every frame in all directions based on flows of its SURF points. The following equation represents how we calculate motion threshold for a frame

in forward direction (x^+). Thresholds for the remaining directions are calculated in the similar way.

$$t_{f_{x^+}} = a_{f_{x^+}} + \alpha(m_{f_{x^+}} - a_{f_{x^+}}) \tag{4.6}$$

Here, $t_{f_{x^+}}$ means the motion threshold for frame f in x^+ direction. $a_{f_{x^+}}$ and $m_{f_{x^+}}$ respectively refer to the average and the maximal flow magnitude at frame f in x^+ direction. The qualification that a point should satisfy to be considered as a moving point is that in at least one of four considered directions, its flow magnitude is somewhat greater than the average flow of that direction. The constant α controls that qualification. In our experiments, we set α as 0.5. Thus, the motion threshold is near to the median of the average and the max flows. However, in some cases, at some frames, all objects including actor stay still, thus it is not necessary that there always must be moving points. We suppose that nothing in a frame moved if all of its thresholds are smaller than 1.

After determining which points are moving points through the frame set, instead of simply taking all points which ever moved like in [85], we select only representative points. We postulate a hypothesis that points with more movements are more reliable and informative. For example, through the whole frame set, points moved 2 times are expected to be more reliable as well as representative than points moved only once. Based on this hypothesis, we propose to select points greedly based on number of times they moved through the frame set. Our algorithm of point selection is described in Algorithm 1. Following Algorithm 1, the group of selected points is only a proportion of moving points but expected to consist of most representative points. In our experiments, we set β as $\frac{1}{2}$. Figure 4.3 shows the effectiveness of our method of selecting interest points over the baseline.

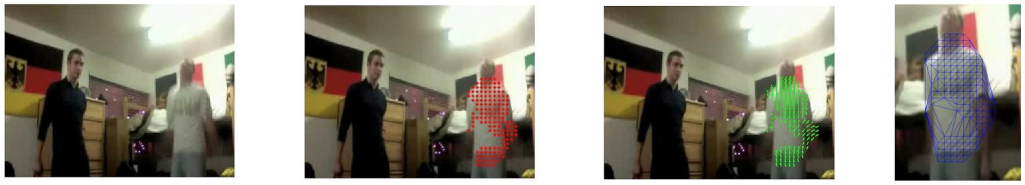

FIGURE 4.5: An example of Delaunay triangulation applied on a frame of "jumping jack" scene. From the left, the first figure shows the original frame. After point selection step, interest points (red star dots) are detected as shown in the second figure. Their optical flows are drawn as green lines in the third figure. The points are then clustered to triangles by using Delaunay triangulation as shown in the most right figure. In comparison with the baseline [85], due to dense sampling, we could obtain many more keypoints. (Please refer to Figure 2.8 for the example of Delaunay triangulation applied on conventional SURF keypoints)

Algorithm 1. Algorithm for selecting interest points

M = maximal number of movements ($M \leq N - 1$)

T = total number of moving points

GS = group of selected points (initialized as empty)

for $i = M$ to 1 **do**

  $GS = |GS$, points moved i times $|$

  if $|GS| \geq \beta T$ **then**

    break;

  end if

end for

end

4.2.4 Descriptorization of Spatio-Temporal Features

After selecting interest points, following the baseline, we apply Delaunay triangulation to cluster them to triples and extract spatio-temporal features from each triple. An example of Delaunay trianglulation result is shown in Figure 4.5. Our proposed ST is constructed based on following descriptors. We classify them to *spatial descriptors* which represent static visual features of points, *temporal descriptors* which present movements of points through the frame set and *spatio-temporal descriptors* which characterize trajectory-based visual features of points or group of points. Below we describe in detail each descriptor.

Spatial Descriptors. Spatial descriptors here refer to SURF descriptors of three points of a triple at the first frame. Following [124], SURF features are extracted with subregions of 3 by 3 pixels, Haar filters of 4 by 4 pixels. Thus we obtain 64-dimension SURF descriptor for each point.

Note that instead of combining three SURF descriptors of three points to form a single spatial descriptor then applying PCA on this descriptor like in our conference paper [30], here we concatenate SURF descriptors with temporal descriptors and spatio-temporal descriptors which we describe below then apply PCA on the whole concatenated vector. We found that this method of feature formation obtains better performance. The reason is that our new version of feature formation can advoid more redundant information than the previous one.

Temporal Descriptors. We extract following 2 temporal features: the first one is originally proposed in our baseline [85] and the second one is inspired by [17].

(1) **A Histogram of Direction of Flows (HDF)**. We generate a 5-dim vector for each interval of each moving interest point using their optical flows. The 5-dim vector consists of x^+, x^-, y^+, y^- and no optical flow x^0. Here x^+ and x^- respectively mean the degree of the positive elements and negative elements along x-axis (similar denotation manner for y-axis). The motion feature for each interval is normalized so that the summation of all the elements equals to 1. All of the 5-dim vectors extracted from $N-1$ intervals are concatenated into one motion vector for each moving point, and totally the dimension of motion feature becomes $(N-1) \times 5$.

(2) **A histogram of Optical Flow (HOOF)**. $3(N-1)$ flow vectors of 3 points are binned to B_o-bin histogram. Following [17], each flow vector is binned according to its primary angle from the horizontal axis and weighted according to its magnitude. That means, a flow vector $v = [x, y]$ with its angle $\theta = tan^{-1}(\frac{y}{x})$ in the range shown in Equation4.7 will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin b.

$$-\frac{\pi}{2} + \pi\frac{b-1}{B_o} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B_o} \tag{4.7}$$

Finally, the histogram is normalized to sum up to 1.

Spatio-temporal Descriptors. We propose to generate the following 3 descriptors. The first two represent visual characteristics of triangles through the frame set. The

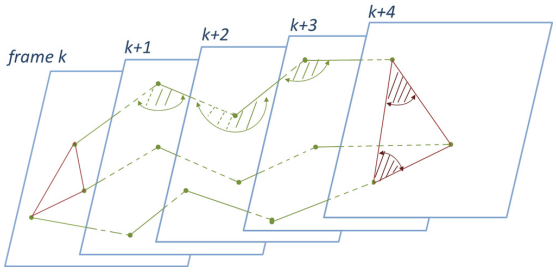

FIGURE 4.6: Illustration of proposed spatio-temporal features. We additionally explore characteristics of interest points by emloying angles of triangles formed by them (red ones) and angles shaped by consecutive trajectories of them (green ones). We show here an example of trajectories of grouped interest points in a frame set of 5 frames. 2×5 smallest angles of triangles are binned to obtain a HAT and 3×3 trajectory based angles are binned to obtain a HAF following proposed method described in Chapter 4.2.4.

last one descriptorizes the shape of trajectories. The last two are newly introduced by us in [30]. Refer to Figure 4.6 for illustration of these proposed two features.

(1) **Areas of Triangle (AT)**: Following the baseline, the areas of the triangle at all frames are calculated then concatenated and normalized to form a N-dimension descriptor.

(2) **A Histogram of Angles of Triangle (HAT)**. To better explore the shape characteristics of the obtained triangles, we propose to investigate their angles by binning them based on their magnitude. Here, we consider only two angles since given the degrees of any two out of three angles, it is sufficient to characterize the shape of a triangle. Using two optional angles is not preferred here since they may be not representative for their triangle. Thus, one can consider using two largest or two smallest angles. However, two largest angles can range from 0^o to 180^o while two smallest angles range only from 0^o to 90^o. Hence, we select two smallest angles since binning them is expected to be more efficient and easier to define histogram bin. Moreover, it cannot happen that both of two smallest angles are larger than 60^o. Based on this observation, we set up histogram bin as follows: for $\theta > 60^o$, the histogram bin is of size 30, otherwise, the histogram bin is of size 15. In this manner, $2 \times N$ smallest angles are binned to 5 bins: [0-15], [15-30], [30-45], [45-60], [60-90]. Each angle is weighted by sum of magnitude of its two edges and normalized at the end.

(3) **A Histogram of Angles of Flows (HAF)**. To exploit trajectories of interest points for modelling the action, some work straightly employ them as descriptors [129]. However, this approach suffers from the problem that trajectories may vary largely due

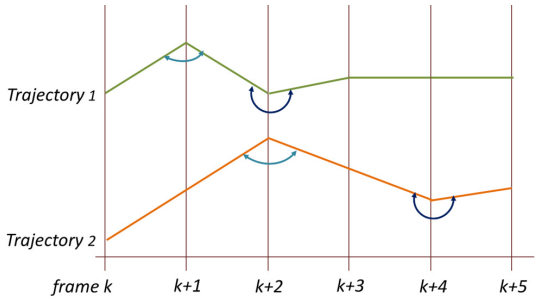

FIGURE 4.7: An example that illustrates the effect of variety in velocity on action recognition and the efficiency of our proposed method. We show trajectories of points which belong to two actors performing the same action in 6 consecutive frames. We assume that the actors move in similar way but at different speed. As shown here, Trajectory 1 which corresponds to faster actor and Trajectory 2 which belongs to lower actor only match at first, thus trajectory based descriptors become nearly totally different. On the other hand, according to our method, exploiting angles shaped by trajectories help to find out more the similarity between these two trajectories. The similar angles (marked by same color) can be binned to the same bin, hence this angle based descriptor can be expected to reduce the effect of diversity in velocity.

to the velocity of the actor. To reduce the effect of variety of velocity, we propose to extract features based on angles shaped by trajectories. These angles are supposed to be more informative than trajectories themselves (See Figure 4.7). The angles are binned by the same method as shown in Equation 4.7. Number of histogram bin for HAF is denoted as B_a.

After calculating all above descriptors, we first concatenate them to form our spatio-temporal feature and then apply Principal Component Analysis (PCA) to reduce the feature dimensionality by a factor of two following [130]. Before PCA process, our feature dimensionality is $(64 + (N-1) \times 5) \times 3$ (SURF and HDF of 3 points) + B_o (HOOF) + N (AT) + 5 (HAT) + B_a (HAF). In our experiments, we set $N = 5$, $B_o = 6$ and $B_a = 4$, thus our proposed descriptor is a 272-dimension vector and after PCA process it becomes a 136-dimension vector.

4.3 Experiments and Results

We first validate the effectiveness of our proposed method on action recognition task with two large-scale and well known datasets: UCF50 and UCF101. The experiment results show that our features are comparable and complementary to most state-of-the-art features. Furthermore, we apply our method to extract features from videos collected

by our system of action shot extraction which is introduced in Chapter 2. According to the results of this experiment, we are able to obtain many more representative shots among the top ranked shots with our method in comparison to the baseline method.

4.3.1 Experiments on Action Recognition

4.3.1.1 Databases, Evaluation Methods and Experimental Setups

In this section, we conduct our experiments on UCF50 and UCF101 data sets to validate the effectiveness of our proposed method and compare our method to the baseline as well as recent action recognition approaches. UCF50 and UCF101 are action recognition data sets of realistic action videos which are collected from YouTube. Both of them can be downloaded from their sites [1]. These data sets are very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions and so on. UCF50 data set is a data set with 6681 videos representing 50 action categories. UCF101 data set contains 13320 videos from 101 action categories. UCF101 data set is an extension of UCF50 data set. Videos for each action in these data sets are grouped into 25 groups, where each group can consist of 4-7 videos of the action. The videos from the same group may share some common features, such as similar background, similar viewpoint. Figure 4.8 shows thumbnails of action categories in the data sets.

As evaluation criteria for UCF50, we follow the method suggested by the authors of the data set [96], that is "Leave One Group Out Cross Validation" which will lead to 25 cross-validations. The videos belonging to the same group are kept being separated in training and testing, since the videos in a group are obtained from single long video, sharing videos from same group in training and testing sets would give high performance. For UCF101, we followed the evaluation set up as suggested in THUMOS Challenge (ICCV'13 Workshop on Action Recognition with a Large Number of Classes) [2]. This challenge aims at exploring new challenges and approaches for large-scale action recognition with large number of classes from open source videos. We adopt the provided three standard train/test splits to evaluate our results. In each split, clips from 7 of the 25 groups are used as test samples, and the rest for training. The result of each

[1] http://crcv.ucf.edu/data/
[2] http://crcv.ucf.edu/ICCV13-Action-Workshop/

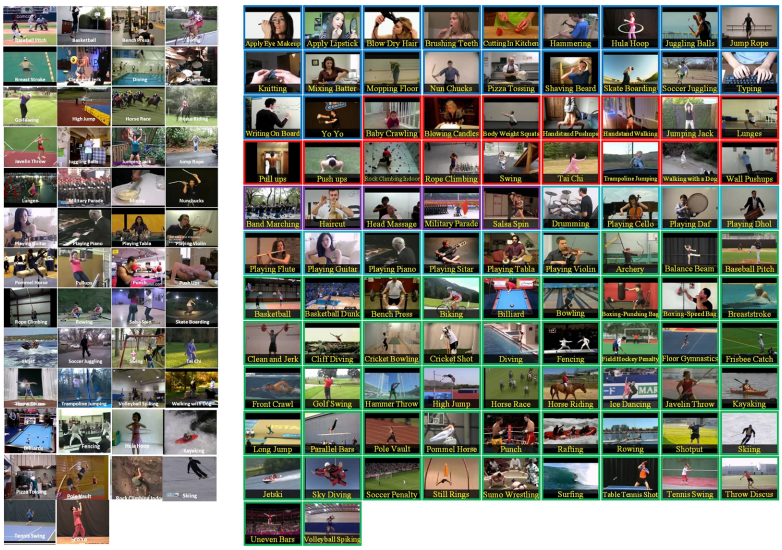

FIGURE 4.8: Thumbnails of UCF50 (left) and UCF101 (right) (quoted from the data set site: http://crcv.ucf.edu/data/). UCF50 data set has 50 action categories collected from Youtube. UCF101 data set is the extension of UCF50 and consists of various action categories including sport activities such as "Basketball Shooting" or "Biking" and daily activities such as "Blow Dry Hair" or "Brush Teeth". The action categories in UCF101 are divided into five types which are shown in bounding boxes with different colors: 1)Human-Object Interaction (blue) 2) Body-Motion Only (red) 3) Human-Human Interaction (purple) 4) Playing Musical Instruments (cyan) 5) Sports (green).

experiment reported here is calculated as the mean of average accuracies over all of the test splits.

In each experiment in the followings, we use Fisher Vector encoding for every kind of experimented features to represent the videos. Following [130], for each feature type, we first reduce the descriptor dimensionality by a factor of two using Principal Component Analysis (PCA). We set the number of Gaussians to K = 256 and randomly sample a subset of 256,000 features from the training set to estimate the GMM. We use VLFeat library [3] with the default settings to perform clustering and encoding. In case of feature fusion, early fusion is applied. That means, fisher vectors of all features are first conducted separately, then they are concatenated to form a single vector before training stage. We use LIBLINEAR library [36] to perform training and classification. By using LIBLINEAR instead of LIBSVM [16] in case of large data, computational time has been shown to be significantly reduced while similar classification rates can be achieved. The library is available online [4]. As for parameter C, we try $C = 0.1, 1, 10, 100$ (default value is 1) and report the best result for each experiment.

[3] http://www.vlfeat.org/
[4] http://www.csie.ntu.edu.tw/c̃jlin/liblinear

TABLE 4.1: Summarization of results by proposed methods. Their methodologies are first explained. "o" means "adopting", blank means "not adopting". For example, P0 can be interpreted as "adopting Dense SURF" and "not adopting point selection, motion compensation and proposed features". Then experiment results on UCF50 and UCF101 are shown. AP means Average Precision.

Method	Dense SURF	Point Selection	Motion Compensation	Proposed Features	AP on UCF50	AP on UCF101
BL [85]					56.3%	41.3%
P0	o				66.2%	50.1%
P0+PS	o	o			78.8%	67.6%
P0+PS+MC	o	o	o		80.4%	69.7%
Proposed	o	o	o	o	**83.6%**	**74.6%**

4.3.1.2 Improvements of Proposed Method Over The Baseline

To validate the improvements of proposed method over the baseline, we conduct experiments to show the efficiency of our improvements in each step: keypoint extraction, interest point selection, motion compensation and feature extraction. In the tables which are shown below, BL refers to the baseline method [85]. P0 refers to the baseline with more sophisticated method for the stage of extracting keypoints. That means, instead of applying the traditional method of SURF keypoint detection like BL, P0 uses Dense SURF [124]. According to Dense SURF, SURF keypoints are extracted using dense sampling on a regular grid. With dense sampling, we can obtain many more keypoints and reduce the risk of information loss. We proposed to improve the interest point selection step by employing flexible motion thresholds (Chapter 4.2.3). We call P0 with our proposed point selection method as P0+PS. We further enhanced our feature extraction method by introducing motion compensation (Chapter 4.2.2). Our method of feature extraction including motion compensation and point selection is denoted as P0+MC+PS. All of these methods (BL, P0, P0+PS, P0+MC+PS) use only features proposed in the baseline [85]. The full model of our method, P0+MC+PS which employs our spatio-temporal features (Chapter 4.2.4), is called as Proposed. The summarization of all methods (including their denotations and experiment results) is shown in Table 4.1.

According to the experiment results, the recognition accuracies are significantly boosted for both data sets by using our proposed methods. As shown in Table 4.1, P0 which employs Dense SURF could obtain more features thus achieved better performance than BL which uses the original SURF. By selecting interest points with flexible motion thresholds as proposed in Chapter 4.2.3 instead of fixed thresholds as in [85], we could obtain

huge improvement (approximately 12% and 17% for UCF50 and UCF101, respectively). This only can be explained that we could select more representative points and reject more redundant ones. Moreover, our proposed method of motion compensation could help accuracies increase approximately by 2%. In comparison to the baseline, in case that camera motion existed, adopting motion compensation could not only make feature extraction be possible but also acquire more precise motion information. Finally, by extracting our proposed features in addition to the baseline features, classification performance was further enhanced (approximately 3% for both data sets). These results demonstrated that, our method could select more discriminative points as well as explore better the characteristics of the points compared to the baseline. We obtained significant improvements on classification accuracies over the baseline. Even though accuracies achieved on UCF101 are always lower than those on UCF50 due to its larger scale and more challenging experiment settings, the improvements on both data sets are consistent.

4.3.1.3 Comparisons to Recent Approaches

Here we compare our proposed method (refered as Proposed above) with recent approaches (the state-of-the-art in the recent two years). Moreover, we combine our features with features of the most successful features, HOG, HOF and MBH extracted within space-time volumes aligned with improved dense trajectories [130]. We apply early fusion to combine features as described in Experimental Setups (Chapter 4.3.1.1). That means, fisher vectors of all features are concatenated to form a single vector before training stage. The combined feature is denoted as ImprovedDT+Proposed, while ImprovedDT refers to the combined form of HOG, HOF and MBH [130]. The experiment results are shown in Table 4.2 (UCF50) and Table 4.3 (UCF101). UCF50 results are shown as reported in the papers. In the case of UCF101, there had not been recent approaches validated on it until THUMOS Challenge. Therefore, the results shown in Table 3 are taken from technical reports which were submitted to the challenge by participating teams.

As for UCF50 data set, according to Table 4.2, our proposed method achieved comparable accuracies with recent approaches on action recognition. Except for Action

Bank [103] which uses high-level action representation, all other approaches exploit low-level representations. Kliper-Gross et al. [64] proposed Motion Interchange Patterns (MIP), which characterize the change from a motion to the next in multiple directions, obtained 72.7% accuracy by using their representation. Reddy et al. [97] achieved 76.9% by combining the MBH descriptor with scene context information. Solmaz et al. [114] reported 73.7% with a GIST3D video descriptor, an extension of the GIST descriptor [88] to video. Shi et al. [112] reported 83.3% using randomly sampled HOG, HOF, HOG3D and MBH descriptors. Our proposed approach outperformed most recent methods except for OrderedDT and ImprovedDT. OrderedDT refers to ordered trajectories which improved original dense trajectories [129] with the idea that not all trajectories but only some trajectories belonging to objects of interest should be selected. The better improved version of dense trajectory based method was proposed by its authors, Wang et al. [130]. In [130], Wang et al. proposed to detect and remove camera motion in order to compensate flow and reject background trajectories. This led to significant improvements for flow based descriptors (HOF and MBH). By combining the most successful features (ImprovedDT [130]) with our proposed features, we could achieve better performance than the states-of-the-art (obtained approximately 3% accuracy gain in the case of UCF50 and 5% in the case of UCF101 in comparison with using ImprovedDT only). This result demonstrates the efficiency of our proposed method as well as the complementarity of our features to conventional features.

TABLE 4.2: Comparisons to recent approaches on UCF50 data set. AP means Average Precision.

Method	AP
Action Bank (Jason et al.) [103]	57.9%
MIP (Kliper-Gross et al.) [64]	72.7%
MBH+Scene (Reddy et al.) [97]	76.9%
GIST3D (Solmaz et al.) [114]	73.7%
Shi et al. [112]	83.3%
OrderedDT (Murthy et al.) [81]	87.3%
ImprovedDT (Wang et al.) [130]	91.2%
Proposed	**83.6%**
ImprovedDT+Proposed	**93.5%**

The results on UCF101 data set are shown in Table 4.3. Note that Wang et al. have not reported the performance of their original features (improved dense trajectory based features, ImprovedDT [130]) on UCF101 data set in the literature. 83.1% is an accuracy

TABLE 4.3: Comparisons to recent approaches on UCF101 data set. AP means Average Precision. DT refers to Dense Trajectory based features [129].

Method	AP
Action Bank (Buffalo team) [128]	64.3%
MoSIFT (USC team) [116]	65.5%
DT with LC-KSVD (UMD team) [20]	65.8%
Proposed	**74.6%**
DT (USC team) [116]	74.8%
DT+MoSIFT (USC team) [116]	77.4%
OrderedDT (Canberra team) [80]	80.1%
Actons (Zhu et al.) [151]	80.9%
ImprovedDT (our implementation)	83.1%
ImprovedDT (Canberra team's implementation) [80]	83.5%
ImprovedDT+Proposed	**84.2%**
OrderedDT+ImprovedDT (Canberra team) [80]	85.4%
Late fusion of multiple features (Florence team) [58]	85.7%
ImprovedDT with SPM (INRIA team) [131]	85.9%

rate that we obtained by implementing Wang et al.'s method with their public code of feature extraction [5] and using our experiment settings. This result is nearly the same with the result that Canberra team achieved by their own implementation of ImprovedDT (83.5%) [80]. In THUMOS challenge, Wang et al. (INRIA team [131]) became the winner of the challenge as they achieved 85.9% accuracy rate. However, that result was obtained not by simply employing by their original features as described in [130]) but by combining multiple spatio-temporal pyramid levels of their features. As they described in their technical report [131], each level of spatio-temporal pyramid can help improve precision by 1 to 2%. Due to the exceptionally huge computational cost caused by applying spatio-temporal pyramids, we did not try spatio-temporal pyramids. They are not suitable for practical use, especially in the case of large-scale data.

In the challenge, USC team [116] used MoSIFT features [18] modeled with Fisher Vector and obtained 65.5% accuracy. By combining MoSIFT with original dense trajectory based features [129], they achieved 77.4%. Buffalo team [128] reported 64.3% by using high-level representation, Action Bank [103]. Florence team [58] used late fusion of multiple features (HOF and MBH aligned with dense trajectories, local SIFT pyramids on grayscale (P-SIFT) and opponent color keyframes (P-OSIFT)) and obtained second best performance in the challenge. Canberra team [80] used both OrderedDT [81] and ImprovedDT [130], obtained 80.1% and 83.5% respectively. By combining these features,

[5] https://lear.inrialpes.fr/people/wang/improved_trajectories

they achieved 85.4% (the third best). Zhu et al. [151] employed their mid-level representation, actons [150], which are weakly supervised learned via a max-margin multi-channel multiple instance learning algorithm. They reported 80.9%. UMD team [20] employed original dense trajectory based features [129] with LC-KSVD classifier [56] instead of SVM classifier and achieved 65.8%. LC-KSVD is an algorithm for learning a label consistent dictionary that represents each input signal as a sparse linear combination of dictionary entries. The important point is that while all of the above approaches as well as most of the teams which participated in the challenge employed conventional features such as Actons, MoSIFT and especially, dense trajectory aligned features, here we used our novel features extracted with our proposed method. Our features achieved comparable performance with conventional features (74.6%). In addition, as shown in Table 3, combining our features with dense trajectory aligned features (ImprovedDT) could obtain better performance than employing only ImprovedDT. This experiment results have shown that our features can capture different characteristics of videos from conventional features.

4.3.2 Experiments on Action Shot Extraction

In this section, we conducted experiments on our system of automatic extraction of relevant video shots for specific actions which is introduced in Chapter 2 with our proposed method of feature extraction in this chapter. In the previous chapters, we used our baseline method [85] to extract video features. In this chapter we show that by using our proposed method instead of the baseline, we can explore more representative and informative features of actions as we obtain many more shots corresponding to the actions.

In these experiments, we employed our system as we described in Chapter 2. Videos are selected based on tag co-occurance frequencies. Selected videos are then divided into video shots and spatio-temporal features are extracted from those shots. For feature extraction, we applied the settings which gain the best performance in above action recognition task. That means we extract features based on densely sampled SURF keypoints with our proposed methods of point selection and motion compensation. Following the previous experiments, we use Fisher Vector encoding to represent the videos.

PCA is first applied to reduce feature dimensionality by a factor of two. For each action, a subset of 100,000 features from its video shots are randomly sampled to estimate the GMM. The number of Gaussians is fixed as 256. Rankings of the video shots are calculated by VisualRank with uniform damping vector. The similarities between shots are calculated by Euclidean distances between their Fisher vectors.

As experiment data, we chose randomly 36 categories among categories with precision lower than 60% by the baseline features (Chapter 2.2). As the result of that choosing, the dataset of this experiment consists of: 6 categories with precision between 40% and 60%, 19 categories with precision between 20% and 40%, and 11 categories with precision below 20%. The experiment results are shown in Table 4.4. Here we compare the performance of our framework with 3 different settings: (1) using VisualRank with the baseline method of feature extraction [85]; (2) using VisualTextualRank (Chapter 3) with the baseline STFs [85]; (3) using VisualRank with our proposed method in this chapter. As shown in Table 4.4, our method outperforms significantly the baseline on most of the categories. The average pricision boosts greatly from 28% to 44%. Moreover, only by applying our method instead of the baseline, we could even achieve better performance than using VisualTextualRank which is the improved version of VisualRank. These results demonstrated that, our method could capture more informative characteristics of the actions compared to the baseline.

4.4 Conclusions

In this chapter, we proposed to improve a method of extracting spatio-temporal features which is able to efficiently select interest points and explore their characteristics. The experiment results validated significant improvement of our method over our baseline for action recognition task. The performance of our method has proved to be comparable to recent approaches on large-scale action recognition. Moreover, the proposed features are complementary with conventional features since the combination of proposed feature and conventional features obtained better results. We also applied our method to extract features of videos in our system of action shot extraction. The experiment results demonstrated the significant improvements of our method over the baseline as we could retrieve much more relevant shots.

TABLE 4.4: Experiments on automatic action shot extraction. VR refers to the original framework which is introduced in Chapter 2. VTR refers to the framework which applies VisualTextualRank (Chapter 3) instead of VisualRank. VR_NF refers to the framework which uses the method of feature extraction proposed in this chapter instead of the baseline.

action	VR	VTR	VR_NF
tie+shoelace	57	73	60
dive+sea	49	41	79
ski	49	60	52
dance+flamenco	45	53	60
dance+tango	41	41	58
play+trumpet	41	59	42
play+drum	40	45	46
swim+crawl	36	49	42
cut+hair	35	42	35
draw+eyebrows	32	32	48
hit+golfball	32	70	40
play+piano	32	34	62
row+boat	32	23	60
clean+floor	31	38	32
cut+onion	31	24	23
shave+mustache	31	30	28
plaster+wall	30	55	50
blow+candle	29	44	39
brush+teeth	28	27	41
catch+fish	28	59	61
drive+car	28	40	42
play+guitar	28	41	87
hang+wallpaper	26	46	45
jump+rope	26	49	55
ride+horse	24	15	56
sew+button	24	46	33
squat	19	34	45
row+dumbell	16	30	44
wash+dishes	15	39	34
comb+hair	14	26	15
swim+breaststroke	13	18	24
serve+tennis	11	27	35
swim+backstroke	9	9	30
serve+volleyball	7	40	35
swim+butterfly	7	9	20
cook+rice	6	11	16
AVERAGE	27.8	38.3	43.7

Chapter 5

Hand Detection and Tracking in Uncontrolled Videos for Fine-grained Action Recognition

5.1 Introduction and Related Work

5.1.1 Introduction

In the previous chapter, we developed low-level features by improving a method of spatio-temporal feature extraction. Focusing on human actions, we found that many of them are mostly operated by hands such as playing music activities ("play+piano", "play+guitar") or cooking activities ("roll+makizushi", "cut+onion") and so on. Our intuition is that, more than low-level features, if we could track hands and exploit hand movement features, we would obtain more sophisticated representation of actions and better results. In this chapter, we design a hand detector (tracker) for uncontrolled videos which are the target data of our research. We aim to additionally use hand features to deeper explore human actions. In fact, our objective is closely related to fine-grained recognition whose targets are actions performed by hands with different objects. Thus this chapter will focus on fine-grained action recognition.

In comparison to general recognition, fine-grained recognition has larger intra-class variability and smaller inter-class variability. In case of action recognition, even for the same

FIGURE 5.1: An example which shows the diversity of an action depending on related objects. We can see that "open" action varies on the target objects (from the top, "open" action with : umbrella, refrigerator, wine bottle and sliding window). Different objects cause disparate movements and directions of hands. This example shows the large intra-class variability of fine-grained action recognition.

type of action, operating with different objects may be related to different movements, directions and positions of body parts. For example, "play an instrument" action in the case of a guitar is completely different from the case of a piano. Another example is "open" action: it varies on the target objects (see Figure 5.1 for the illustration). Due to that characteristic, fine-grained action recognition requires deeper analysis of how human perform the actions with specific objects. In other words, in fine-grained action recognition, local manipulation motion details (e.g., subtle movements of hand in operating an object) are much more important than global information like in general action recognition.

In recent years, fine-grained activity recognition has attracted attention of research in action recognition field, especially after the release of the database for fine-grained activity classification of cooking activities in 2012 [100]. However, fine-grained action recognition has still been ignored. Particularly, there has been no action databases specifically designed for fine-grained action recognition. By applying our system while focusing on hand movement features, construction of a fine-grained action database can be expected to done without any difficulty.

In this chapter, we propose to classify fine-grained actions solely based on how people perform them with the objects using their hands. We demonstrate that hand related motion features are discriminative and representative for fined-grained human action recognition (see Figure 5.2 for an example). Thus, we want to take all possible arm/hand motions into consideration to represent the actions. Since motions of hands also contain those of arms, and in some cases, not the entire arm but only hands move, in order to

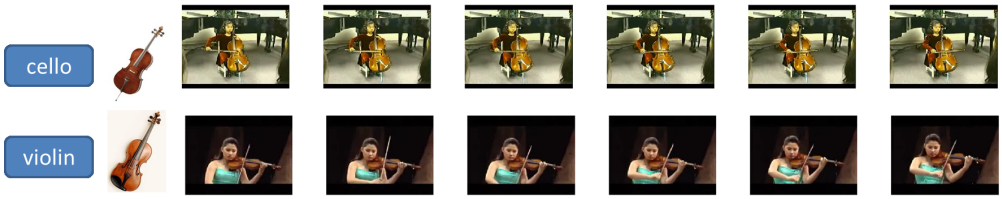

FIGURE 5.2: An example which shows that actions with objects involved may be recognized by motion features of arms/hands. The video shots are from UCF101 dataset. We can see that cello and violin look very similar, since they are in the same class of musical instruments (string instruments). Therefore, it is not an easy task to distinguish the actions related with them ("play" in this case) using the two instrument detectors. However, while playing them, people put their arms/hands in different positions and move them in different directions. Consequently, exploiting motion features of arms/hands can be expected to be able to help classify "play cello" and "play violin".

handle as many cases as possible, here we focus only on movements of hands. In this chapter, we propose a method of hand detection and tracking and apply its results to the problem of fine-grained action recognition.

Despite of much effort on hand detection and tracking, the task has still been known a tremendously challenging task. Hands are the most flexible human body parts compared to others. Their appearance can change unpredictably since they can be closed or open, and the fingers can have various articulations. Moreover, in videos, they are naturally the fastest moving body parts. Particularly, in realistic videos, there may be multiple moving objects and there also exists camera motion that can cause noise. This means that in realistic videos hands are very hard to detect due to a number of difficulties. In this chapter, we propose to exploit multiple cues including hand shape, skin color, upper body position and flow information to detect hands in videos. Our objective is to obtain 2D+t sequences of bounding boxes which tightly bound hands in the videos. We demonstrate that using motion features extracted only from hand regions can achieve comparable performance to using motion features extracted from the whole frame. That means hand motion features are the most informative representation of human actions involved with hand movements. Moreover, we further enhance fine-grained action recognition precision by exploiting displacement features of hands which belong to the most reliable hand tracks. To the best of our knowledge, we are the first to exploit only hand related motion features to the problem of fine-grained action recognition. This work of ours is reported in our paper [31].

To validate the effectiveness of our hand detector, we use VideoPose2 dataset [1]. This

[1] http://vision.grasp.upenn.edu/cgi-bin/index.php?n=VideoLearning.VideoPose2

dataset was originally developed for the challenging task of upper body estimation. To test the efficiency of our method on action recognition, we conduct experiments on playing-instrument group of UCF101 [61] dataset which is the only data that matches our purpose. UCF101 is one of the most challenging action dataset up to date with the large variations in human pose, object appearance, viewpoint, background and illumination conditions. Experimental results show the effectiveness of our approach.

The rest of this chapter is organized as follows. Chapter 5.2 describes our proposed method of hand detection and tracking. Chapter 5.3 explains how we apply the detecting and tracking results to the problem of fine-grained action classification. Experiments and discussions about their results are presented in Chapter 5.4. Finally Chapter 5.5 gives conclusions.

5.1.2 Related Work

Some recent fine-grained activity recognition approaches take into account interactions between human hands and objects [83, 149]. Since their approach relies on object detection, they have to learn object detectors of all related objects. This process requires costly annotations. Moreover, they do not consider the case when the objects are visually similar. For instance, in case of "play an instrument" action, since violin and cello share the same visual characteristics, their detectors are supposed to fail to distinguish them. Consequently, relying on object detectors may make it easy to confuse "play violin" action and "play cello" action. On the other hand, according to our method, disparate actions associated with different but visually similar objects can be classified (see Figure 5.2 for the illustration).

Hand detection is a popular topic in visual recognition which has a quite long history and a wide range of applications such as human computer interaction, sign language translators, human pose recognition and surveillance. In the early stage of development, hand detection technique required markers or colored gloves to make the recognition easier. Second generation methods used low-level features such as color (skin based detection) [82, 76] or shape [2]. Most recent works on hand detection in videos are performed in 3D [98, 125, 15, 87]. They employ depth information provided by depth cameras. As one in a few recent 2D hand detectors for videos, the hand detector proposed by Sapp et al. [105] exploits flow field. They propose to extract motion discontinuities

by computing the gradient magnitude of the flow field, and learn a linear filter via SVM using this motion discontinuity magnitude cue specific to hands. Hands are detected as regions with the max response from the detector at each frame location over a discrete set of hand orientations. In their work, the results of hand detection are only used as additional cues for limb localization since their final purpose is not hand detection but upper body pose estimation.

Most of hand tracking methods assume that hands are the most moving objects in an image frame. In [147], Yuan et al. proposed to use a temporal filter to select the most likely trajectory of hand locations among multiple candidates obtained by "block flow" matching. In [6], Baltzakis et al. proposed a skin color based tracker which allows the utilization of additional information cues such as image background model, expected spatial location, velocity and shape of the detected and tracked segments. The benefit of their trackers is that they can track hands in real time. However, their trackers only work under constrained environments where the background is unchanged, so that simply subtracting background can bring them enough cues to infer the most moving objects which refer to hands.

5.2 Proposed Method of Hand Detection and Tracking

5.2.1 Hand Detection

Here we aim to automatically estimate the hand locations using flow information and two trained detectors: a upper body detector and a static hand detector. As for flow estimation, we use DeepFlow proposed by[137]; for upper body detection, we employ Cavin's upper body detector [2]; for detection of static hands, we apply a state-of-the-art hand detector in still images proposed by Mittal et al.[79]. We improve their hand detector, originally developed only for hand detection in still images, to become a hand detector in videos by exploiting motion information and introducing upper body pose based spatial constraints.

[2] http://groups.inf.ed.ac.uk/calvin/calvin_upperbody_detector/

5.2.1.1 Method of The Baseline

We, first, briefly describe the method of static hand detection in [79], which is used as the baseline of our hand detector. According to [79], hand hypotheses are first proposed by three dependent detectors: a sliding window hand shape detector, a context based detector, and a skin based detector. Then, the proposals are scored by all three detectors and a trained model for scores is used to verify them. The hand shape detector was trained using Felzenszwalb et al. [38]'s part based deformable model with HOG features. The contexts here refer to the cues captured around the hands, especially the wrists. In order to learn the contexts, another part based deformable model [38] was trained from the hand bounding boxes which were extended to cover the wrists. The skin detector, first, builds a skin mask based on the skin color of face(s) detected by OpenCV face detector. It then detects skin regions by fitting lines using Hough transform and finding the medial axis of the blob-shaped regions. The hands are hypothesized at the ends of the lines.

The hand bounding boxes proposed by above three detectors are scored and combined as follows:

Hand detector score: the score obtained directly from hand detector.

$$\alpha_1 = \beta_{HD}(b) \tag{5.1}$$

where β_{HD} is the scoring function of the hand detector [38].

Context detector score: the score obtained by max-pooling over all bounding boxes which overlap with given hand boxes. The overlap threshold is set as 0.5.

$$\alpha_2 = \max_{b_h \in B_h} (\beta_{CD}(b_h)) \tag{5.2}$$

B_h refers to the set of context bounding boxes overlapping with the hand bounding box b_h. β_{CD} is the scoring function of the context detector [38].

Skin detector score: the score calculated by the fraction of pixels belong to skin regions in a given bounding box and denoted as α_3.

The three scores are combined into a single feature vector $(\alpha_1, \alpha_2, \alpha_3)$. This vector is then classified by a trained linear SVM classifier [12]. Finally, bounding boxes are suppressed depending on their overlap with other highly scored boxes using super pixel based non-maximum suppression. The superpixels are obtained by Arbelaez et al.'s method of image segmentation [3].

Mittal et al. trained their detector by using the data which was collected by themselves from various public image datasets including PASCAL VOC 2007 [3], PASCAL VOC2010 [4], Poselet [8], Buffy stickman [5], INRIA pedestrian [23] and Skin dataset [59], with 2861 hand instances for training and 660 hand instances for test in total. According to their experimental results, 48.2% of the test instances were correctly detected.

5.2.1.2 Proposed Method

Even though Mittal et al.'s hand detector achieved good performance, it needs two conditions about the data to work well: first, image resolution should be high and second, face should be easy to detect. Hands in images with good resolution commonly have clear shape, so that shape detector can be effectively employed. Moreover, most of faces in their test data can be seen from front view, so that face skin based hand detection is possible. However, here we have to deal with more complex and totally unconstrained data. In our data, many videos have low resolution or are taken under bad light condition, and faces are sometimes hard to be recognized. In such cases, we cannot find any of shape and/or color cues to detect hands. Thus, instead of employing Mittal et al.'s detector as it is, we propose to make it possible to work in such videos by introducing upper body based spatial constraints and motion information. The pose and position of detected upper body are used for two purposes: to estimate face region and to refine final detection results. On the other hand, flow information is exploited in two directions: to select upper body and to rescore hand hypotheses.

Our proposed method of hand detection is a three-step method which can be summarized as follows: (1) Detecting upper body by employing upper body detector and motion information, (2) Finding hand hypotheses based on multiple static cues, (3) Inferring the best hand hypotheses by exploiting motion cue and upper body based spatial constraints.

[3] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/
[4] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/
[5] http://www.robots.ox.ac.uk/ vgg/data/stickmen/

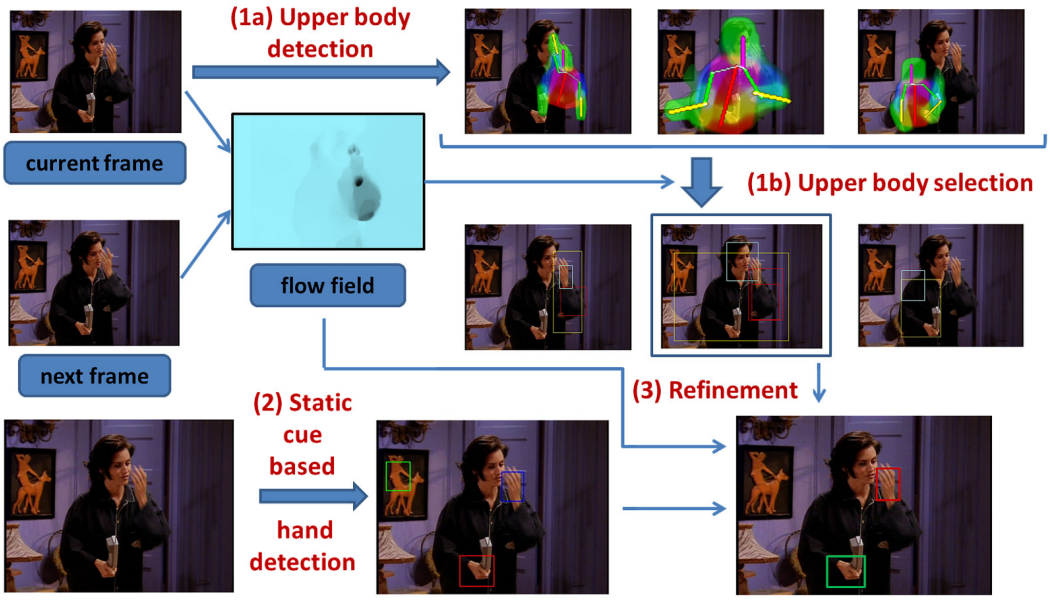

FIGURE 5.3: Illustration for our proposed method of detecting hands. (1) For a given frame, we first apply Calvin et al.'s upper body pose estimator to obtain proposals of upper body pose. (1a) Each proposal consists of sticks with different colors: pink, red, green and yellow which respectively refer to position of head, torso, upper arm and lower arm. To infer the best detection, we employ the motion of flow field. We segment the frame by magnitude of flow field to obtain regions with disparate movements. For each detected upper body, its score is redefined as the ratio of regions which are supposed to be hands. These regions should, first, be in motion, and second, be not too large or too small compared to the upper body. We show these regions by the red bounding boxes. The detection at the right side contains no motion, thus it is not supposed to be the good one. (1b) The middle detection is considered to be better than the left one since it contains more motion based hand hypotheses. Upper bodies and faces are marked by yellow and light blue bounding boxes respectively. (2) The face of the selected upper body (the middle one), along with hand shape and context, are then used as static cues for hand detection following Mittal et al.. The three best results obtained by hand detection using static cues are shown in the middle image of the last row. The best is represented by the red bounding box, the second best is green and the third best is blue. As we can see, the second best is a failed detection even though it has hand shape. (3) We refine the detection results by considering motion information and upper body position and obtain the final results as shown in the right image of the last row.

Refer to Figure 5.3 for the illustration of our proposed method of hand detection in videos.

For a given frame, at the first step, we apply Calvin et al.'s upper body detector and flow information to detect the most dominant upper body. This upper body detector has been demonstrated as a powerful human pose estimator and applied by many approaches recently. One of this detector's benefits is that it can estimate rather precisely the head position even when the face is hard to be detected. This detector returns several results, each of them contains position of head, torso and two limbs, and scores for each result.

However, the problem is that not all results returned by this detector are perfect, and the good ones are sometimes not highly scored. Moreover, we found that even when the faces and the torsos are quite precisely localized, it is not going that well for the limbs.

Assume that there exists at least one good prediction among the results, we infer it by introducing motion information and spatial constraints. We postulate the two following holistic hypotheses: (i) hypothesis about hands: hands are the most moving body parts in a upper body, and generally looked not big compared to the upper body from common views; (ii) hypothesis about the main actor: the main actor is generally in motion and captured in the easiest way to recognize. That means his or her upper body is likely in the middle of the image frame, and/or bigger than the others. Based on the first assumption, "good" upper body should cover moving regions, and these regions are supposed to include hands. For each detected upper body, we first segment it to regions with different movements by the gradient magnitude of the flow field. Regions that are smaller than upper body area multiplied by predefined area threshold t_a then become motion based hand hypotheses of that upper body. Score of a upper body is redefined as the ratio between areas of hand hypotheses which lie inside and outside that upperbody and finally normalized by area of that upper body. In the case that there are no significant movements (no moving regions with average flow magnitude being larger than flow threshold t_f), "good" upper body is simply selected based on the second assumption: the more centered and the bigger, the higher probability to be selected. In our experiments, t_a is fixed as 0.5 and t_f is fixed as 1.

The second step is hand detection based on multiple static cues using Mittal et al.'s detector. The face of chosen upper body is used to detect skin regions. At the final step, detection results of the previous step are rescored by introducing following flow score and upper body score.

Flow score: calculated as the average of gradient magnitude of flow of pixels in detection result. This score is normalized to have value between 0 and 1. We denote it as α_4.

Upper body score: determined by using spatial constraints based on position and area of upper body. It is calculated as percentage of area within the detected hand which overlaps with the upperbody. We also give penalties for detected hands that are

too big compared to the upper body. For such detections, their upper body scores are fixed as -1. We denote upper body score as α_5.

The final score of a given bounding box is defined as follows:

$$\text{Mittal}'\text{s detector score} + w_f * \alpha_4 + w_u * \alpha_5 \tag{5.3}$$

w_f and w_u are weights for flow score and upper body score respectively and determined by experiments. We tried all values from 0.1 to 0.9. Based on our experiments, $w_f = 0.7$, $w_u = 0.2$ obtained the best performance.

5.2.2 Hand Tracking

In order to reduce the computational cost, we process hand detection for only one frame in every k frames. Thus, we need to track obtained detections and automatically link detecting and tracking results over time. We also want to compensate for missing detections as well as search for the most reliable hand tracks.

We track h highest scored bounding boxes of every detection through L frames forwardingly. Since we obtain one detection in every k frames, we need to consider $L * h/k$ bounding boxes. We capture the persistence of hands over time with simple flow based tracking. We take the average flow of a bounding box to propagate it from a frame to the next. A reliable bounding box should overlap with many others during its propagation. A track of a detected bounding box will be employed if the bounding box overlaps more than 50% with any of h bounding boxes of at least n frames which have hand detection processed among L frames. In our experiments, $h = 2, L = 15, k = 3, n = 2$. Some example results of our method of hand detection and tracking are shown in Figure 5.4. As shown in Figure 5.4, we are able to not only compensate missing or undone detections but also remove false detections.

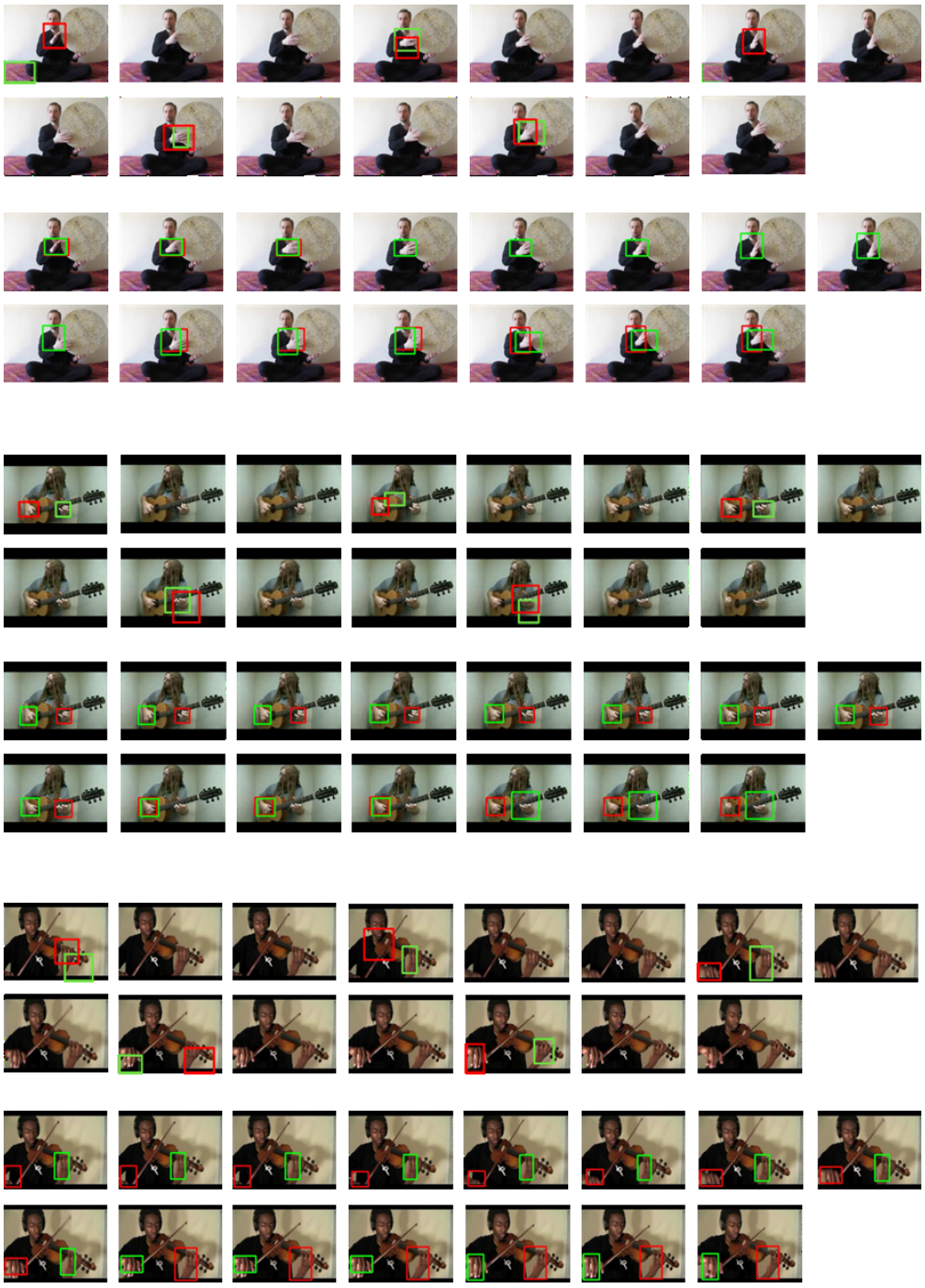

FIGURE 5.4: Example results of our method of hand detection and tracking on the group of playing instruments in UCF101 dataset. From the top, first the detection results, then the tracking results of "playing daf", "playing guitar" and "playing violin" are respectively shown. Among 15 consecutive frames, there are 5 frames which have hand detection processed. Only 2 top scored bounding boxes are shown. We track hands and keep hand tracks which overlap with at least 2 detection results. As the results, we can eliminate some failed detections and missing detections as well as obtain quite good hand bounding box sequences.

5.3 Applications on Action Recognition and Shot Extraction

Here we describe how we employ the results of hand detection and tracking obtained by our above method to classify actions in a given dataset or extract relevant shots of specific actions.

5.3.1 Overview of Our Approach

We first apply our proposed hand detector on each video in the data. To reduce computational cost, we do not perform hand detection for all frames but for only one in every k frames. In our experiments, k is set to 3. Next, based on the detection results, we track all highly scored bounding boxes through L frames to obtain connected and more reliable hand regions. In our experiments, L is set to 15. We then apply Wang et al. [130]'s method to extract dense trajectory aligned motion features and our hand displacement features from the detecting and tracking results. We conduct a Fisher vector for each type of extracted features. To combine different features, we concatenate their Fisher vectors. In case of action recognition, we train a multiclass linear SVM to classify the videos. By focusing only on regions which are expected to be the most related to the actions instead of considering the whole frame, we can improve action recognition precision. In case of action shot extraction, we apply our system which retrieves automatically relevant video shots for given actions from the Web data as we describe in Chapter 2. Instead of extracting only low-level features and using BoV representation as we did in the previous chapters, we detect hands and extract hand related features as we describe in this chapter.

In this chapter, we apply Fisher encoding methodology as described in [130]. The descriptor dimensionality is reduced by half using Principal Component Analysis (PCA). A subset of 256,000 features are randomly sampled from the training set to estimate the GMM and the number of Gaussians is set to 256. Each video is represented by a 2DK dimensional Fisher vector for each feature type, where D is the feature dimension after performing PCA. The following subsection explains in detail about the features.

5.3.2 Feature Extraction

We extract features based on 2D+t sequences of hand bounding boxes obtained by our proposed method of detecting and tracking hands. We apply Wang et al.'s method [130] to extract dense trajectories and their aligned motion features: HOF (Histograms of Optical Flow) and MBH (Motion Boundary Histograms) from all detecting and tracking results. Their method recently became the state of the arts for action recognition. According to their method, dense trajectories are obtained by tracking sampled points using optical flow fields for multiple spatial scales. HOF and MBH descriptors are computed within space-time volumes around the trajectories. HOF directly quantizes the orientation of flow vectors. MBH splits the optical flow into horizontal and vertical components, and quantizes the derivatives of each component.

In this chapter, we extract dense trajectories for only points which lie inside detected and tracked bounding boxes. If a frame has hand detection processed, its h highest scored detections will be used, otherwise, tracking results will be employed. We demonstrate that hand movements are discriminative and representative enough for actions operated by hands.

Beside dense trajectories and their aligned motion features, we extract hand track feature which describe the shape of hand trajectory by using average flow magnitude of hand regions in complete hand tracks. Given a trajectory of length L, its shape is described by a sequence $S = (\Delta P_t, ..., \Delta P_{t+L-1})$ of displacement vectors $\Delta P_t = (P_{t+1} - P_t)$. Here $P_t = (x_t, y_t)$ indicates the location of point P at frame t. The vector is normalized by the sum of the magnitudes of the displacement vectors:

$$S' = \frac{(\Delta P_t, ..., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} ||\Delta P_j||} \tag{5.4}$$

This vector is referred to descriptor for trajectory of point P. For our hand track features, only the center points of consecutive hand bounding boxes are taken into account. As the result, we obtain one descriptor for each hand track. While dense trajectories are extracted from all detection and tracking results, our descriptors are obtained from only reliable hand tracks. Thus even though they seem to be less informative than dense trajectories, they are expected to be a useful representation for actions as well.

5.4 Experiments and Results

We conducted experiments to validate the efficiency of first, our method of detecting and tracking hands and second, our method of classifying actions based on our results of hand detection and tracking. Experiment results show the effectiveness of our approach. As for extraction of action shots, since hand detection requires unexpectedly long computational time, especially due to computational cost of image segmentation and pose prediction, we could not apply our method to our large-scale data as it is. We intend to conduct experiments on action shot extraction after improving our method in terms of processing speed.

5.4.1 Experiments on Hand Detection

Here we want to show how our proposed utilization of static cues and motion information can improve hand detection in videos. We compare detection performance between our detector, Mittal et al.'s detector [79] which uses only static cues and Sapp et al.'s detector [105] which employs only motion information.

We validated our proposed method of hand detection on VideoPose2.0 dataset. The dataset consists of 14 video shots collected from movie source. It was originally developed only for the task of upper body estimation. Therefore, the exact locations of hands are not provided. We had to annotate hands in every frame by ourselves. There are 2453 frames and 3814 hands in total.

In these experiments, we detected hands in every frame. The performance is evaluated using average precision following Mittal et al. [79]. A detection is considered true if its overlap score is more than 0.5. The overlap score of a detected bounding box B_d is defined as $O = \frac{area(B_g \bigcap B_d)}{area(B_g \bigcup B_d)}$, where B_g is the annotated grouth-truth bounding box. The results are summarized in Table 5.1and some detection examples are shown in Figure 5.5.

First we validated the effectiveness of using faces of selected upper bodies instead of OpenCV face detector. As we can see in Table 5.1, the result was slightly improved. This is because VideoPose2.0 dataset has high resolution so that faces are usually big and clear enough for OpenCV detector to detect. The precision was significantly enhanced by introducing flow score. The first three rows of Figure 5.5 show the effectiveness of

FIGURE 5.5: Some examples of our detection results. We show two detections with best scores for each image frame. The best is shown in red, the second best is shown in green bounding box. The three upper rows of this figure show some detection examples in VideoPose2.0 dataset to compare the performance of the baseline, flow based detector and our detector (from the top, respectively). As we can see, our detector can detect more hands, especially hands blurred by their movements. Especially, in the case that there are more than one character (the second image from the right), our detector tends to detect moving hands since they are expected to belong to the main character. On the other hand, using only static cues gives higher scores for static hands which may belong to the character in supporting role (the second example from the right). Using only motion cues (flow) concentrates on detecting moving regions (which sometimes belong to other body parts or background objects). The last row of this figure shows some detection results for the group of playing instruments in UCF101 dataset.

TABLE 5.1: Results of hand detection. We conducted experiments on VideoPose2.0 video dataset and compared our hand detector with our baseline (Mittal et al.'s hand detector) and Sapp et al.'s flow based hand detector. Our (+upper body) means using face of selected upper body for skin detector. Our (+flow) means adding flow score to refine detection results. Our (+flow+body) means using our full proposed method which employs both flow information and body position based constraints to improve the final results.

Method	Precision
Mittal et al. [79]	41.7%
Sapp et al. [105]*	18.6%
Our (+upper body)	42.6%
Our (+flow)	45.5%
Our (+flow+body)	46.3%

*Their flow based hand detector

our detector over our baseline and flow based detector. While Mittal et al.'s detector sometimes failed to detect moving hands, mostly due to their unclear shape, our detector, by considering motion information, could detect them. This demonstrates that motion cue is extremely important for detecting hands in videos. However, employing only motion information can not robustly detect hands as Sapp et al.'s flow based detector

could achieve only 18.6% precision. Their flow based detector only concentrates on detection regions moving similarly to trained hands. Our proposed method which utilizes static cues and motion information achieved the best results. By adding upper body based spatial constraints, the precision was further improved. Our method of hand detection improved the baseline approximately 5%.

5.4.2 Experiments on Fine-grained Action Classification

Here we applied the results of hand detection and tracking to fine-grained action classification. We aimed to classify fine-grained actions based on how persons move hands to operate the actions. The actions should involve a large amount of hand movements throughout the time they are performed. For an action keyword, the actions should look somewhat disparate when performed with different objects. However, there was too few public data which matches our purpose. We found only the group of playing instruments in UCF101 dataset as suitable data for us to validate our method. UCF101 is a very challenging action data set as its video shots are collected from Web source. The data set has 5 action groups, but only the group of playing instruments is suitable for the purpose of fine-grained action classification.

The group of playing instruments in UCF101 dataset consists of 1428 video shots of actions of playing 10 types of musical instruments: cello, guitar, violin, daf, dhol, piano, tabla, sitar, flute and drum. The shots in each action category are grouped into 25 groups, where each group can consist of 4-7 shots of the action. The video shots from the same group may share some common features, such as similar background and similar viewpoint. We followed evaluation set up as suggested in the THUMOS Challenge (Chapter 4.3). We adopted their provided three standard train/test splits to conduct experiments. In each split, clips from 7 of the 25 groups are used as test samples, and the rest for training. The result of each experiment reported here is calculated as the mean of average accuracies over the three provided test splits. We train multiclass linear SVMs [121] to perform action recognition.

For data from UCF101 dataset, to reduce computational cost, we performed hand detection for only one in every three frames. To compensate the detections through the video as well as to find reliable hand tracks, we tracked hands as described in the Chapter 5.2.2. Since UCF101 is a large dataset without hand annotations, we could not

TABLE 5.2: Results of classification of actions in videos. DT means dense trajectories originally proposed in [130]. HDT means dense trajectories restricted to detected hand regions. HOF_{dt} and HOF_{hdt} refer to HOF features aligned with DT and HDT respectively (similarly with MBH). HT means our proposed hand track based displacement features. + means concatenating descriptors to a single descriptor before training and testing (early fusion).

Method	Precision
DT	66.7%
HOF_{dt}	83.8%
MBH_{dt}	86.6%
$MBH_{dt} + HOF_{dt} + DT$	87.3%
HDT	66.1%
HOF_{hdt}	81.4%
MBH_{hdt}	85.7%
$MBH_{hdt} + HOF_{hdt} + HDT$	86.2%
$\boldsymbol{HT}$	**36.0%**
$MBH_{dt} + HOF_{dt} + DT + HT$	87.6%
$\boldsymbol{MBH_{hdt} + HOF_{hdt} + HDT + HT}$	**88.5%**

validate the performance of our method of hand detection and tracking on this data in details. However, based on experimental results, we demonstrate that extracting features from regions specified to hands can achieve comparable performance to extracting from the whole frame. Our baseline in the experiments here refers to the method of extracting dense trajectories proposed by Wang et al. [130]. The results of our experiments are shown in Table 5.2.

As shown in Table 5.2, using only hand displacement features obtained 36% accuracy and using dense trajectories with their aligned motion features which were extracted from detected hand regions achieve comparable recognition performance to using original dense trajectories which were extracted from more regions. Even though precision rate of hand detection is not significant, imprecisely detected regions do not affect the final results that much since they are also informative (they are detected and employed by the baseline). The baseline, improved dense trajectory based method, extracts features only from foreground regions which move robustly. Instead of using all of those regions, in our method, we concentrate only on hand regions. The point is, despite of the fact that we use less information, we achieved comparable results to the baseline. That means our detection results are representative enough for the actions. Moreover, by combining multiple motion features considering hand positions, we could improve the baseline. This result demonstrates that the proposed method can extract the features

which have different characteristics from the conventional features. We also could prove that hand related motion features are particularly useful to recognize human actions.

5.5 Conclusions

In conclusions, we developed an effective hand detector in uncontrolled videos and obtained promissing results. Furthermore, we proposed to improve action recognition precision by additionally considering hand movements. Our experiment results showed that this consideration is effective. We try to deeply consider hand movements for the problem of fine-grained action classification in uncontrolled videos. To the best of our knowledge, we are the first to do that. This is the largest contribution of this chapter. Even if hand detection accuracy was only about 50%, employing the hand detection could help improve action recognition accuracy. This is a meaningful result even though the improvement is not remarkably significant. If hand detection accuracy is further enhanced, the benefit which action recognition gains from that enhancement can be expected to be larger.

However, according to our experiment results, even with the state-of-the-art approaches, fine-grained action classification on only a small-scale dataset could not achieve desirable performance. We hope that in the future, action recognition community will draw more attention to the problem of fine-grained action recognition.

As future works, we intend to improve our method in terms of processing speed, so that we can apply it to large-scale action shot extraction which is the final objective of this dissertation.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this dissertation, we proposed a framework of extracting automatically relevant Web video shots of specific action classes. Our proposed framework can help reduce tremendous human effort on building large-scale database for action recognition. To the best of our knowledge, we are the first to aim at automatic construction of such a large-scale action shot database. Although a few modest manual scanning may still be needed to use these video shots as training data, there is no doubt that human effort can be significantly reduced in comparison to fully manual database construction. We conducted large-scale experiments for 100 human actions and 12 non-human actions and obtained promissing results.

By our orginal framework without improvements of feature extraction and shot ranking, the average precisions at the top ranked 100 shots are respectively 36.6% and 14.9% for human actions and non-human actions. We proposed to introduce Web images to boost ranking positions of video shots which are visually similar to action related images. According to our experiment results, introducing Web images helps to enhance the performance for human actions by approximately 8% and for non-human actions by approximately 14%.

We also proposed to improve the conventional ranking method, VisualRank, by additionally exploiting textual information of the data. We applied our ranking method, VisualTextualRank, to shot extraction step of our framework. Our experiment results

showed that our method could help retrieve more important video shots at the top than the conventional one as the average precision for 100 human actions was enhanced by approximately 7% with VisualTextualRank.

Furthermore, we proposed to improve a method of extracting spatio-temporal features [85] which was employed in the above works. We addressed its problems such as the failured in feature extraction of videos containing camera motion or the holistic decision of motion threshold in the selection of interest points. Moreover, we proposed novel features by exploring more aspects of selected points. According to our method, we were able to efficiently select more interest points and explore better their characteristics. The experiment results validated significant improvements of our method over the baseline [85] for not only automatic extraction of Web action shots but also action recognition task. By using our method instead of the baseline for feature extraction step of our framework, the precisions of nearly 40 human actions boosted greatly from 29% to 44% on average.

We also designed a system of hand detection and tracking in uncontrolled videos. By tracking hands and exploiting hand movement features, we could obtain more sophisticated representation of videos and better results for hand motion based actions. The improvements over the baseline using the results of our hand detection system in the experiments on hand detection and fine-grained action recognition demonstrated the effectiveness of our system.

6.2 Future Works

As future works, first we intend to improve our system of hand detection and tracking in terms of computational cost to make it feasible for our framework of large-scale automatic extraction of Web action shots. Our system is supposed to help improve precisions for hand motion related actions. We also plan to adopt more context features such as human-object interactions or scene information to our methods in the future. By using these features and our proposed features to feature extraction step, our framework can be expected achieve better performance. We also want to try to use relevant actions shots obtained by our framework to train and evaluate classifiers for specific actions to

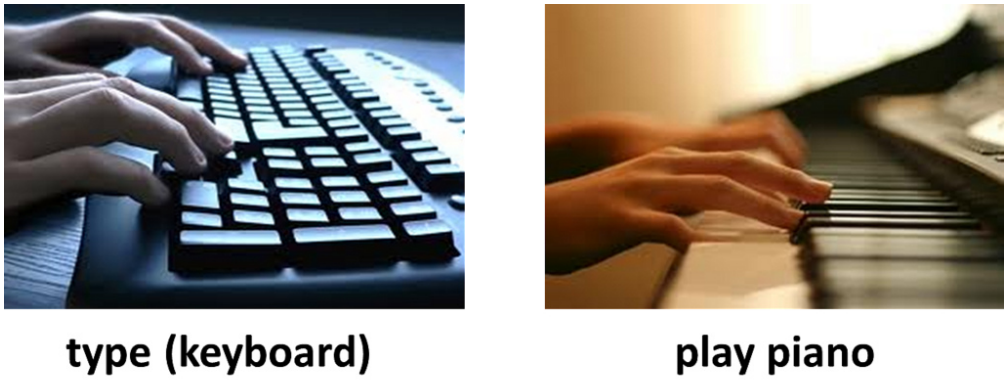

FIGURE 6.1: Images of "type" (left) and "play piano" (right). Even though they are textually unrelated, they should be categorized into the same action group since the action to perform them look really similar. While typing or playing piano, people usually use their fingers as shown in this figure.

show that these shots can be used as training data for action classification. To filter out irrelevant shots among top ranked shots, we intend to use crowd sourcing.

As one of benefits of constructing huge action databases using Web videos with minimal manual supervision, we started to consider conducting visual analysis of verbs. So far building an immense database for the conduction of that analysis has been considered as an extremely exhausted work. By adopting our proposed framework in this dissertation, construction of large-scale action databases requires much less human effort as well as labor cost. Thus visual analysis of verbs using Web videos becomes accordingly more feasible. Visual analysis of verbs can be understood as qualification of relationships between the concepts of verbs and the features of their corresponding video shots or more precisely, the characteristics of the actions performed in these shots. Understanding this relationship can help us categorize any verb based on the actions related to it. For instance, "type" as in "type keyboard" and "play piano" are totally unrelated according to their definition but in fact, they are "visually similar" - the action to perform them look similar. (see Figure 6.1). Moreover, we can also classify objects based on the way human interact them as the results of visual analysis between verb phrases which are comprised of their nouns and a specific verb. For example, we have "udon", "ramen", "onigiri", "hamburger" as the nouns which we want to categorize and we have "eat" as our verb. By analyzing the visual relationship between "eat udon", "eat ramen", "eat onigiri" and "eat hamburger", we can classify "udon" along with "ramen" to a group of food, and "onigiri" along with "hamburger" to another group (see Figure 6.2).

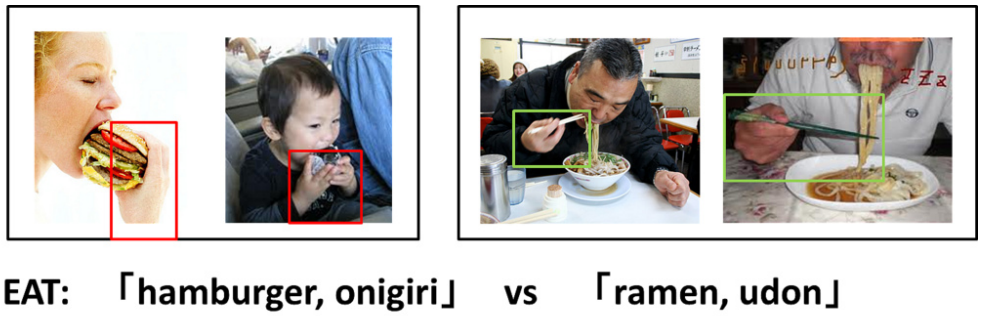

FIGURE 6.2: Images of people eating "hamburger", onigiri, "ramen", "udon" (respectively from left to right). When people eat "udon" or "ramen", they usually use chopsticks while for "hamburger" and "onigiri", they commonly use their own hands. Thus we can classify "udon" along with "ramen" to a group of food, and "onigiri" along with "hamburger" to another group.

Bibliography

[1] J. K Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73:428–440, 1999.

[2] A. Angelopoulou, J. G. Rodríguez, and A. Psarrou. Learning 2D hand shapes using the topology preservation model GNG. In *Proc. of European Conference on Computer Vision*, pages 313–324. 2006.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[4] I. Atmosukarto, B. Ghanem, and N. Ahuja. Trajectory-based fisher kernel representation for action recognition in videos. In *Proc. of International Conference on Pattern Recognition*, pages 3333–3336, 2012.

[5] 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. of ACM MM Workshop on Social Media*, pages 3–7, 2010.

[6] H. Baltzakis, A. Argyros, M. Lourakis, and P. Trahanias. Tracking of human hands and faces through probabilistic fusion of multiple visual cues. In *Computer Vision Systems*, volume 5008, pages 33–42. 2008.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. of IEEE International Conference on Computer Vision*, 2005.

[8] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. of IEEE International Conference on Computer Vision*, 2009.

[9] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2559–2566, June 2010.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the ACM International World Wide Web Conference*, 1998.

[11] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 41–48, 2009.

[12] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[13] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21(1):125 – 136, 2003.

[14] C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13:129–155, 1995.

[15] T. Cerlinca and S. Pentiuc. Robust 3D hand detection for gestures recognition. In *Intelligent Distributed Computing V*, volume 382 of *Studies in Computational Intelligence*, pages 259–264. 2012.

[16] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*. http://www.csie.ntu.edu.tw/c̃jlin/libsvm/.

[17] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1932–1939, 2009.

[18] M.-Y. Chen and A. Hauptmann. MoSIFT: Recognizing human actions in surveillance videos. Technical report, 2009.

[19] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary. Temporal sequence modeling for video event detection. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2235–2242, June 2014.

[20] H. Cho, H. Lee, and Z. Jiang. Evaluation of LC-KSVD on UCF101 action dataset. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[21] N. I. Cinbins, R. G. Cinbins, and S. Sclaroff. Learning actions from the web. In *Proc. of IEEE International Conference on Computer Vision*, pages 995–1002, 2009.

[22] N. I. Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *Proc. of European Conference on Computer Vision*, pages 494–507, 2010.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of IEEE Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[24] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proc. of European Conference on Computer Vision*, pages 71–84, 2010.

[25] J. Deng, W. Dong, R. Socher, J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 6 2009.

[26] H. N. Do and K. Yanai. Automatic construction of an action video shot database using web videos. In *Proc. of IEEE International Conference on Computer Vision*, 2011.

[27] H. N. Do and K. Yanai. Automatic collection of web video shots corresponding to specific actions using web images. In *CVPR Workshop on Large-Scale Video Search and Mining*, 2012.

[28] H. N. Do and K. Yanai. Large-scale web video shot ranking based on visual features and tag co-occurrence. In *Proc. of ACM International Conference Multimedia*, pages 525–528, 2013.

[29] H. N. Do and K. Yanai. Automatic extraction of relevant video shots of specific actions exploiting web data. *Computer Vision and Image Understanding*, 118(0):2 – 15, 2014.

[30] H. N. Do and K. Yanai. A dense SURF and triangulation based spatio-temporal feature for action recognition. In *Proc. of International Multimedia Modelling Conference*, volume 8325, pages 375–387, 2014.

[31] H. N. Do and K. Yanai. Hand detection and tracking in videos for fine-grained action recognition. In *Proc. of ACCV Workshop on Human Gait and Action Analysis in the Wild (HGAAW)*, 2014.

[32] H. N. Do and K. Yanai. VisualTextualRank: An extension of VisualRank to large-scale video shot extraction exploiting tag co-occurrence. *IEICE Transactions on Information and Systems*, 2015. (in press).

[33] P. Dollar, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. of Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[34] L. Duan, D. Xu, and S. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2012.

[35] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1491–1498, 2009.

[36] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, June 2008.

[37] J. D. R. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor. Improving "bag-of-keypoints" image categorisation. Technical report, 2005.

[38] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[39] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. of IEEE International Conference on Computer Vision*, pages 1816–1823, 2005.

[40] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. of European Conference on Computer Vision*, pages 242–255, 2004.

[41] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[42] R. Filipovych and E. Ribeiro. Robust sequence alignment for actor-object interaction recognition: Discovering actor-object states. *Computer Vision and Image Understanding*, 115(2):177–193, 2011.

[43] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. In *Proc. of British Machine Vision Conference*, pages 30.1–30.13, 2012.

[44] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[45] N. Haering, P. L. Venetianer, and A. Lipton. The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19(5-6):279–290, 2008.

[46] T. H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[47] B. Herbert, E. Andreas, T. Tinne, and G. Luc. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, pages 346–359, 2008.

[48] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, Aug 2004.

[49] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):797–819, Nov 2011.

[50] A. Jain, A. Gupta, M. Rodriguez, and L.S. Davis. Representing videos using mid-level discriminative patches. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2571–2578, 2013.

[51] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2013.

[52] F. V. Jensen, H. I. Christensen, and J. Nielsen. Bayesian methods for interpretation and control in multi-agent vision systems. In *Proc. SPIE 1708, Applications of Artificial Intelligence X: Machine Vision and Robotics*, pages 536–548, 1994.

[53] H. A. Jhuang, H. A. Garrote, E. A. Poggio, T. A. Serre, and T. . HMDB: A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2011.

[54] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(1):13–24, Jan 2010.

[55] Y-G. Jiang, Q. Dai, X. Xue, W. Liu, and C-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *Proc. of European Conference on Computer Vision*, volume 7576, pages 425–438, 2012.

[56] Z. Jiang, Z. Lin, and L.S. Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, 2013.

[57] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1870–1890, 2008.

[58] S. Karaman, L. Seidenari, A. D. Bagdanov, and A. D. Bimbo. L1-regularized logistic regression stacking and transductive CRF smoothing for action recognition in video. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[59] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 25–32, 2010.

[60] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725–1732, June 2014.

[61] S. Khurram, R. Z. Amir, and S. Mubarak. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[62] I. Kim, H. Choi, K. Yi, J. Choi, and S. Kong. Intelligent visual surveillance - a survey. *International Journal of Control, Automation and Systems*, 8(5):926–939, 2010.

[63] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. of British Machine Vision Conference*, pages 995–1004, 2008.

[64] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Proc. of European Conference on Computer Vision*, volume 7577, pages 256–269. 2012.

[65] T. Ko. A survey on behavior analysis in video surveillance for homeland security applications. In *IEEE Workshop on Applied Imagery Pattern Recognition Workshop (AIPR '08)*, pages 1–8, Oct 2008.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[67] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2003.

[68] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[69] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of IEEE Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[70] L. Li and L. Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2007.

[71] J. Liu, J. Luo, and M. Shah. Recognizing realistic action from videos. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2009.

[72] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney. Video event recognition using concept attributes, 2013.

[73] L. Liu, L. Shao, and P. Rockett. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, 2012.

[74] L. Liu, L. Sun, Y. Rui, Y. Shi, and S. Yang. Web video topic discovery and tracking via bipartite graph reinforcement model. In *Proc. of the ACM International World Wide Web Conference*, pages 1009–1018, 2008.

[75] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[76] C. Manresa, J. Varona, R. Mas, and F. Perales. Hand tracking and gesture recognition for human-computer interaction. *Electronic letters on computer vision and image analysis*, 5(3):96–104, 2005.

[77] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. of IEEE International Conference on Computer Vision*, 2009.

[78] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Proc. of ICCV Workshop on Video-Oriented Object and Event Classification*, 2009.

[79] A. Mittal, A. Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proc. of British Machine Vision Conference*, pages 1–11, 2011.

[80] O. V. R. Murthy and R. Goecke. Combined ordered and improved trajectories for large scale human action recognition. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[81] O. V. R. Murthy and R. Goecke. Ordered trajectories for large scale human action recognition. In *ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, pages 412–419, 2013.

[82] D. B. Nguyen, E. Shuichi, and T. Ejima. Real-time hand tracking and gesture recognition system. pages 19–21, 2005.

[83] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. June 2014.

[84] J. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting moving people from internet videos. In *Proc. of European Conference on Computer Vision*, pages 527–540, 2008.

[85] A. Noguchi and K. Yanai. A SURF-based spatio-temporal feature for feature-fusion-based action recognition. In *Proc. of ECCV WS on Human Motion: Understanding, Modeling, Capture and Animation*, 2010.

[86] E. Nowak, F. Jurie, W. Triggs, and M. Vision. Sampling strategies for bag-of-features image classification. In *Proc. of European Conference on Computer Vision*, pages IV:490–503, 2006.

[87] I. Oikonomidis, M. I. A. Lourakis, and A. Argyros. Evolutionary quasi-random search for hand articulations tracking. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2014.

[88] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[89] Paul Over, George Awad, Jon Fiscus, Brian Antonishek, Martial Michel, F. Smeaton, Alan, Wessel Kraaij, and Georges Quénot. TRECVID 2011 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Pro. of TRECVID 2011*, 2011.

[90] M. Pantic, A. Pentland, A. Nijholt, and T. Huang. Human computing and machine understanding of human behavior: A survey. In *Proc. of the International Conference on Multimodal Interfaces*, ICMI '06, pages 239–248, 2006.

[91] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[92] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *Proc. of European Conference on Computer Vision*, volume 3954, pages 464–475. 2006.

[93] O. P. Popoola and W. Kejun. Video-based abnormal human behavior recognition - a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):865–878, Nov 2012.

[94] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, October 2007.

[95] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.

[96] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[97] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[98] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proc. of ACM International Conference on Multimedia*, pages 1093–1096, 2011.

[99] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[100] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1194–1201, June 2012.

[101] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele. Script data for attribute-based recognition of composite activities. In *Proc. of European Conference on Computer Vision*, volume 7572 of *Lecture Notes in Computer Science*, pages 144–157. 2012.

[102] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[103] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1234–1241, 2012.

[104] F. A. Sanchez, J. A. Mensink, and T. . Improving the fisher kernel for large-scale image classification. In *Proc. of European Conference on Computer Vision*, pages 143–156, 2010.

[105] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1281–1288, June 2011.

[106] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[107] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proc. of IEEE International Conference on Computer Vision*, 2007.

[108] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proc. of International Conference on Pattern Recognition*, pages 32–36, 2004.

[109] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Pro. of the International Conference on Multimedia*, pages 357–360, 2007.

[110] W. Shandong, O. Omar, and S. Mubarak. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Proc. of IEEE International Conference on Computer Vision*, pages 1419–1426, 2011.

[111] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2595–2602, June 2013.

[112] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 2595–2602, 2013.

[113] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27*, pages 568–576. 2014.

[114] B. Solmaz, S. Assari, and M. Shah. Classifying web videos using a global video descriptor. *Machine Vision and Applications*, 24(7):1473–1485, 2013.

[115] C. Sun and R. Nevatia. Large-scale web video event classification by use of fisher vectors. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 15–22, Jan 2013.

[116] C. Sun and R. Nevatia. USC action recognition system with a large number of classes. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[117] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems 26*, pages 2553–2561. 2013.

[118] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1250–1257, June 2012.

[119] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[120] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. 2008.

[121] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, December 2005.

[122] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov 2008.

[123] H. Uemura and S. Ishikawa K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *Proc. of British Machine Vision Conference*, 2008.

[124] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha. Real-time visual concept classification. *IEEE Transactions on Multimedia*, 12(7):665–681, 2010.

[125] M. Van den Bergh and L. Van Gool. Combining RGB and ToF cameras for real-time 3d hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision*, pages 66–72, Jan 2011.

[126] J. C. Van-Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[127] E. Vig, M. Dorr, and D. Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *Proc. of European Conference on Computer Vision*, volume 7578, pages 84–97, 2012.

[128] R. Xu W. Chen and J. J. Corso. Action bank for large-scale action classification. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[129] H. Wang, A. Klaser, C. Schmid, and C-L. Liu. Action recognition by dense trajectories. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 3169–3176, June 2011.

[130] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. of IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[131] H. Wang and C. Schmid. LEAR-INRIA submission for the THUMOS workshop. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.

[132] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[133] L. Wang, Y. Qiao, and X. Tang. Mining motion atoms and phrases for complex action recognition. In *Proc. of IEEE International Conference on Computer Vision*, pages 2680–2687, 2013.

[134] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[135] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[136] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241, 2011.

[137] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *Proc. of IEEE International Conference on Computer Vision*, 2013.

[138] G. Willems, T. Tuytelaars, and L.V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. of European Conference on Computer Vision*, pages 650–663, 2008.

[139] J. Winn, A. Criminisi, and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *Proc. of IEEE International Conference on Computer Vision*, pages 1800–1807, 2005.

[140] K. Yanai. Generic image classification using visual knowledge on the web. In *Proc. of ACM International Conference Multimedia*, pages 67–76, 2003.

[141] K. Yanai and K. Barnard. Probabilistic Web image gathering. In *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 57–64, 2005.

[142] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1794–1801, June 2009.

[143] Q. Yang, X. Chen, and G. Wang. Web 2.0 dictionary. In *Proc. of ACM International Conference on Image and Video Retrieval*, pages 591–600, 2008.

[144] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2010.

[145] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2011.

[146] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Proc. of IEEE International Conference on Computer Vision*, pages 492–497, 2009.

[147] Q. Yuan, S. Sclaroff, and V. Athitsos. Automatic 2D hand tracking in video sequences. In *IEEE Workshops on Application of Computer Vision*, volume 1, pages 250–256, Jan 2005.

[148] X. Zhen, L. Shao, D. Tao, and X. Li. Embedding motion and structure features for human action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012.

[149] Y. Zhou, B. Ni, S. Yan, P. Moulin, and Q. Tian. Pipelining localized semantic features for fine-grained action recognition. In *Proc. of European Conference on Computer Vision*, volume 8692, pages 481–496, 2014.

[150] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *Proc. of IEEE International Conference on Computer Vision*, pages 3559–3566, 2013.

[151] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. A two-layer representation for large-scale action recognition. In *Technical Reports of the ICCV Workshop on Action Recognition with a Large Number of Classes (THUMOS'13)*, 2013.