VISUALIZING VIDEO SOUNDS THROUGH SOUND WORD ANIMATION
擬音語アニメーションによる動画音響の可視化手法

by

Fangzhou Wang
王方舟

A Master Thesis
修士論文

Submitted to
the Graduate School of the University of Tokyo
on February 20, 2014
in Partial Fulfillment of the Requirements
for the Degree of Master of Information Science and Technology
in Computer Science

Thesis Supervisor: Takeo Igarashi　五十嵐健夫
Professor of Computer Science

ABSTRACT

Sound information in video plays an important role in constructing audience experience. On the other hand, there are many circumstances where the audience cannot watch video with sounds. Subscripts are conventionally used as visual aids to provide the missing sound information. However, conventional subscripts are far less expressive for non-verbal sounds since it is designed to visualize speech sound. To address this problem, we propose a method to automatically recognize sound categories of non-verbal video-sounds and transform it to sound words for visualization. The sound word is animated in response to the change of the sound from moment to moment. Its position also changes dynamically depending on the position of the sound source object in the video. This will provide natural visual representation of non-verbal sounds with rich information about sound category and dynamics. The sound word animation is not only useful when watching video without sound, but also enjoyable for audience since it works as visual effects to enhance the visual scene.

論文要旨

動画に含まれる音の情報は，動画の視聴体験を構成する重要な要素である．一方で，実際に視聴者が動画を視聴する際は，状況によっては必ずしも音を伴って視聴できない場合が存在する．従来，このような状況下で音の情報を視覚的に補う手段としては字幕が用いられており，近年では音声認識技術を用いた自動字幕生成手法なども提案されている．一方で，これら字幕のほとんどが人の発した声を文字に書き起こしたものであり，声以外の一般的な音の情報に関する表現力は非常に乏しいのが現状である．そこで本論文では，動画中に生起する音の種類を自動で判別し，擬音語（オノマトペ）を用いて可視化する手法を提案する．生成された擬音語は音の変化に合わせてアニメーションされる．その表示位置も映像中の音源物体の位置に基づき動的に変化する．これにより，動画中に含まれる一般的な音の種類およびそのダイナミクスを自然な形で可視化し，視聴者に伝えることが可能になる．提案手法によって生成された擬音語アニメーションは，動画を音を伴わずに視聴する際に有用なのはもちろんのこと，映像シーンを積極的に誇張する効果を持つため，映像表現の一種として視聴者を楽しませる効果も持つと考えられる．

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Takeo Igarash. His sincere, fair, and honest comments have taught me things that are truly important and necessary to do a great work. It was a great experience to work under his supervision for this two and a half year. I will never forget what I have learned from him.

I also thank Dr. Sakamoto for his care and advice on my work, especially for advices for conducting user evaluation. It is his deep insight in the user study that made me understand its basics. Dr. Kunio Kashino and Dr.Hidehisa Nagano from NTT Research Lab gave me a number of fruitful advices to improve this work. Without the discussions with them this work could not have been polished to its current shape. Dr. Yang Li and Mr. Liyong Chen were my mentors during the internship at Google, Inc. They were very smart and kind. It was really of great fun to work under them.

Genki Furumi, Yuki Koyama, and Naoki Sasaki are who have joined the Igarashi Lab as a master student together with me. They were truly nice people and I was happy to spend time with them all. I also thank to Makoto Nakajima, Lasse Laursen, and other lab members for being friendly and having fruitful discussions with me. I will never forget all of them.

I thank all the friends I got along with. They had made my life filled with joy and gave me the energy to conduct research for this two and half years. Lastly, I would like to thank my parents for supporting me for as long as 24 years for my life and study. It could never be expressed by word how much I am thankful to them.

Contents

1 Introduction **1**

2 Related Work **4**
2.1 Sound Processing and Visualization 4
2.1.1 Speech Visualization 4
2.1.2 Music Visualization 4
2.1.3 Environmental Sound Recognition and Visualization 5
2.2 Visual Processing 6
2.2.1 General Object Recognition 6
2.2.2 Video Annotation 6
2.2.3 Animating Texts 6

3 Prototype System Design **8**
3.1 Target Video 8
3.2 Animation Design 8

4 Algorithm **13**
4.1 Overview 13
4.2 Sound Processing 14
4.2.1 Sound Identification 14
4.2.2 Volume Estimation 16
4.3 Animation Generation 17
4.3.1 Volume Envelope Segmentation 17
4.3.2 Generating Animation Item 18
4.4 Animation Positioning 21
4.4.1 Video Cost Field 22
4.4.2 Animation Cost Field 23
4.4.3 Position Optimization 24
4.4.4 Updating Video Cost Field 27

5 Results **28**
5.1 Resultant Video Sequence 28
5.2 Performance 32

6 User Study **33**
6.1 User Study with Crowdsourcing 33
6.2 Questionaire Design 34
6.2.1 Comparison of Videos With and Without Sound Word Animation 34
6.2.2 Comparison of Different Animation Styles 36
6.3 Results 36

6.3.1 Comparison of Videos With and Without Sound Word Animation . . . 37
6.3.2 Comparison of Different Animation Styles . . . 39

7 Discussion **47**
7.1 Design Guideline . . . 47
7.1.1 Font Style . . . 47
7.1.2 Choice of Sound Word . . . 47
7.1.3 Positioning Style . . . 48
7.2 Constructing Natural Audience Experience . . . 48
7.2.1 Choosing Suitable Design . . . 49

8 Conclusion and Future Work **50**

References **51**

Chapter 1

Introduction

These days, video is being more and more popular as entertainment contents. Hundreds hours of video are uploaded every minutes to online video hosting services such as YouTube [85] or Dailymotion [24], and billions hours of video uploaded are watched each month [68]. As smart phones or tablet PCs become more and more popular, you can watch, capture, and share video contents anywhere anytime to have fun. There is no doubt that more and more people will become consumer and producer of video contents along with improvement of these mobile devices and video processing technologies.

The early stage of videos are "silent films" and were initially presented without sound. The silent films were later presented along with musics, until all of them were replaced by "talkie" films that have sound tracks synchronized with the video frames [10]. These days, most of video contents are composed of both sound tracks and video frames. A sound track synchronized with video frames often plays an important role in providing an integrated "audiovisual" experience to the audience. Not only human-spoken sound in the video itself contains important information, but even non-verbal sounds, such as engine noise in car race video or the roar of scoreline in a soccer game, would have a large influence to the audience experience.

Although sound information is essential elements in designing the audience experience of a video, it is not always the case that the audience is able to get a full access to the sound. The hearing impaired is either physically impossible to hear at all, or require acoustic aids or sound to be played with large volume. Even for unimpaired person, still there are cases that sound is not available, such as:

1. Video sounds need to be muted when displayed at public or silent space s.t. in a train or an authentic bar.

2. The audiences want to watch multiple video contents at the same time, e.g. multi-view functions on recent TV devices.

3. The audience may not want to wear headphones because he or she thinks it troublesome, or no headphone available at all, when they are required to keep silent.

In these cases, the audiovisual experience of video contents would be strongly depressed by the lack of sound information. One conventional way to address this problem is to add text captions. Most common style of text captions is showing a static text sentence that describing the contents of sound at the bottom or side of the screen, and changing the text sentence from time to time along with the changing of sounds. This traditional style of text caption is fairly effective

Figure 1.1: An example of the visualization result by sound word animation.

for visualizing dialogues or spoken words by human. This is because the most important information for audience is not the voice sound itself, but the linguistic information behind it. On the other hand, one of the largest problems with the traditional style of text caption is that it has far less capability to describe non-verbal sounds, such as environmental sounds or sound effects [60]. It is not clear how to describe information of non-verbal sounds with texts, since there is no clear linguistic information such as grammar or vocabulary behind it. Also, unlike spoken words, it is very important for non-verbal sounds to describe not only its category, but also dynamics such as intensification or attenuation, because a large part of non-verbal sound information lies within its dynamics rather than mere categories. It is nearly impossible to describe this kind of dynamics with the traditional style of static text captions.

To address this problem, we propose to 1) transform non-verbal sounds into "Sound Word" and 2) animate the generated sound words to describe the dynamics of the sound. Figure 1.1 shows an example of visualization result. A sound word, or onomatopoeia, is a word used to describe non-verbal sounds. Some studies show that sound words are more effective in sound searching tasks compared to just using a category name such as "Wood impact sound" or "A buzzer" [77]. Sound words are especially popular in comics, which is in need to describe many action scenes with intensive sounds and motions in paper material. It is also used in some TV dramas or programs to enhance the visual and auditory actions of characters [2]. While for all of these contents the sound words are manually added by the creator, our final goal is to automatically add sound word animation to a video by analyzing its video frames and sounds. The proposed algorithm recognizes the category of sound emerging in the input video and converts it into sound words, and analyze the intensification and attenuation of the sound to generate sound word animation. The algorithm could also recognizes sound source objects in the video frames to determine the position in the video frames where the sound word should to be added. The proposed method would be useful for online video hosting services, where the amount of video uploaded is extremely large and it is impossible to added annotations manually to all of

them.

We also conducted a user study to verify the effectiveness of using sound words for non-verbal sound visualization. We have conducted the study on a crowdsourcing service [7] with over 700 anonymous crowdworkers. The result shows that compared to traditional static text captions, the sound word animation is more effective in providing the dynamics of sound volume, is a more natural representation of sound, and is more useful for video without sound. They also provided qualitative comments on how different style of sound word animation effects the audience experience and suggestions for improvement.

The main contribution of this work is as follows:

1. A method to automatically generate sound word animation from input video sound.

2. A method to dynamically and automatically position the generated sound word animation to the input video regarding its visual contents.

3. A qualitative user study to verify the effectiveness and clarify design guidelines of automatically-generated sound word animations.

Chapter 2

Related Work

2.1 Sound Processing and Visualization

Recognition and visualization of sounds have been an important challenge in the field of computer science for a long while. The research field could be roughly divided into three depend on the category of the sound to process: speech, music, and environment sound. Our research has a strong relation to environmental sound processing. In following subsections we describe the related work for those three target categories respectively.

2.1.1 Speech Visualization

A traditional way to visualize speech sound in the video is to show a static text caption. These days, a lot of video content available on DVD includes closed captions in several languages, which the audience can switch on and off. The text caption is also available on TV broadcasting in some country [4]. However, a traditional static text caption has several problems such that 1) it requires to be prepared manually, or 2) it is difficult to describe the volume dynamics, speed, or emotion of the sound. Some online video hosting service provide a functionality to automatically generate text captions from video sound [85]. This functionality could reduce the tedious work of generating and assigning text script to video by hand. Hong et al. [40] proposed a method called "Dynamic Captioning". In their method, the text caption is placed near the face of the speaker to clarify the speaker of the dialogue, and the color of text is changed word-by-word depending on the speaking speed. The sound volume is also visualized as a bar chart on the side of text captions. They achieve this by taking time-aligned subtitles and script of each character as input, and exploiting speech recognition and face detection method to figure out the place of the speaker's face for each dialogue. There are other studies proposing to visualize speech by using animation of a virtual 3D head model [61], [13] instead of text captions, which is especially beneficial for hearing impaired persons.

2.1.2 Music Visualization

Various method has been proposed for music visualization. Most popular of them would be wave visualizer bundled with some music players [76, 58]. A traditional approach is to produce a static image that represents the feature of music [31, 17, 46]. Goto et al. proposed several visualization methods for promoting user to interact with musics to enhance the auditory experience [38]. Some studies [32, 41, 52] proposed methods to generate a new music video by taking video files and audio clips as input. Other studies use public photos

to generate music videos [65, 16] or personal photos to generate a slide show [82]. While these studies focus on generating visual contents exploiting music information, there are also studies focus on visualizing groups of music such as music archives [57]. However, few studies have been conducted for providing aid for audiences who is inaccessible to sounds. Nanayakkara et al. [53] proposed a method to use visual aids and a haptic chair to provide sound information to the hearing impaired. Although the efficiency of the proposed method is shown through their user study, it requires a special vibrating chair, which is unavailable on the market.

2.1.3 Environmental Sound Recognition and Visualization

The word "environmental sound" may include all the meaningful sounds except speech and music. Several research has been done on how to identify different environmental sounds in different categories[35, 22, 23]. Through the development of robotics, the importance of environmental sound processing is being higher and higher [21]. Its applications not only limited to robotics, but also include smart mobile applications [49], home automation [78], surveillance and security applications [25]. Several research focus on providing aid for hearning impaired or elderly person, while most research focus on how to support their daily lives by notifiying and visualizing environmental sound on portable displays [**?**, 84], or a PC display[11]. Matthews et al. [51] conducted a survey on preferred designs of these systems. In summary, no previous research is focused on visualizing environmental sound in videos.

Sound words, or onomatopoeia, is an important tool in environmental sound processing [80]. Wake et al. [77] proposed a system that enables sound word to search environmental sounds, and reported that using sound words for searching could achieve higher performance than merely using the category name of the sound. Ishihara et al. [42] proposed a method to directly generate various Japanese sound words from the input sound using Japanese sound-imitation syllables. Ito et al. [43] proposed to use sound words to control robot motion. Komatsu et al. conducted a user study on Japanese sound words and quantified these sound words using parameters such as sharpness, softness, dynamic, and largeness [45]. Terashima et al. [71] proposed a method to allow comic creators to easily annotate sound words depend on their drawing. However, little research is conducted on visualization methods using sound words. Yamamoto et al. [83] proposed a method to visualize environmental sound using different types and size of fonts. This seems to be the only work that focused on visualization by sound words. Similar as proposed by Ishihara et al. [42], their methods directly transform audio waveform to sound words. The size of the font is determined by sound volume, and color and type of fonts are determined by parameters such as volume, sound pitch or center of frequency. However, their methods is limited to transforming a short sound with under an well-cotrolled experimental environment. Therefore, their method could hardly deal with "dirty" sounds in video where multiple sounds in different categories are mixed together. Furthermore, the generated sound words are static and could not depict the dynamics of sustained sound (e.g. engine sound of a car). The method we propose is different from them in that it can 1) deal with the mixture of sounds, 2) visualize the dynamics of sound by animating the generated sound word, and 3) position the sound words into appropriately by taking the visual position of sound source into consideration.

2.2 Visual Processing

2.2.1 General Object Recognition

The research field of general object recognition could be roughly divided into classification or detection. While the purpose of classification is to label an image with appropriate category (e.g. [86, 79]), detection is to judge whether an object of specific category exists in an image and determining the bounding box of the object (e.g. [30, 73]). Generally, detection problem is more difficult than classification. The result of the PASCAL Visual Object Classes challenges [29] 2012 (PASCAL VOC 2012) shows while the state-of-art methods for classification would achieve approximately 60-90% average precision, detection method could only achieve 20-60% average precision thorough 20 object categories [27]. The performance is far behind the state-of-the-art face detection [70, 75] methods that could achieve over 90% detection rate, or human detection methods [26, 64] that could achieve around 50-80% detection rate. However, compared to the approximately 10-40% average precision through 10 object categories in the result of PASCAL VOC 2006 [28], the improvement of this research field is obvious. In this work we utilize the method proposed by Felzenszwalb et al. [30], which is released as a MATLAB library with trained model, to implement the research prototype system.

2.2.2 Video Annotation

Video summarization is a major application of video annotations. Goldman et al. [36] propose to use texts and arrows to summarize the motion of camera and objects in video into one static image. Nienhaus et al. [54] proposed to use dynamic glyphs to summarize a 3D animation by specifying the animation of the moving object with a graph structure. Some other studies focus on motion data of the human body and visualize the body movements with arrows [15, 20]. While all of these methods focus on summarization, Goldman et al.[37] proposed to exploiting particle tracking techniques for video sequence and enabled the user to easily annotate paints or texts in the video. Similary, Santosa et al. [63] proposed to exploiting trajectory analysis of objects in video to propagate paintings to object through multiple frames. Both of latter two techniques require the user to specify the annotation item, while our focus is to automatically generate and place sound word animations as annotation items for sound visualization.

Several studies have been conducted to determine an optimal position of video annotation. Some these methods focus on annotation for videos [74], while others are for real-time AR (Augmented-Reality) applications [62, 56]. All of these methods assume the annotation to be static text information with considerable duration to be posed on screen. On the contrary, our method focus on positioning animating objects that have shorter durations and dynamically change its size and opacity from time to time. We take these dynamic animation parameters into consideration during positioning, which is not described in these previous work. Our method could generate better positioning results by exploiting more thorough analysis on the input video proposed by these works.

2.2.3 Animating Texts

Lee et al. [47] are the first group that proposed a system for generating animated texts. These animated texts are called “Kinetic Typography”, and they argue they have “ability to convey emotion, portray compelling characters, and visually

direct attention" [47]. They also built a graphical interface for the user to easily generate these animated texts [33], and shown it is applicable for text-based interpersonal communication through user study[48]. Matsushita et al [87, 50] proposed systems to annotate animated sound words into comics. While all these systems focus on helping the author to create animated texts manually, Strapparava et al. [69] proposed to analyze the linguistic emotion of input text for automatically generating animation. Note that our research is different from theirs in that we take the sound information as input for animation instead of linguistic information. Rashid et al. [59] proposed a framework to animate text captions in video to convey emotion. They conducted a user study for animated text captions[60] to measure how the animation design effects audience experience. They concluded that "enhanced" captions of which the position is fixed at the bottom of video is most preferred, and text based sound effects confused participants. However, their study has several flaws as follows: 1) They proposed the design of animation caption to convey emotion, but did not describe how to design animation for non-verbal sounds. The design of animated sound words could be arbitrary and the result could be biased by its design. 2) The study is limited to the comparison between two different video contents, therefore the result could be biased by the difference of video contents itself. 3) The resort only claims that the animated sound words are less likable than emotionally animated texts. This does not mean it is negatively perceived by the audience. Our user study described in chapter 6 shows that the animated sound word is still considered useful by the audience when watching videos such as car racing without any sound.

Chapter 3

Prototype System Design

There are an infinite number of sound and video categories. Similarly, there is infinite design space for how to overlay information on a video. This research does not focus on how to recognize sound and visual objects, but how to generate a good visual aid for non-verbal sounds in video by making use of computer vision and audition techniques. Therefore, we decided to limit the target video to a specific category, and animation to be a simple combination of scaling and opacity control of sound words, in order to simplify the problem. Please note that this limitation does not mean that the proposed algorithm has little scalability, provided a better sound or visual recognition algorithm will be applied in the future.

3.1 Target Video

In this research the target video category is limited to "car racing" only. There are mainly two reasons for this. First, from auditory perspective, car video contains very limited categories of sound and is easy to process. Generally, the less categories of sound are to be identified, the easier the problem would be. We found that most of the sound in a typical car racing video could be identified as either "Engine sound" or "Squeal sound (Drifting sound)". This means that a reasonable visualization result could be generated for car videos even if only two sound categories are to be identified. Second, the major generic object detection algorithms have higher performance in detecting cars than other object categories [27]. This would help the research to put its focus not on object detection itself, but on how to utilize the result of object detection for sound word animation generation and positioning.

3.2 Animation Design

Although there are some video contents that use sound words to visualize the sound [2], there is no "de facto standard" form of sound word animation. The simplest way to generate the sound word animation would be a direct mapping of the sound volume to the size of generated sound words. However, we found that this way of animation generation has several visual problems as follows:

Figure 3.1: Visual problems of a direct mapping of sound volume to size. The frame rate of video is 30 fps. (Top) Captured every 30 frames. Little change of the size of the sound word could be found in total 2 seconds. (Middle) Captured every 4 frames. The size of the sound word drastically changes in a very short time. (Bottom) Captured every 90 frames. The sound word remains visible for a long duration and makes it difficult to change its position.

1. The sound with a long attenuation time would make the sound word animation shrinking too slowly. This is inconsistent with auditory perception of human, which loses attention to continuous sound more rapidly (Figure 3.1-top).

2. It captures the dynamics of sounds too sensitively and generates a "Bouncing" effect that is visually noisy and unnatural (Figure 3.1-middle).

3. It is difficult to change the position of animation from time to time because each animation item a long duration (Figure 3.1-bottom).

In order to avoid these problems, we decided to adopt a more sophisticated animation design as described in following sub-subsections. Although the design may seem arbitrary, through generating mockup results we found that it is one of the simplest form of a sound word animation that could be acceptable for the audience. We adopt this design as an initial research prototype to figure out whether the idea of a sound word animation as a visual aid is beneficial at least in some case. We will discuss deeper how the design effects audience experience in chapter 6 and 7.

Duration of Animation

The whole video sound is divided into sound segments depending on its category and volume, and an animation item is generated for each segment (Figure 3.2). This makes it easier for the system to dynamically position the generated sound words based on the position of the sound source object.

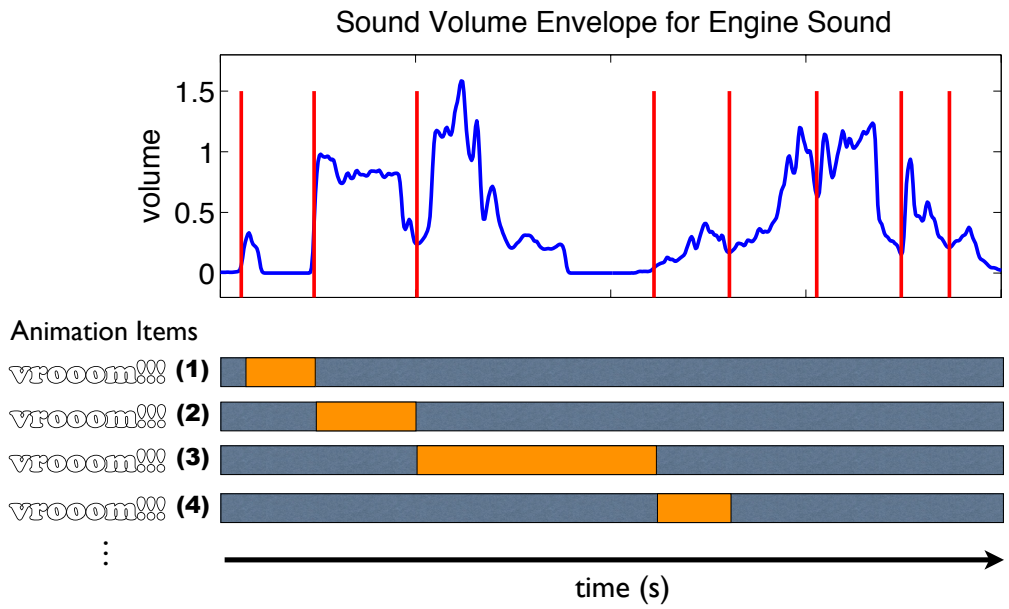

Figure 3.2: Animation items are generated by dividing the sound volume envelope for each sound category.

Design of sound word

Sound words are pre-rendered texts by the system for each sound category. We designed two pairs of sound word image for the squeal and engine sound of car racing video, one in English and the other in Japanese (Figure 3.3).

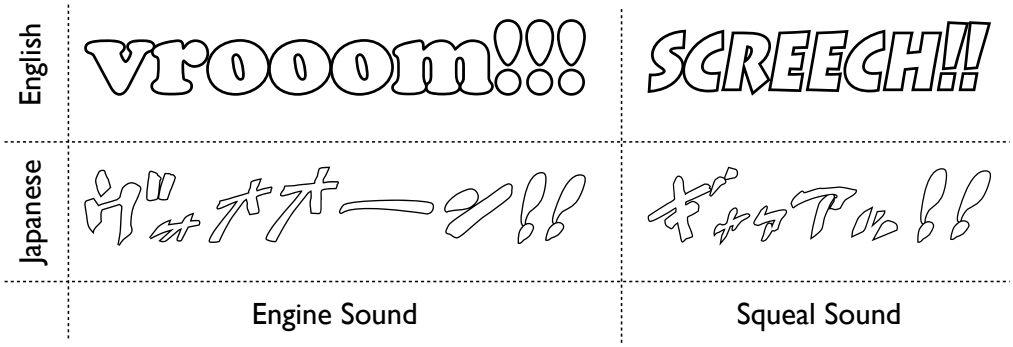

Figure 3.3: Different design of sound words based on its category and language.

Parameters of animation

The animation item is generated by changing the size and opacity of a static sound word image based on the change of sound volume. Different designs are respectively applied to intensification and attenuation phases of the sound (Figure 3.5).

(Intensification phase) the opacity of animation item is 100% and the size increases following the sound volume.

(Attenuation phase) The opacity of animation item decreases following the

sound volume. The size either slowly decreases when the attenuation phase is long (Figure 3.5-top), or extends intensification phase when the attenuation phase is short (Figure 3.5-bottom).

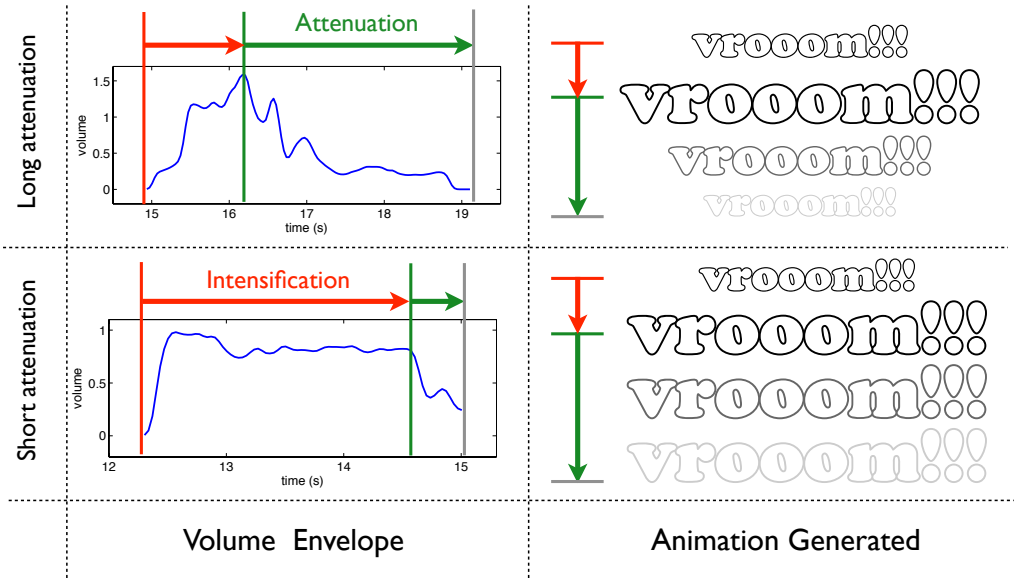

Figure 3.4: Different animation styles based on the length of the attenuation phase.

Positioning Style

Three positioning styles are designed. We have implemented all these styles in the prototype system, and can easily switch between different positioning style.

(Static Positioning) The position of sound word animation is always fixed (Figure 3.5-top).

(Dynamic Positioning without Movement) The position of animation items change based on the position of the sound source object. Each animation item does not change its position once appeared. (Figure 3.5-middle).

(Dynamic Positioning with Movement) Similar as above, but each animation item could change its position after it appeared following the movement of the sound source object (Figure 3.5-bottom).

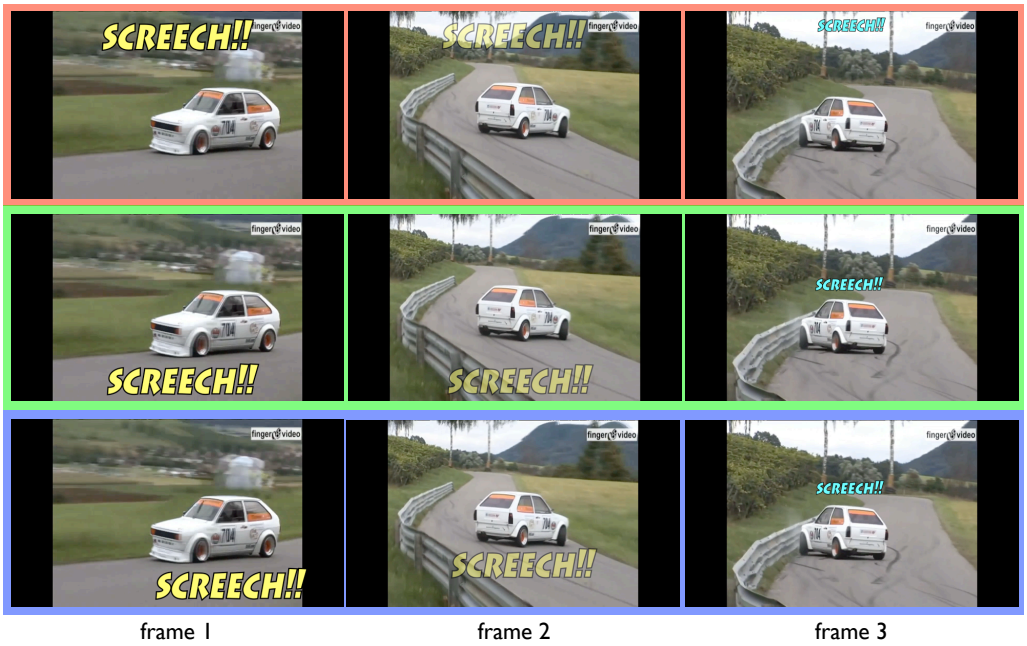

Figure 3.5: Three types of positioning style. Two animation items are presented in different colors. (Top) Static Positioning: the position is always fixed. (Middle) Dynamic Positioning without Movement: the position of animation items dynamically change based on the position of sound surce object (between frame 1-2 and frame 3). (Bottom) Dynamic Positioning with Movement: the position of animation items also move after it once appeared (between frame 1 and 2). Note that the animation items in this figure are manually positioned for description.

Chapter 4

Algorithm

The algorithm for generating and positioning sound word animation is described in this chapter.

4.1 Overview

The algorithm we propose takes a video with a sound track as input, and outputs a video with sound word animation annotated. For simplicity of implementation, we transform the input video size to be 1280x720 pixels with frame rate of 30 fps, and input sound to a moraural sound be with sampling rate of 22.05kHz and quantization bit rate of 16bit. Figure 4.1 describes the overview of the aprocessing flow of the algorithm. It consists of three parts: Sound Processing, Animation Generation, and Animation Positioning. First, sound processing part identifies sounds throughout the video and computes time-series posterior probability for each pre-defined sound category (e.g. Engine Sound). The algorithm therefore estimates the sound volume for each pre-defined sound category. Second, animation generation part exploits the result of volume estimation for each sound

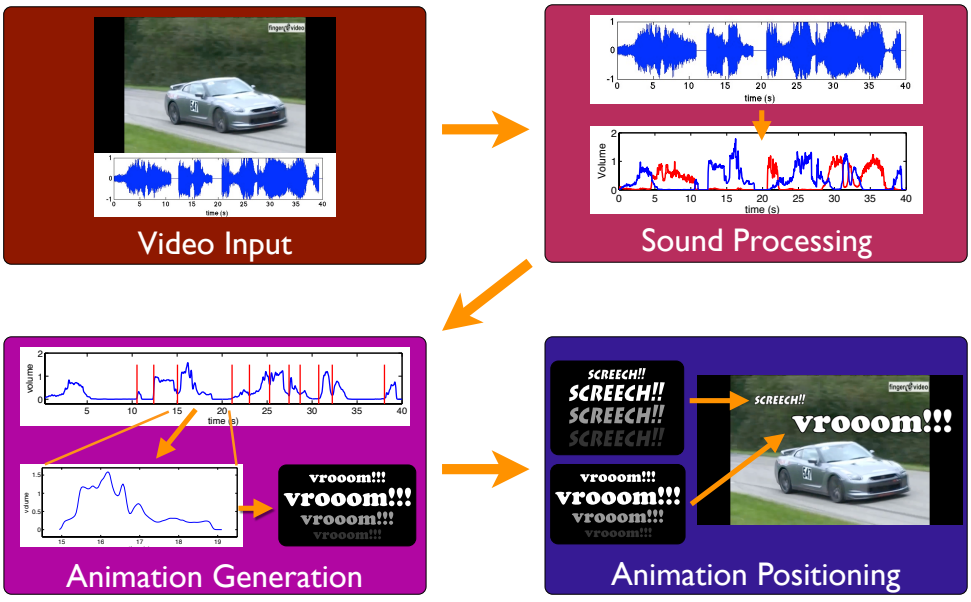

Figure 4.1: An overview of the proposed algorithm. The processing flow consists of three parts: 1) Sound Processing, 2) Animation Generation, and 3) Animation Positioning.

category to generate the sound word animation items. Finally, the generated animation items are positioned in the video considering the position of the sound source object in the video frames.

4.2 Sound Processing

The goal of the sound processing part is to compute a time-series sound volume, or volume envelope, for each sound category. This is achieved by calculating the time-series posterior probability of classification for each sound category, and multiply the result with volume envelope of the original input sound (Figure 4.2). A time-series posterior probability represents the probability of the sound to be labeled as each sound category throughout the timeline of the input sound. Although the posterior probability does not exactly stand for the mixture ratio of sound in each sound category, we found that this method could be a good approximation for the purpose of this research. More complex algorithms such as Marching Pursuit [39] or Non-Negative Matrix Factorization [67] could be applied to improve the result.

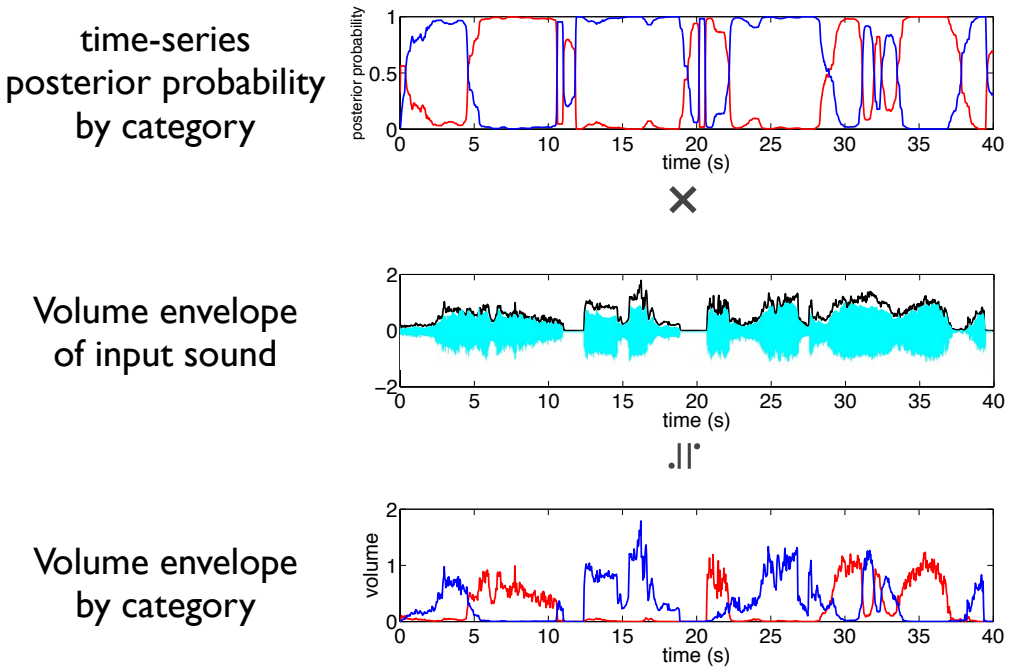

Figure 4.2: The volume envelope for each sound category is calculated by multiplying time-series posterior probability of classification by the volume envelope of the input sound.

4.2.1 Sound Identification

The process of sound identification consists of three steps (Figre 4.4). First, the algorithm divides the input sound into very short segments by sliding a fixed-size cropping window over the sound waveform. We set the window size to be 16512 data points and the sliding size to be 768 data points, which is equivalent to approximately 0.749 and 0.0348 seconds in temporal axis (Figre 4.4-a). Second, for each segment we transform the waveform to a feature vector with 127 dimensions (Figure 4.4-b). Third, we use a support vector machine (SVM) to compute

the posterior probability of the feature vector being classified to each sound category, e.g. "Engine Sound" or "Squeal Sound" (Figre 4.4-c). We classify all the feature vectors generated along with the timeline of input sound to generate a time-series data of posterior probabilities. Since the resultant time-series data are rough and does not relect actual human perception, as final process the algorithm smooth the data with a median filter to produce final results. We chose a median filter for smoothing because it removes noise while preserving "edges" of the time-series data. In following sub-subsections, we will describe the detail of SVM classifier, learning data sets we used to construct the classifier, and how to produce a feature vector from the sound segments.

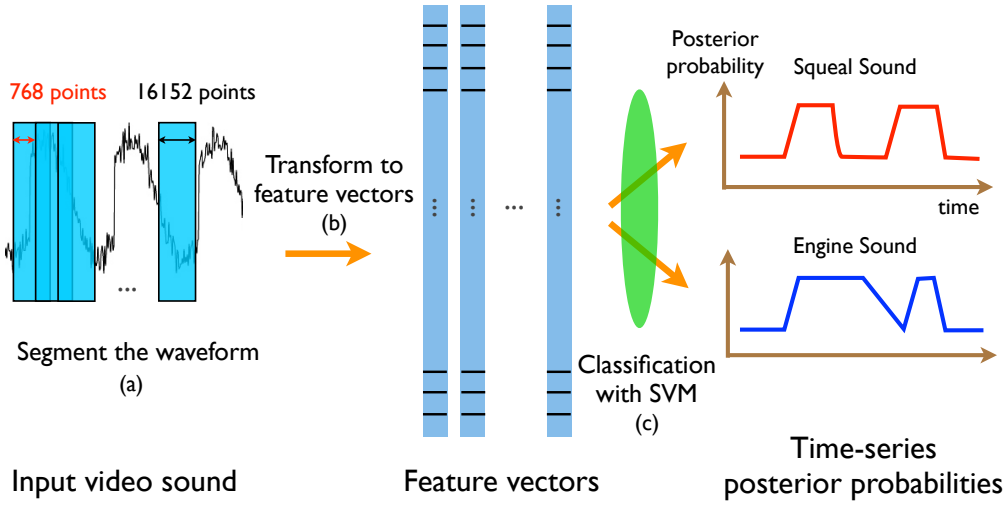

Figure 4.3: Three steps for sound identification. a) Segment the input waveform with a sliding window. b) Transform each segment into a feature vector. c) Classify the feature vectors and calculate the time-series posterior probabilities.

SVM Classifier

A classifier is defined as a program that takes a real vector as input and outputs a label it is to be classified. The input real vector is also called feature vector, and is generated from the data to be classified. A classifier is usually constructed by a supervised machine-learning approach of learning a large number of pairs of training data sets and their ground-truth labels. Support vector machine (SVM) is currently one of the most popular classifiers. One of the advantages of SVM is that it can achieve higher performance even when the learning data sets is smaller [34]. A number of implementations of SVM are published as open-source libraries, which is beneficial for application prototyping.

The original SVM proposed by Boser et al. [14] does not have functionality to calculate posterior probability, i.e. the probability of a feature vector to be classified to each label. However, there are some studies to add the functionality [81], and these methods are implemented in some SVM libraries such as LIBSVM [18]. Our implementation adopted LIBSVM, and trained the SVM with sound data that belong to each sound categories to be identified. Note that the source of training data is not the input video itself and multiple videos could be classified with one pre-trained classifier.

Learning Data Sets

We prepared the learning data sets for the classifier from Freesound.org [5]. We gathered 9 different squeal sound file and 13 engine sound files. The length of the sound in total are respectively 16.56 seconds for squeal sound and 57.08 seconds for engine sound. Similar as identification process described above, we divide each learning sound data into very short segments by sliding a cropping window with 16512 data points. We change the sliding size of the window depending on the duration of each sound to generate 100 feature vectors for each sound file. This will prevent generating a large number of similar feature vectors, which would do harm to classification performance.

Feature Vector Design

The precision of classification results does not depend only on the performance of a classifier, but rather on the design of feature vector generated from the input data. Many feature vectors for sound data are proposed in previous research, and many of them are based on short time Fourier transform or wavelet transform. A short time Fourier transform is an operation to slide a fixed-size cropping window over the sound waveform and generate the Fourier transform for each short wave segment. Lining up the absolute values of resultant vector of Fourier transform generates an image called a spectrogram (Figure 4.4-b). A spectrogram is an image that represents how the volume of sound in each frequency band changes along with the time. Since the original Fourier transform assume the sound to be repeated infinitely, the cropped sound segment is usually multiplied by a window function to simulate a repeating sound.

The design of feature vector in our algorithm is as follows. First, we generate a spectrogram for the input sound. We set the window size to be 256 data points and the sliding size to be 128 data points, which is equivalent to approximately 0.0116 and 0.00580 seconds in temporal axis. Since the length of input sound is fixed to 16512 data points, the generated spectrogram can be represented as a 128x128 pixel image (Figure 4.4-a). Second, we apply a 2-D Fourier transformation to the generated spectrogram. In the resultant image of 2D Fourier transformation, pixel values that is close to the horizontal axis represents how the sound changes with time, and that is close to vertical axis represents how the sound changes with frequency (Figure 4.4-b). Since most of the information is concentrated on these two axis, we pick the values on these axis and concatenate them to generate a final feature vector (Figure 4.4-c).

4.2.2 Volume Estimation

We estimate the sound volume of each sound category by multiplying the volume of input sound with time-series posterior probability for each sound category. Similar as sound identification, we estimate the volume envelope of input sound by sliding a fixed-size cropping window over the sound waveform and compute the Root Mean Square (RMS) of the cropped segments. The definition of RMS is as follows:

$$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{n} x_i^2} \tag{4.1}$$

where N represents the size of the window and x_i represents ith data point inside the window. We set $N = \lfloor 22050/60 \rfloor = 367$ in our implementation to prevent the estimation result to be either too rough or too smooth. We set the sliding

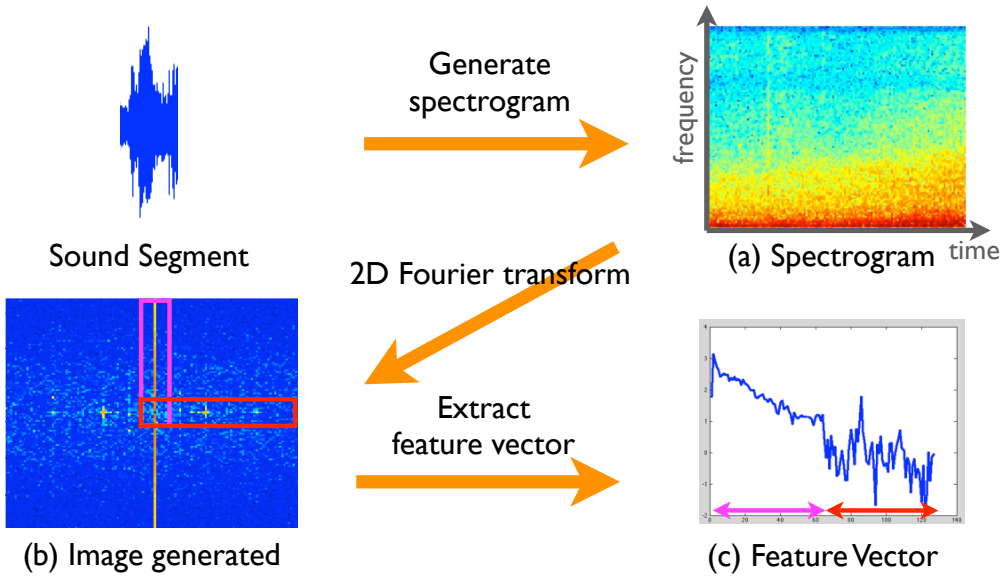

Figure 4.4: Algorithm for feature vector transform.

size of the window to 1 for calculation to keep the resolution of the data and downsample the resultant volume envelope for multiplication with time-series posterior probability data.

4.3 Animation Generation

4.3.1 Volume Envelope Segmentation

The animation generation part analyzes the volume envelope of sounds for each category to generate sound word animation. As described in section 3.2, our design principle first requires to divide the whole volume envelope of the sound into multiple segments. This is achieved based on the method provided by et al. The overview of their method is as follows [42]:

1. Calculate peaks of volume envelope and construct a peak set.
2. Calculate the ratio of two adjacent peaks in the peak set.
3. If the ratio outweighs the threshold, the volume envelope is to be divided at the abyss between the two peaks. Otherwise the smaller peak is to be excluded from the peak set.
4. Go back to 2 if the peak set is not empty.

Figure 4.5 shows an example of segmentation result. We found the result generated by their method tends to be over-segmented for animation generation. In particular, some generated segments are too short to construct an animation item (Figure 4.6-a). Therefore, we modify the constraint "If the ratio outweighs the threshold" in step 2 above to "If the ratio outweighs the threshold th and the distance of the two peaks is larger than tw", to assure each segment has a reasonable length (Figure 4.6-b). We set $th = 1.5$ and $tw = 20$ in our implementation.

We resample the volume envelope to 30 data points per seconds before segmentation for convenience of later process. Since the volume envelope generated through sound processing part is rough and inappropriate for segmentation, we

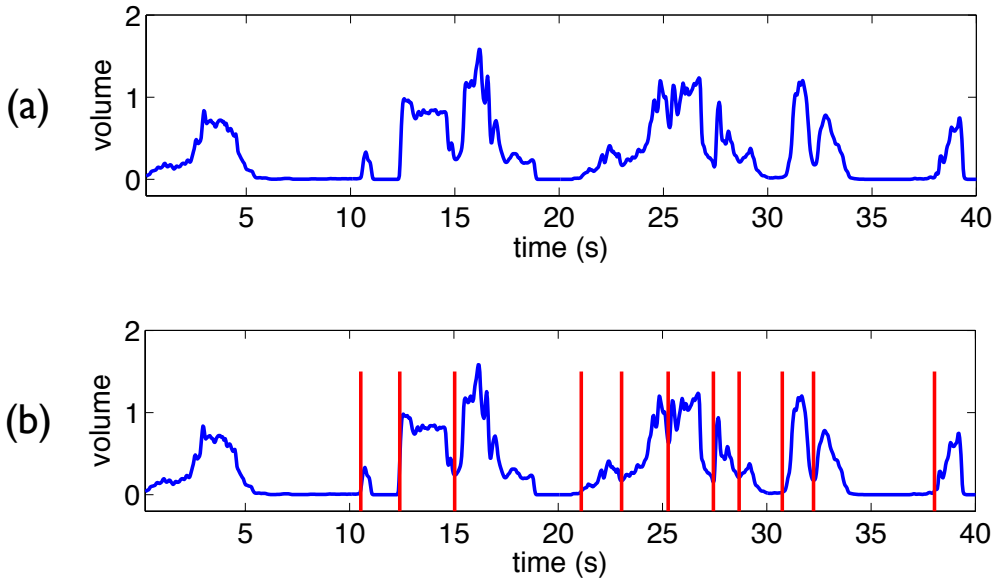

Figure 4.5: Example result of volume envelope segmentation. (a) Original volume envelope (b) Segmentation result

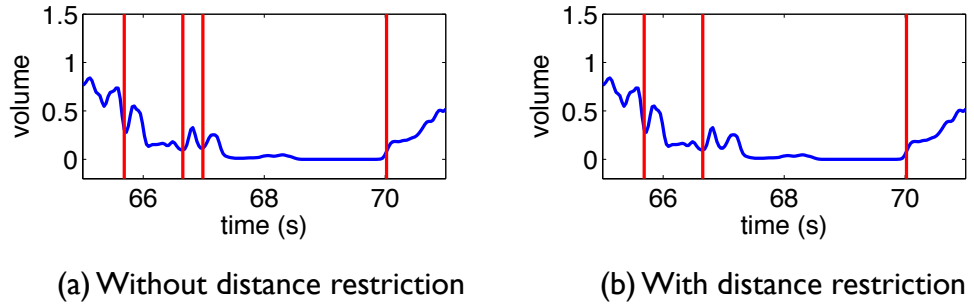

Figure 4.6: Effect of adding minimum distance restriction between two peaks.

apply a median filter before segmentation to smooth the envelope while preserving "edges". We set the window size of the filter to 7 in our implementation. Note that the median filter is only applied for deciding the segmentation point and we generate segmented volume envelope for the data before filtering.

4.3.2 Generating Animation Item

As the second step of animation generation, we generate an animation item from each segmented volume envelope. Following our design principle, we change the form of animation in intensification phase and attenuation phase. Therefore, we first resegment each segmented volume envelope by following algorithm:

1. Find the position $(t_{Pl}, v(t_{Pl})$ of highest peak in the segment (Figure 4.7-a).
2. Find the rightmost peak $(t_{Pr}, v(t_{Pr})$ in the timeline that is higher than $\lambda \cdot v(t_{Pl})$ (Figure 4.7-b,c).
3. Divide the segment into three part by t_{Pl} and t_{Pr} (Figure 4.7-d).

where v(t) represents the volume envelope of the segment on time-volume plane (Figure 4.7-a). λ is a parameter within $[0, 1)$ and we set $\lambda = 0.8$ in our implementation. We respectively name the left, middle, and right part of the segment as "head", "body", and "tail". The body may not exist in some case since t_{Pl} could be identical to t_{Pr}. Intuitively, the head corresponds to intesification phase of volume envelope and the tail to attenuation phase. It is a design problem how to generate animation for the body, and we decided to apply same form of animation as head because some segmented volume envelopes has a long body part where we want to make the animation visible. Therefore, we divide the segmented volume envelope at t_{Pr} to define intensification phase and attenuation phase.

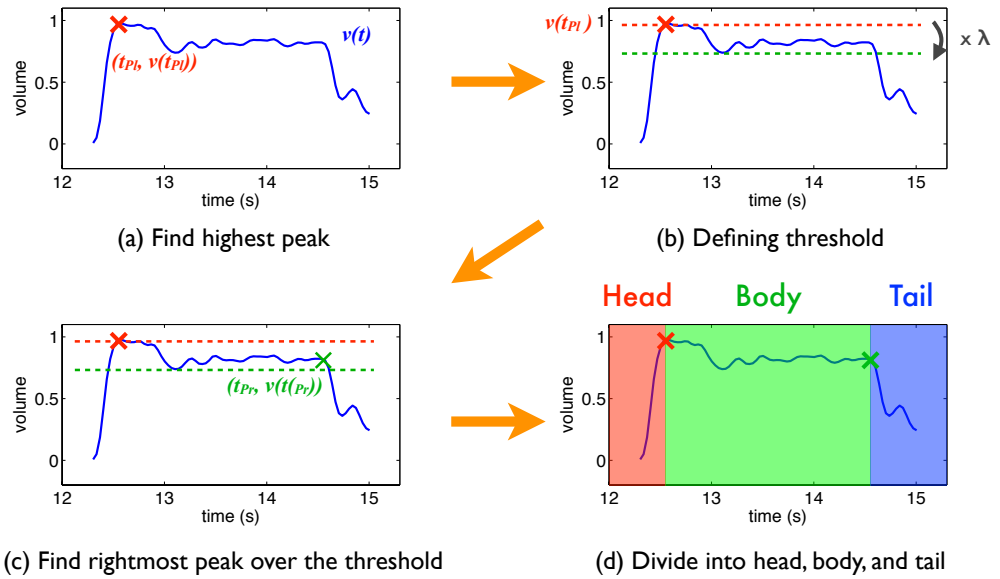

Figure 4.7: Algorithm for resegment volume envelope to intensification and attenuation phase. We define head and body parts as intensification phase, and tail part as attenuation phase.

Insentification phase

We first apply a Gaussian filter for the head and body. Contrary to volume envelope segmentation, we decide not to use a median filter because we want to smooth edges to prevent a large sound word suddenly appears. We set the window size of Gaussian filter to 7 and the standard deviation $\sigma = 1.2$. Second, we simply map the smoothed volume envelope to font size, and fix the opacity to a 100% to generate an "expanding" animation (Figure 4.8 - 1, 2).

Attenuation phase

We apply different animation for attenuation phase depending on its duration. First, we approximate the attenuation curve in the tail part by a Gaussian function (Figure 4.9). The definition of the curve is as follows:

$$V = v(t_{Pr}) \cdot exp(-k \cdot (t - v(t_{Pr})^2) \tag{4.2}$$

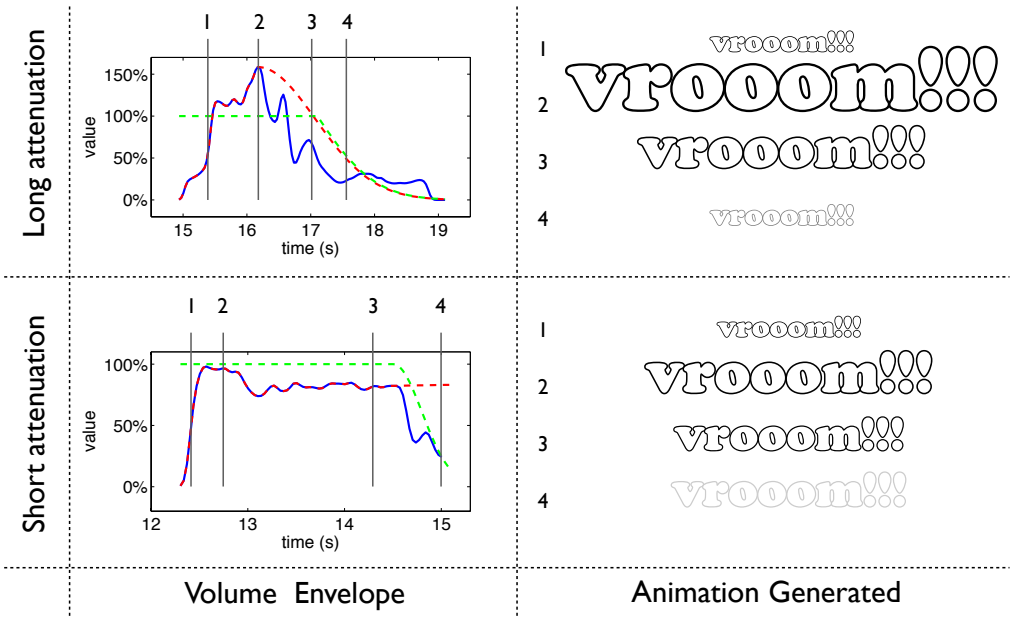

Figure 4.8: Animation generated for the segment with long / short attenuation phase. The blue line shows the original volume envelope, the red show size parameters, and green show opacity parameters. The generated animation frames correspond to each time point 1, 2, 3, 4 are shown on the right half.

Approximation by the curve defined above could be solved as a linear regression problem through following transformation:

$$
\begin{aligned}
V &= v(t_{Pr}) \cdot exp(-k \cdot (t - v(t_{Pr})^2) && (4.3)\\
\ln(V) &= ln(v(t_{Pr})) - k \cdot (t - t_{Pr})^2 && (4.4)\\
ln(v(t_{Pr})) - \ln(V) &= k \cdot (t - t_{Pr})^2 && (4.5)\\
W(V) &= k \cdot T(t) && (4.6)
\end{aligned}
$$

where $W(V) := ln(v(t_{Pr})) - \ln(V)$ and $T(t) := k \cdot (t - t_{Pr})^2$. Assume we have an envelope of the tail part where data $v_1, ..., v_n$ corresponds to time $t_1, ..., t_n$ ($1...n$ are frame numbers). The estimated value of k is as follows:

$$k = \frac{\sum_{i=1}^{n} V(v_i) \cdot T(t_i)}{\sum_{i=1}^{n} T(t_i) \cdot T(t_i)} \qquad (4.7)$$

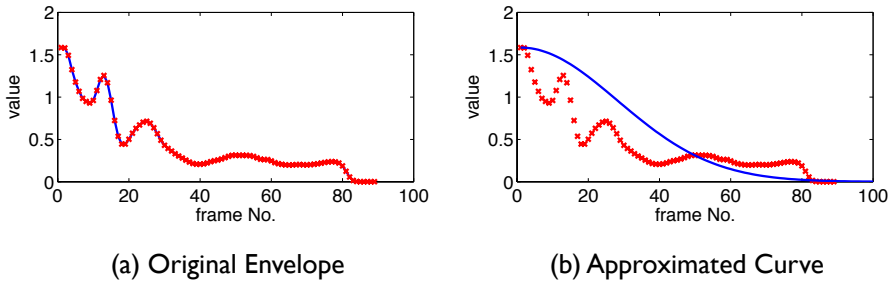

Figure 4.9: Approximation of the tail part by a Gaussian function.

We categorize the attenuation phase to be "long" when the length of tail part is larger than a threshold τ. We set $\tau = 45$ in our implementation, which is 1.5 seconds in time. For long attenuation phase, we vertically adjust the size of approximated Gaussian curve and map it to the font size. We also horizontally shrink the Gaussian curve to 2/3 of tail length, and map the curve to the opacity of the font after keeping 100% opacity for 1/3 of tail length (Figure 4.8-top). These mappings generate a "slowly shrinking & fading-out" animation effect. For short attenuation phase, we first approximate the gradient of the end of the head part. This is achieved by calculating $\frac{\Delta v}{\Delta t}$, where $\Delta v = v(t_{Pr}) - v(t_{Pr} - \Delta t)$. We therefore linearly extend the body (or head $t_{Pl} = t_{Pr}$) part to the end of the tail part using a line with slope $\alpha \cdot \frac{\Delta v}{\Delta t}$ (Figure 4.8-bottom). We set $\alpha = 0.525$ in our implementation. We map the extended line to the font size to generate an animation. We map the approximated Gaussian curve directly to the opacity of the font. These mappings generate a "Rapid fading-out" animation effect. The example result of generated animations is shown in figure 4.8.

4.4 Animation Positioning

The final part of our algorithm positions the generated animation items to the input video. As we described in section 3.2, there are three types of positioning method; static, dynamic-without-movement, and dynamic-with-movement. For the static positioning, we respectively pose the generated animation items on the top and bottom of the video. We pose "Squeal Sound" at the top and "Engine Sound" at the bottom for car videos. For dynamic positioning, we designed a greedy, iterative algorithm as described in figure 4.10. First, we design 3-dimensional video cost fields of the input video depending on the position of the sound source object (Figure 4.10-a). Second, we define 3-dimensional animation cost fields for each animation items (Figure 4.10-b). Therefore, we iteratively determine the position for each animation item by solving an optimization problem (Figure 4.10-c) to minimize the inner product of these two cost fields. After

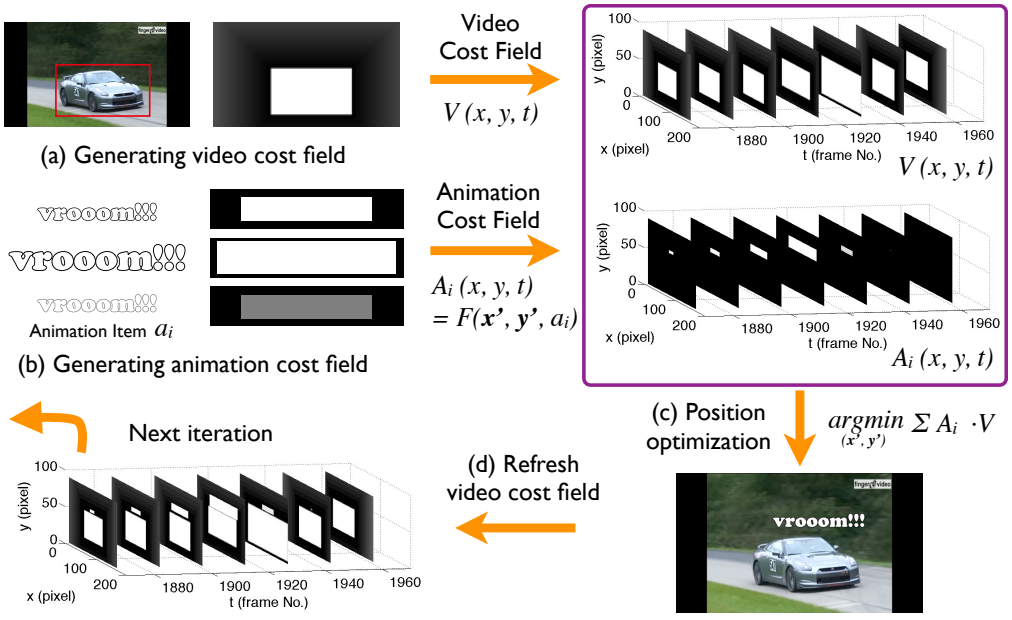

Figure 4.10: Algorithm overview for animation positioning.

positioning each animation items, the video cost function will be updated depend on the positioning result (Figure 4.10-d) to avoid overlapping placement of animation items. We describe the detail of each step in following subsections. We update the video cost field once the final position of the animation item is determined to avoid multiple animation items to overlap each other.

4.4.1 Video Cost Field

Generally, a video could be visualized as a 3D image in a discrete pixel-time space, as shown in figure 4.11. The t axis in figure 4.11 corresponds to the frame number, x and y-axis correspond to the position of pixels in each frame. Similarly, we define the cost field $cost = V(x, y, t)$ as a scalar field in the same 3D pixel-time space (Figure 4.12-d).

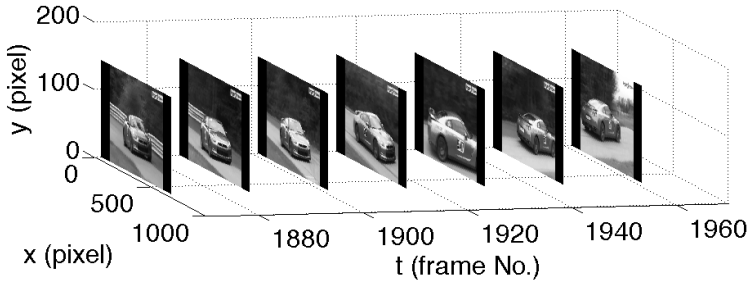

Figure 4.11: Video visualized in 3D pixel-time space.

The cost field is generated by processing 2D cost image for each video frame, and line those image up by the frame number in the pixel-time space (Figure 4.12). The procedure to process the 2D cost image is as follows. First, we detect the bounding box of the sound source object for the input frame image with the Deformable Part Model algorithm [30] (Figure 4.12-a). Second, we generate a low resolution, monotone image that represents the position and size of the detected bounding box in the input frame. All the pixels inside the bounding box is fillead with 1.0 (white pixel), and other pixels as 0.0 (black pixel) (Figure 4.12-b). We set the resolution as 160x90 pixels in our implementation. Third, for every black pixel b_i we caculate its distance $D(b_i)$ to the nearest white pixel. Chessboard distance is used for the calculation instead of Euclidean distance, and the distance is normalized to [0, 1] by the size of the image. Finally, we fill the black pixel with value $d \cdot D(b_i)$ to generate the 2D cost image (Figure 4.12-c). For all x, y, t that exceeds the size of video cost image is set to 10.0 as a penalty. Intuitively, the cost image is designed to avoid the animation item to overlap the sound source object, to run off the boundary of video picture, or to be positioned too far from the object. We found that chessboard distance is more appropriate for this design than Euclidean distance, because with Euclidean distance the algorithm tends to prioritize the area on the top and bottom of the bounding box. Note that we do not distinguish multiple object of same category when it appears at the same time, because we found that in many cases it is difficult to clarify them even with sound. This could possibly be solved by sound source identification algorithms when multi-channel sound is available with video.

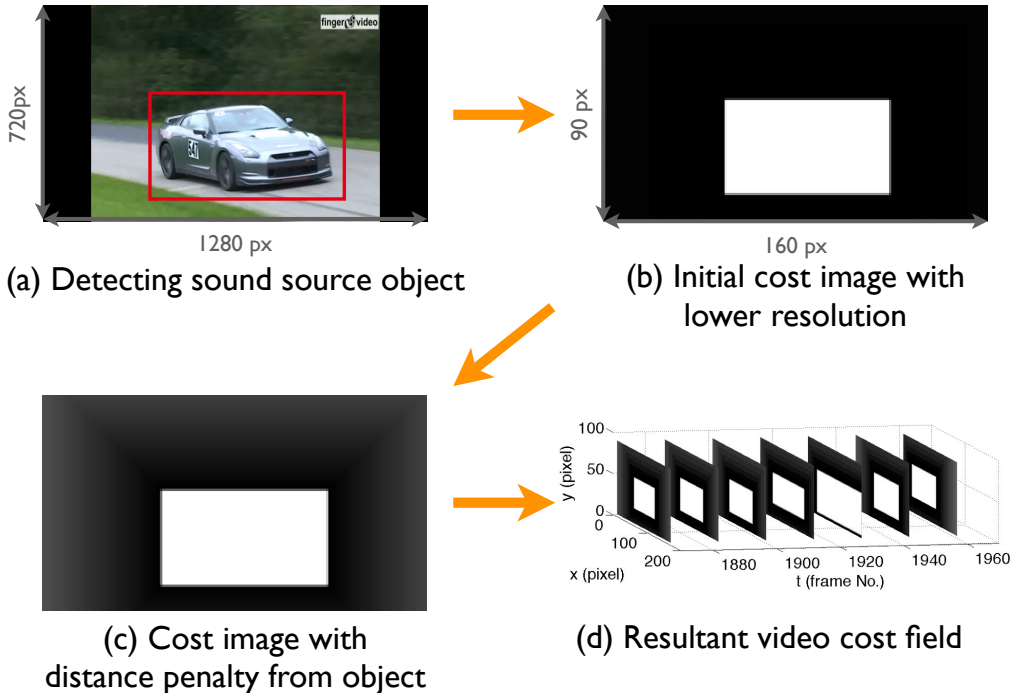

Figure 4.12: Algorithm for generating 2D video cost image. Lining up the resultant images by time generates a 3D video cost field.

4.4.2 Animation Cost Field

The animation cost field is defined by its position in the video. Assume the position of an animation item a_i is given as $\boldsymbol{x'} = x'(t_s), ..., x'(t_e), \boldsymbol{y'} = y'(t_s), ..., y'(t_e)$. $\boldsymbol{t} = t_s, ..., t_e$ are frame numbers of the animation item, which is already defined while generating the animation item. We therefore define a functor F that returns the animation cost field A_i by taking $\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{y}(\boldsymbol{t})$ as input.

$$A_i(x, y, t) = F(\boldsymbol{x'}, \boldsymbol{y'}, a_i) \tag{4.8}$$

Similar as video cost field, we line up 2D animation cost image by frame number to generate the cost field. Assume we are going to generate an animation cost image for frame number t_n. First, we generate a black image that has the same resolution as video cost image (160x90 pixels in our implementation) (Figure 4.13-a). Second, we resize the animation image at frame t_n to according to the ratio of input video size and the size of video cost image, and pose the resized animation image according to the given position $x(t_n), y(t_n)$ (Figure 4.13-b). Third, we generate a bounding box for the posed animation image (Figure 4.13-c). Finally, we fill a value between $[0, 1]$ to the image pixels within the bounding box, based on the opacity of the animation image (4.13-d). Note that the whole process described above can be represented as functor F (Figure 4.14). Figure 4.14-c shows an example of the resultant cost field of an animation item.

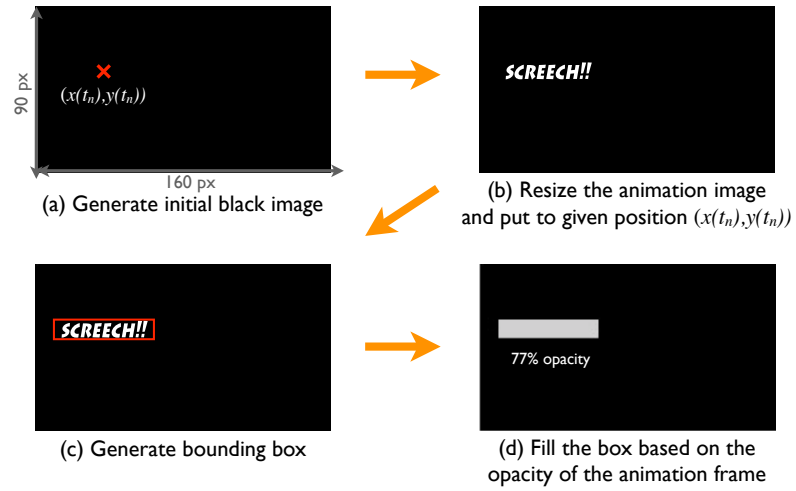

Figure 4.13: Algorithm for generating a 2D animation cost image.

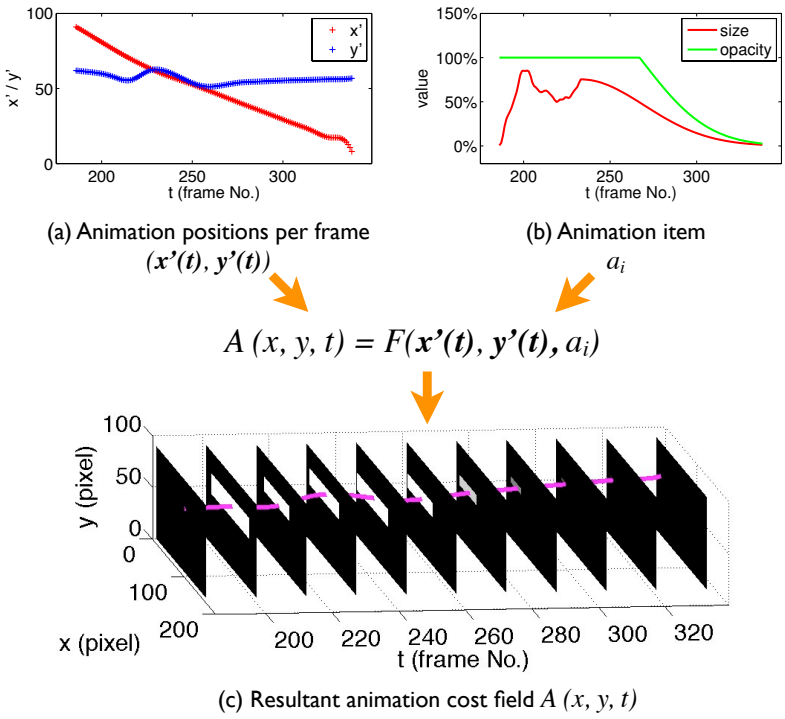

Figure 4.14: The whole process of animation cost field generation can be represented as a functor $F(\boldsymbol{x'}, \boldsymbol{y'}, a_i)$, which takes the time-series position of the animation item as variable. The purple line in (c) shows the time-series position of animation described in (a).

4.4.3 Position Optimization

We adopt an iterative, greedy algorithm to determine the position of the generated animation item one by one. First, we sort the animation items in descending order of value c calculated as follows.

$$c = \sum_{x,y,t} F(\boldsymbol{x'}, \boldsymbol{y'}, a_i) \tag{4.9}$$

Note that any arbitrary $\boldsymbol{x'}$ and $\boldsymbol{y'}$ generates same o value, because it only depends on the size, opacity, and duration of animation item. Intuitively, we want to prioritize processing of the animation items of which these parameters are larger.

Second, we position each animation item $\boldsymbol{p} = (\boldsymbol{x'}, \boldsymbol{y'})$ following the order calculated above. This is basically processed by optimizing following function:

$$C(\boldsymbol{p}) = \sum_{x,y,t} F(\boldsymbol{p}, a_i) \cdot V(x, y, t) + k \cdot S \tag{4.10}$$

Equation 4.10 means that we set the cost as inner product of animation cost field $A_i(x, y, t) = F(\boldsymbol{p}, a_i)$ and video cost field $V(x, y, t)$ (Figure 4.10-c). S is a smoothing factor and k is its coefficient to avoid resultant $\boldsymbol{p}$ to be too rough. Directly optimizing $C(\boldsymbol{p})$ may be extremely inefficient since the variable $\boldsymbol{p}$ may have over 100 dimensions (Figure 4.14-a). Furthermore, the resultant $\boldsymbol{p}$ would be too rough for positioning. Therefore, we define $\boldsymbol{p}$ as a parametric function of p' i.e. $\boldsymbol{p}(\boldsymbol{p'})$, where $\boldsymbol{p'}$ is newly introduced parameter that has lower number of dimensions than $\boldsymbol{p}$. In following sub-subsections we describe how we define $\boldsymbol{p}(\boldsymbol{p'})$ and S and conduct optimization for dynamic positioning with and without movement, respectively.

Without-movement Positioning

The without-movement positioning defines the same position for all the animation frames. Therefore, the $\boldsymbol{p}(\boldsymbol{p'})$ is defined as follows:

$$\boldsymbol{p'} = (X, Y) \tag{4.11}$$

$$\boldsymbol{p}(\boldsymbol{p'}) = (\boldsymbol{x'}, \boldsymbol{y'}) = (X, X, ..., X, Y, Y, ..., Y) \tag{4.12}$$

$$S = 0 \tag{4.13}$$

where $\boldsymbol{p'} = (X, Y)$ represents the static position of the animation. An example value of $\boldsymbol{p'} = (X, Y)$ and its corresponding $\boldsymbol{p}(\boldsymbol{p'}) = (\boldsymbol{x'}, \boldsymbol{y'})$ is shown in figure 4.15. Since the number of dimensions of $\boldsymbol{p'}$ is only two, the optimization problem can be solved by searching all possible (X, Y) within the resolution of video cost images.

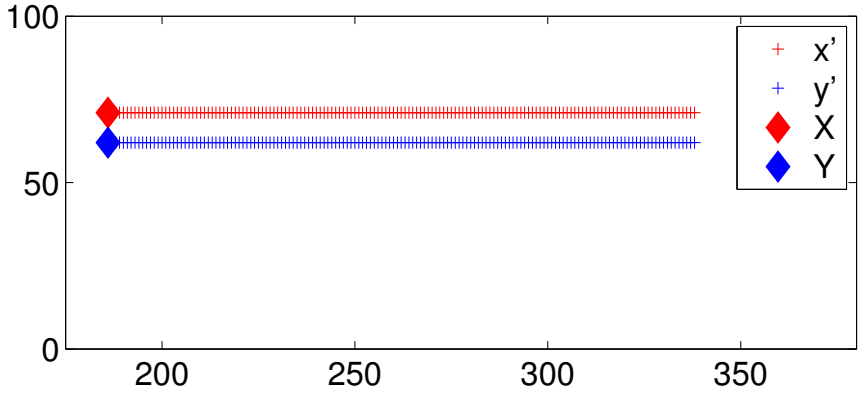

Figure 4.15: An example value of $\boldsymbol{p'} = (X, Y)$ and its corresponding $\boldsymbol{p}(\boldsymbol{p'}) = (\boldsymbol{x'}, \boldsymbol{y'})$. The parameter $\boldsymbol{p}$ with a large number of dimensions has successfully reduced to $\boldsymbol{p'}$ with only two dimensions.

With-movement Positioning

We reduce the dimension of $\boldsymbol{p}$ by approximating the polyline it represents with a spline curve for dynamic positioning. Assuming the spline curve has control points $(X_1, Y_1), ..., (X_n, Y_n)$ and represents a polyline $(x'(t_s), y'(t_s), ..., (x'(t_e), y'(t_e))$, the $\boldsymbol{p}(\boldsymbol{p'})$ is defined as follows:

$$\begin{aligned} \boldsymbol{p'} &= (X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_n) && (4.14) \\ \boldsymbol{p}(\boldsymbol{p'}) &= (\boldsymbol{x'}, \boldsymbol{y'}) = (x'(t_s), ..., x'(t_e), y'(t_s), ..., y'(t_e)) && (4.15) \\ S &= S(\boldsymbol{p'}) && (4.16) \\ &= \sum_{i=2}^{n-1} (|0.5 \cdot (X'_{i-1} + X'_{i+1}) - X'_i| + |0.5 \cdot (Y'_{i-1} + Y'_{i+1}) - Y'_i|) && (4.17) \end{aligned}$$

An example value of $\boldsymbol{p'} = (X_1, ..., Y_n)$ and its corresponding $\boldsymbol{p}(\boldsymbol{p'}) = (\boldsymbol{x'}, \boldsymbol{y'})$ is shown in figure 4.16. We take the control points for every 15 frames, including $\boldsymbol{p}(t_s)$ and $\boldsymbol{p}(t_e)$. This operation reduces the number of dimensions of $\boldsymbol{p'}$ to around 10 to 30. The smoothing factor S contributes to keep the gradient of the spline curve stable. We adopt BFGS quasi-newton algorithm [66] implemented in MATLAB [6] for optimization. We transform the discrete variables and cost field to continuous space using spline interpolation to apply the algorithm, and discretize the resultant $\boldsymbol{p}$ to determine final pixel-wise position.

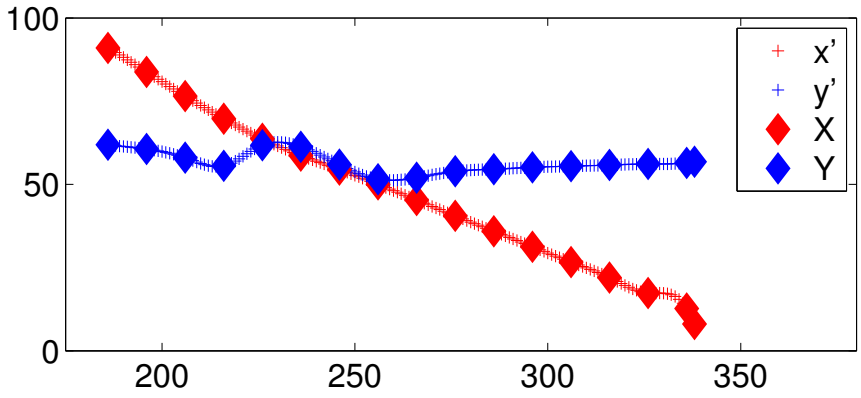

Figure 4.16: An example values of $\boldsymbol{p'} = (X, Y)$ and its corresponding $\boldsymbol{p}(\boldsymbol{p'}) = (\boldsymbol{x'}, \boldsymbol{y'})$. The parameter $\boldsymbol{p}$ with over 300 dimensions has successfully reduced to $\boldsymbol{p'}$ with 34 dimensions.

One problem with quasi-newton algorithm is how to determine the initial value of $\boldsymbol{p'}$ to start optimization. Our idea is to exploit the without-movement positioning method to determine initial $\boldsymbol{p'}$. First, we calculate cost $C(\boldsymbol{p}(\boldsymbol{p'}))$ for all pixels with $\boldsymbol{p}$ and $\boldsymbol{p'}$ defined in equation 4.11, 4.12. This produces an image with each pixel (x, y) filled with value $C(\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}))$ (Figure 4.17-a). Second, we find regional minima in the image by comparing each pixel with its 8 adjacent pixels. We mark the pixel as regional minimum when there are no adjacent pixels that has larger value. Filling value 1.0 to all regional minima produces an image as shown in figure 4.17-b. Third, we remove the continuous white area in the image by iteratively apply a Gaussian filter and set "gray" pixels to be black (Figure 4.17-c). In our implementation the window size and sigma value of the Gaussian filter are set to 5 and 1, respectively. Finally, an image with sparse white points is generated (figure 4.17-d). Each white point represents an initial $\boldsymbol{p'}$ value for

optimization, i.e. assuming the position of a white point is at (X_i, Y_i), we set the initial $\boldsymbol{p'}$ as $\boldsymbol{p'} = (X_i, ...X_i, Y_i, ..., Y_i)$. Finally, we optimize $C(\boldsymbol{p}(\boldsymbol{p'}))$ for all these initial values and choose the minimum cost result as the final optimization result.

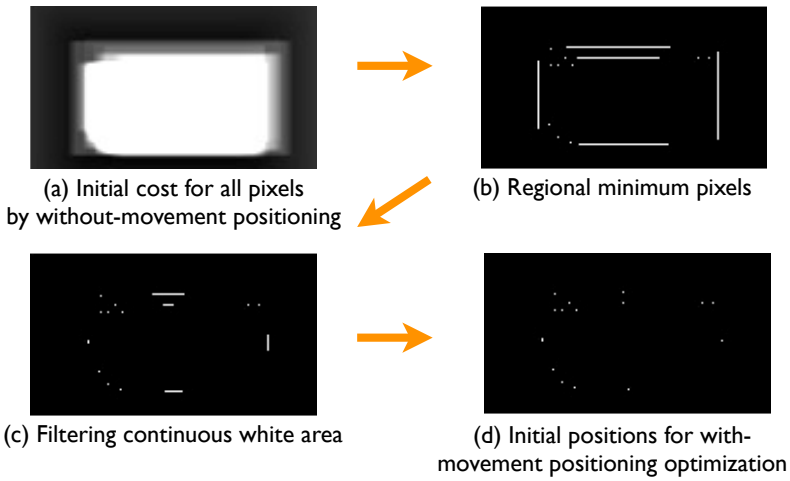

Figure 4.17: Algorithm for determining initial positions in with-movement positioning optimization.

4.4.4 Updating Video Cost Field

We update the video cost field after position of each animation items is determined, in order to avoid multiple animation items to overlap. This is achieved by adding the animation cost field to video cost field, i.e.

$$V_{new}(x, y, t) = V_{old}(x, y, t) + Ai(x, y, t) \tag{4.18}$$

Figure 4.18 shows an example result of updating.

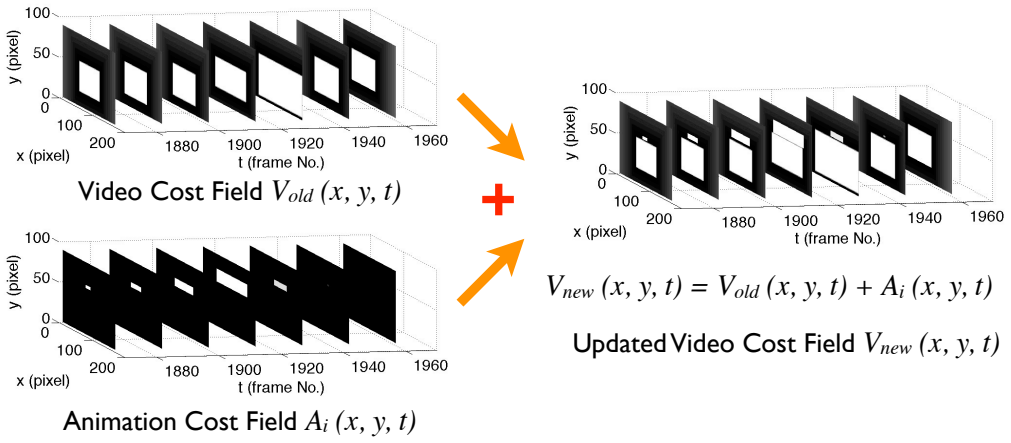

Figure 4.18: The video cost field is updated when an animation item has been inserted into the video. The new video cost function is used for positioning next animation item a_{i+1}.

Chapter 5

Results

In this chapter, we show several video sequences with sound word animation annotated. We take the input video from a car racing video product "Zusammenfassung der Bergrennen 2010" [3]. All these annotations are automatically generated by our system. We also briefly discuss performance of the proposed algorithm at the end of this chapter.

5.1 Resultant Video Sequence

Figure 5.1 shows the resultant video sequence generated by our system. The sound word animations are positioned with "dynamic with movement" style, which is the most complex animation style that our system can generate. The result shows that the category and the dynamics of the sound volume in the video are clearly visualized by the animated sound words. The mixture of sound in different categories is also well visualized by by showing multiple sound words at once.

Figure 5.2 shows a comparison of the generation result with traditional static caption style without any animation. The static caption style is generated by directly mapping the volume of the sounds of each category to the size of the corresponding sound word. The size is set to 0% when the sound volume is below a specific threshold, otherwise 50%. While it is quite easy to feel the dynamics of sound volume with animation, it is nearly impossible to gain the dynamics with traditional static caption style.

Figure 5.3 shows a comparison between three different positioning style. The four rows on the top show the comparison between static positioning and dynamic positioning without Movement, and the bottom four shows the comparison between dynamic positioning with and without movement. The result shows the dynamic positioning method appropriately positions the generated animation near the sound source object. The difference between dynamic positioning with and without movement is smaller, but the animation items successfully follow the movement of the sound source object in with-movement positioning style.

Figure 5.1: Video sequence with sound word animation annotated. Captured every 15 frames of the video with 30 fps. (red&green) Sounds in multiple categories are successfully visualized. The mixture of sound is represented by showing multiple sound words at once. (blue) Shrinking & fading animation for a long attenuation phase.

Figure 5.2: Comparison of (red) animated sound words and (blue) static sound words in traditional caption style. While the static sound words do not provide any dynamics of sound volume, our method with animation successfully provides the dynamics and revealed multiple times of drifting and acceleration on the engine.

Figure 5.3: Comparison of (red) static, (blue) dynamic without movement, and (green) dynamic with movement positioning style. The dynamic positioning method has successfully positioned the generated sound word animation near the sound source object. Dynamic positioning with movement follows the moving sound source by continuously changing the position of animation items.

5.2 Performance

The performance of the proposed algorithm can be roughly divided into three factors: 1) The precision of sound identification, 2) calculation cost for sound processing and animation generation, and 3) calculation cost for positioning.

Currently, the precision of our sound identification method is not high enough to robustly process car videos. In our experiment, the algorithm could only precisely identify around 60 seconds in a 472 seconds of input sound from a car racing video (around 10 to 15%). The larger amount of learning data sets and sophisticated design of feature vector in necessary to improve the performance. However, our research focus is not on sound identification and separation, but on how to apply the identification result for visualization. Considering more sophisticated algorithms for sound identification are widely studied in the field of sound processing, it would not be hard to improve the identification performance by applying those algorithms, provided the amount of learning data sets is enough.

We conducted a simple experiment with a video with 110 seconds to measure the calculation cost of the algorithm. A PC with Intel 1.8GHz core i7 CPU and 4GB memory is used for the experiment. The calculation cost for volume estimation and animation generation was relatively light. The transformation of the input sound to the feature vector for identification took the longest time of 18.34 seconds, while all other process finished in less than 1.5 seconds. The calculation cost will get higher for longer duration of the input sound and more number of sound categories to process. On the other hand, this is not a significant issue because it is quite easy to parallelize the whole process of sound processing and animation generation.

The calculation cost for dynamic positioning is relatively heavier, especially for with-movement style positioning. The heaviest cost lied in visual object detection with Deformable Part Model [30], which took around 2 seconds for processing one video frame. 11.73 seconds were taken to generate initial video cost field with the object detection results. Dynamic positioning of 58 animation items to the video in total took 31.25 seconds for without-movement style and 2576.30 seconds for with-movement. In with-movement optimization, 28.18 seconds were taken for determining initial positions, and 2529.17 seconds were taken for optimization with quansi-Newton method. Again, it is not a significant issue, since it is quite easy to parallelize the whole positioning method. Object detection and video cost field generation can be processed independently for each video frame. Position optimization could also be parallelized by separating the video into different parts by taking the distribution of animation items in the timeline into consideration. Further, the proposed method does not require the high-speed calculation in its nature, since the whole algorithm is processed automatically. What the user has to do is just throw the input video to the system and wait for the result.

Chapter 6

User Study

We conducted a preliminary user study to evaluate how our automatically-generated sound word animation effects the audience experience. We conducted two types of comparative user study. The first (study A) was designed to figure out how the existence of sound word animations effects the audience experience, while the second (study B) was to figure out how the difference in animation design effects the audience experience. For the first user study, we compared videos with sound word animation and no text captions at all. For the second, we compared videos annotated with different types of sound word animation. Sounds are muted for all these videos to simulate cases when participants have no access to sounds. We recruited over 700 people through online crowdsourcing services. They were shown videos with different types of text captions and asked to answer a questionnaire about the video. In the following sections we describe how we utilize the crowdsourcing service for the user study, detailed design of the questionnaire, and the result of the user study.

6.1 User Study with Crowdsourcing

Crowdsourcing services are being more and more common these days. Several services such Amazon Mechanical Turk [1] allow the user to post simple "microtasks" to the crowd and ask crowdworkers to complete them with specified prices. A number of studies have been conducted on how to exploit the power of crowdworkers as "human processors" to achieve tasks that are difficult for computer [55, 19]. One difficulty in utilizing a crowdsourcing service to have some work done is to control the quality of crowdworkers, since there are a number of malicious crowdworkers who "provide nonsense answers in order to decrease their time spent and thus increase their rate of pay." [44]. Kittur et al. [44] proposed a guideline for utilizing these crowdsourcing services for user studies. They recommended the user study to include have "explicitly verifiable questions" of which the answer could be easily identified for unmalicious crowdworkers. They also recommended the user study to "require as much or less effort to complete in good faith than providing malicious answers". They reported that qualitative question such that "asking users to generate keyword tags for the content" could also be a good way to verify unmalicious workers.

We used Yahoo! crowdsourcing service for the user study [7]. Following the guideline, we introduced two explicitly verifiable questions in our study to exclude malicious answers. We also asked the user to provide qualitative comments and exclude answers with low-quality comments such as nonsense (e.g. "dsdfghjklkjhgfdsasdfghjklkjhgfds"), malicious (e.g. "You are such a fool"), or uninformative comments (e.g. "I don't know", "I can't understand what this

questionnaire is for"). Note that we exclude these comments before we look at their answer for other questions in order to avoid the exclusion process to be biased.

6.2 Questionaire Design

We have designed within-subject study (i.e. the participants see two different types of videos) for both user studies. The procedure is very simple: participants see online questionnaire form and answer all the question items. We put the videos on the top of the form so that the user can repeatedly see the video to answer. A general procedure for this type of user study is to show two kinds of video separately and ask the participants to answer the questionnaire for each video in order (i.e. show video A, answer questionnaire for A, show video B, answer questionnaire for B). Questionnaire with Likert scale is used as qualitative measurements in such cases. However, this procedure may not be suitable for our user study where participants are crowdworkers. The general procedure requires to ask similar questionnaire to the user for multiple times therefore takes longer time. Since the crowdworker basically want to shorten the time for task to increase the rate of pay, longer questionnaire may harm the quality of their answer (e.g. tend not to read the question carefully). Longer questionnaire also increase the amount of effort required to complete the questionnaire in good faith, which is not recommended in the guideline by [44]. One way to solve this problem is to switch the within-subject study to between-subject, where each participant sees only one type of video and answer questionnaire. On the other hand, this disables the user to provide specific comments on the difference between two types of video.

As a preliminary user study, we decided to keep the study as within-subject study to gain comments for future improvement of the system. We therefore decided to show two different types of video at once. Four questions in the first study asked how much the participant "agree/disagree" to a statement with ordinal 7-point Likert scale, while other questions asked which of the two videos is more applicable for a statement. A 7 point scale from 1 to 7 is used for this purpose, and we name this as "7 point A/B scale". This way of qualitative measuring has drawbacks that it only represents the preference of the participant between two videos but not the preference in general. For example, even if the participant thinks that video B is more interesting than video A, it does not mean that the participant actually thinks that video B is an interesting video in general. However, our way of qualitative measuring is still useful in order to gain an initial user feedback to the first prototype of automatically-generated sound word animation. A formal user study with qualified participants should be conducted against a more developed version of sound word animation in the future.

6.2.1 Comparison of Videos With and Without Sound Word Animation

In the first user study, we compare videos with and without sound word animation . We composed two identical car racing video with 46 seconds duration for comparison, and annotated sound word animation to one of the two with our system. We adopt the fixed positioning method, where the generated sound word animations are always positioned to fixed positions in the video. This is the simplest design our system could generate thus could be a good baseline of sound

Figure 6.1: The user see two videos with different types of annotation concurrently. The order of the two videos is randomized among participants. Note that we used Japanese sound words in the user study since most of the participants are Japanese.

word animation for comparison. We designed to show animation corresponding to the engine sound on the bottom, and the squeal (drifting) sound at the top. We therefore vertically positioned the two videos and combined them into one (Figure 6.1). We named the video on the top as A and bottom as B. We randomized the positional order in order to take balance among the participants.

The composition of the questionnaire is as follows. On the first page, we described the purpose of the study and asked several basic questions concerning the gender, age, and how long they watch videos every week. We put the video on the top of the second page, with a short description about difference between two types of videos on the top of the video. Text instructions are also shown to note the participant that he or she can repeatedly play the video and that the video is composed without sound. The video is followed by 9 question items. Table 6.1 shows the detail of these items and their order. At the end of the questionnaire the participants were asked to comment on "how this type of caption could be improved" with at least 30 characters.

No.	Question	Answer Type
1	I can gain the dynamics of sound volume.	7 point A/B scale
2	I can distinguish which object is making sound.	7 point A/B scale
*3	Which is the color of 4th car appeared in the video?	Multiple choice
4	This type of caption is a natural representation of sound.	7 point Likert scale
5	This type of caption makes video watching enjoyable.	7 point Likert scale
6	This type of caption is visually noisy.	7 point Likert scale
7	This type of caption is useful for video without sounds.	7 point Likert scale
*8	When the white car first appeared in the video?	Multiple choice
9	How this type of caption could be improved?	Comment with texts

Table 6.1: Question items for the comparison of videos with and without sound word animation. Note that question number with * is set for quality control.

6.2.2 Comparison of Different Animation Styles

In the second user study, we compare videos with different types of sound word animation annotated. We used the same video as the first user study as input and generated videos with four different types of annotation by our system:

SPC Statically Positioned Caption: The annotation is an static image and position is fixed. Similar as conventional closed caption style.

SPA Statically Positioned Animation: The annotation is animated and the position of animation items are fixed.

DPA Dynamically Positioned Animation without movement: The annotation is animated and the position of animation items changes dynamically. Each animation item does not move its position once appeared.

DPAM Dynamically Positioned Animation with Movement: The annotation is animated and the position of animation items changes dynamically. Each animation item could move its position after it appeared.

The visual difference of these four animation styles is shown in figure 5.2 and 5.3. The generation method of SPC is same as the one used in chapter 5. Similar as the first user study, for SPC and SPA we designed to show animation corresponding to the engine sound on the bottom, and the squeal (drifting) sound at the top.

We compared three pairs of these four types of annotation, 1) SPC vs. SPA, 2) SPA vs. DPA, and 3) DPA vs. DPAM. SPC and SPA are different in whether the sound word is animated or not, SPA and DPA are in dynamically positioned or not, and DPA and DPAM are in the animation items move or not. These comparison pairs allows us to measure how each design factor effects the audience experience and would provide a good future direction to improve the system. The basic composition of the questionnaire is also same as the first user study, except no question items are answered in Likert Scale. Table 6.2 shows the full list of qestion items. At the end of the questionnaire the participants were asked to comment on "Please describe the advantage and disadvantage of caption type A and B, respectively" and "How these types of captions could be improved" with at least 30 characters, respectively.

No	Question	Answer Type
1	I can gain the dynamics of sound volume with this caption.	7 point A/B scale
2	The caption is appropriately positioned.	7 point A/B scale
3	I can distinguish which object is making sound with this caption.	7 point A/B scale
4	This type of caption is a natural representation of sound.	7 point Likert scale
*5	Which is the color of 4th car appeared in the video?	Multiple choice
6	This type of caption makes video watching enjoyable.	7 point Likert scale
7	This type of caption is visually noisy.	7 point Likert scale
8	This type of caption is useful for video without sounds.	7 point Likert scale
*9	When the white car first appeared in the video?	Multiple choice
10	How these types of captions could be improved?	Comment with texts

Table 6.2: Question items for the comparison of different animation styles. Note that question No with * is set for quality control.

6.3 Results

In this section we describe the result of the two user studies, respectively. As we described in section 6.2, the answers gained is filtered to exclude malicious

crowdworkers. Valid answers have cleared all the quality checking questions and provided reasonable comments.

6.3.1 Comparison of Videos With and Without Sound Word Animation

229 answers were gained and 145 of them were filtered as valid (63.3%). Figure 6.2 show the histograms of the answers in percentage and table 6.3 shows the mean and sample standard deviation of answers for each question.

The result shows that the majority of participants thought that the sound word animation is effective for gaining the dynamics of sound volume (Q1) and distinguish the sound source object (Q2). On the other hand, it did not contribute much to making the video enjoyable (Q4). The majority of participants thought that the sound word animation with static positioning was less natural as a representation of sound (Q3), and found it visually noisy (Q5). However, the participants thought that the sound word animation was useful for video without sounds overall (Q6).

We also asked participants to give comments on "How this type of caption could be improved". These comments are summarized in table 6.4. It turned out that more than half of the participants thought that the design of the font or choice of sound words were not appropriate, especially the choice of font type was unwelcome and strongly required revising. It is interesting while some participants thought that the sound word should be smaller to be less noisy, others thought that it should be made larger to have more impact. The sound words were received monotonous since there were only two kinds of word in the video, and some participants thought that the choice of specific word to be unnatural for the sound it visualized. Concerning animation, a better placement to avoid interfering visual and the position of sound source was required, which is one of our research goals. Some participants also thought that the animation should be shown less frequently and limited to important or dynamic scenes. Other suggestions were that the user should be able to switch it on/off depend on his or her tastes and needs. Overall, the participants tended to think that the design of sound word animation, fonts, and the choice of the word should be added more variety to improve the audience experience.

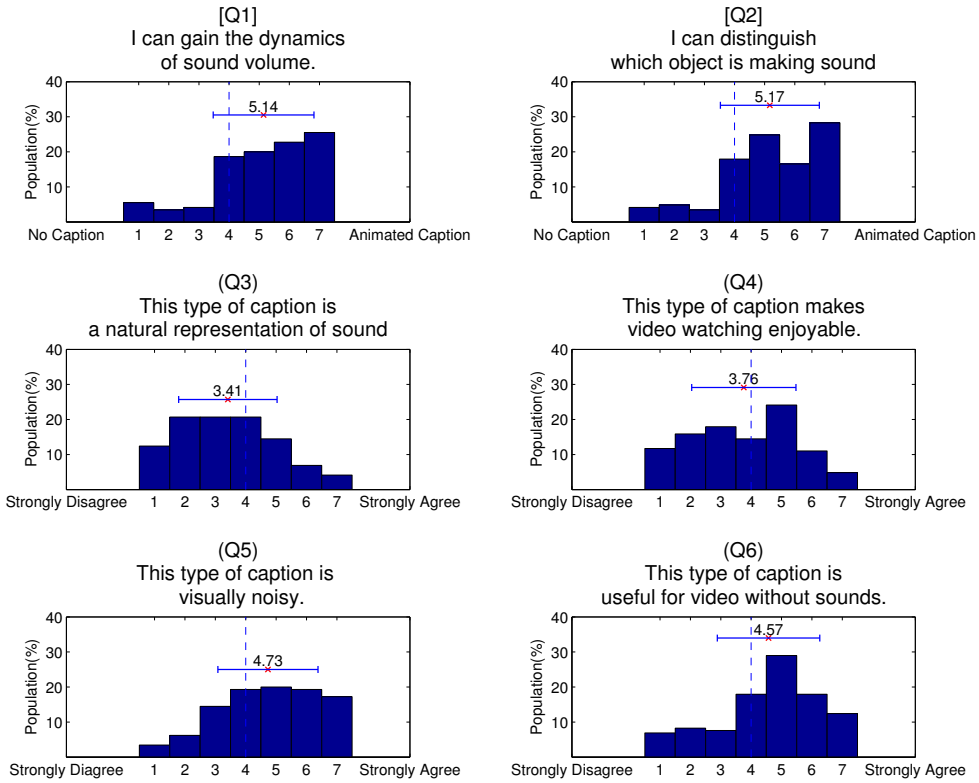

Figure 6.2: Histgram of answers for each question. The red cross shows the mean value and error bar shows the sample standard deviation. Note that while Q3 to Q6 were answered with 7-point Likert scale, Q1 and Q2 were answered with 7-point A/B scale.

No	Question	mean	stdev
Q1	I can gain the dynamics of sound volume.	5.14	1.67
Q2	I can distinguish which object is making sound.	5.17	1.64
Q3	This type of caption is a natural representation of sound.	3.41	1.61
Q4	This type of caption makes video watching enjoyable.	3.75	1.71
Q5	This type of caption is visually noisy.	4.73	1.64
Q6	This type of caption is useful for video without sounds.	4.57	1.69

Table 6.3: The mean and sample standard deviation for each question.

Major category	Minor Category	Major comments
Font(87)	Type(48)	"Hard to read", "Should be improved", "Too comical", "Should be more simple"
	Size(17)	"Too big / should be smaller", "Should be larger"
	Color(13)	"Add more variation", "Should be improved"
	Others(9)	"Add more variation", "Should be improved"
Sound Word(35)	Variation(20)	"Add more Variation"
	Unnatural(9)	"The sound word chosen does not suits the sound"
	Others(6)	"Use English instead of Katakana", "Use shorter words"
Animation(32)	Placement(11)	"Should be fixed either top or bottom, not both", "Should consider the visual information", "Should be placed near sound source"
	Timing(9)	"Should be shown less frequently", "Should be limited to dynamic scenes"
	Movement(3)	"Should not move too frequently", "Slight movement would be enough"
	Others(9)	"Add more variation", "Should be improved"
Others(36)	No need(11)	"Just noisy", "Cannot concentrate on video"
	Others(24)	"Intersting", "Should be able to switch On/Off", "Should add some baloon effect or line effect"

Table 6.4: Summary of comments on "How this type of caption could be improved". The number indicates the number of comments mentioned about the category topic. Note that the comment provided by a single participant could include multiple category.

6.3.2 Comparison of Different Animation Styles

Statically Positioned Caption (SPC) vs. Statically Positioned Animation (SPA)

183 answers were gained and 113 of them were filtered as valid (61.7%). Figure 6.3 show the histograms of the answers in percentage and table 6.5 shows the mean and sample standard deviation of answers for each question.

The result shows a strong tendency of participants to think that Statically Positioned Animation (SPA) provided more dynamics of sound information than Statically Positioned Caption (SPC), of which the size of sound word were fixed (Q1). They also tended to think that SPA is more effective on a distinguishing sound source object (Q3) and made video watching enjoyable. There is less difference in Q2 since the position of both SPA and SPC is basically same. The difference in terms of visual noisiness is also smaller (Q6). Overall, the participants tended to think that SPA is more useful for video without sounds (Q7).

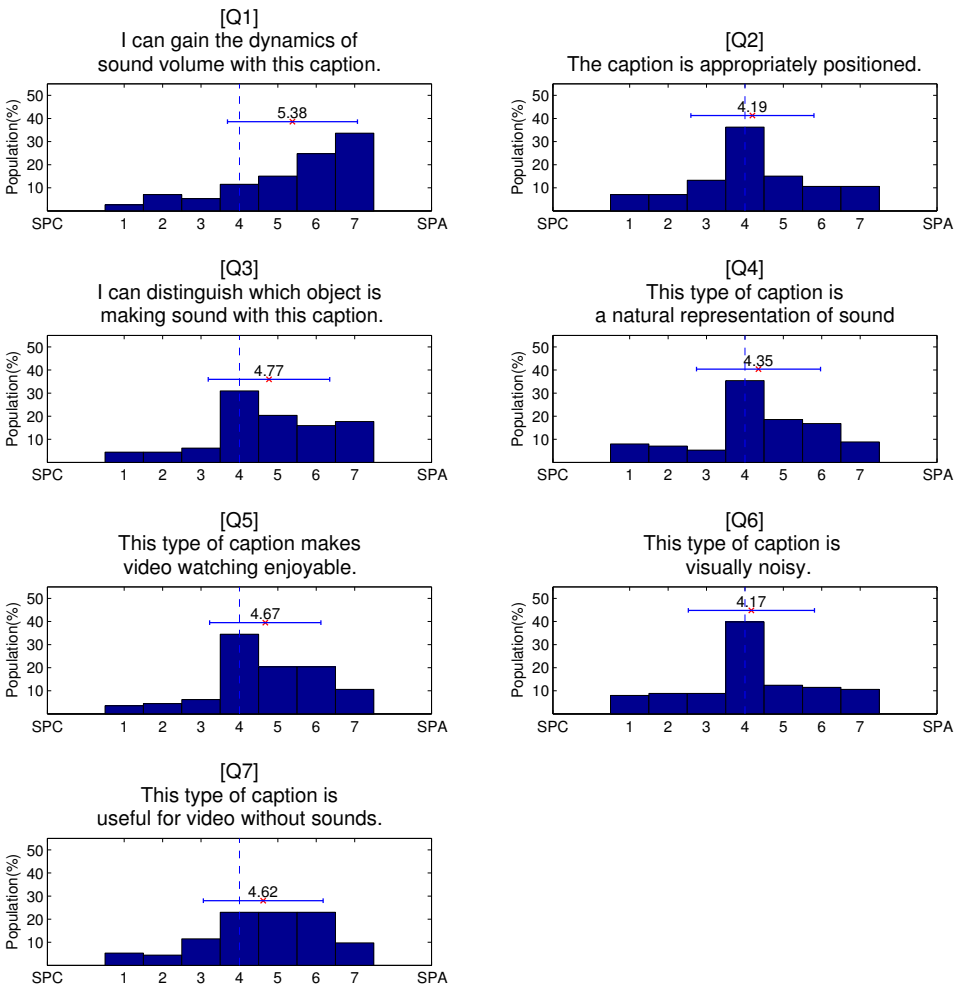

Figure 6.3: Histogram of answers for each question. The red cross shows the mean value and error bar shows the sample standard deviation. All questions were answered with 7-point A/B scale.

No	Question	mean	stdev
Q1	I can gain the dynamics of sound volume with this caption.	5.38	1.69
Q2	The caption is appropriately positioned.	4.19	1.60
Q3	I can distinguish which object is making sound with this caption.	4.77	1.58
Q4	This type of caption is a natural representation of sound.	4.35	1.61
Q5	This type of caption makes video watching enjoyable.	4.67	1.45
Q6	This type of caption is visually noisy.	4.17	1.64
Q7	This type of caption is useful for video without sounds.	4.62	1.56

Table 6.5: The mean and sample standard deviation for each question.

Table 6.6 shows the comments gained on "Please describe the advantage and disadvantage of caption type A and B, respectively" (SSC Positive/Negative & SPA Positive/Negative) and "How these types of captions could be improved" (Suggestive Comments). The comments show that the participants liked SPA in that in having more dynamics and interesting but poorer in readability. The SPC is preferred in that is had higher readability, but was thought to be boring

and provided less dynamics. We found it interesting that in both methods a certain number of participants exist that commented the annotation is noisy. One reason why SPC is percieved noisy was the "blinking effect", which is caused by the caption suddenly appearing and dissapearing in case the sound volume is changind drastically. The smooth animation of SPA will prevent such effect to appear.

We show the comments given on "How these types of captions could be improved" on the bottom row of the table 6.6. Since many comments given are similar in those given in the first, we picked up the comments that seemed to took the difference of two annotation method into consideration. Some comments suggested setting the lower boundary of size of sound word animation, since it was sometimes too small and hard to read. Other major opinions are that the method should be chosen depending on the category of video contents (e.g. movie, drama, news) or the preference of the audience, which is very similar to the comment in the first user study that suggested the audience should be able to switch on/off of the caption.

Major category	**Minor Category**	**Major comments**
SPC Positive	Good Readability(36)	"Easy to read", "Larger words are easier to read"
	Less noisy (15)	"Does not interfere with visual", "Able to focus on car", "Calm and stable"
	Others (10)	"Easy to understand the category of sound", "Do not get feed up"
SPC Negative	Less Dynamics(16)	"Do not feel the dynamics of sound", "Hard to know the sound volue", "Less reality"
	Boring(15)	"Too simple", "No impact on visual", "Got bored"
	Noisy (12)	"Blinks too freaquently", "Too large and interfere with the visual", "Noisy"
	Others(6)	"Difficult to understand the category of sound", "Too formal", "Less effective"
SPA Positive	More Dynamics(72)	"Able to gain how sound volume changes", "I can feel dynamics", "Easier to imagine sounds"
	Interesting (4)	"Quite interesting", "Makes the video enjoyable"
	Others (4)	"The caption is standing out", "More expressive"
SPA Negative	Noisy(32)	"Visually noisy", "Interfere with the visual", "Cannot concentrate on visual", "Less healthy for eye"
	Poor Readability(18)	"Difficult to read", "Cannot read when the word is small"
	Too appealing (6)	"Too appealing", "Too much sound words"
	Others(8)	"Too comical and not serious", "I got bored", "Hard to keep attention on words"
Suggestive Comments	"Avoid being too dynamic", "Avoid the caption to be too small", "Use different caption type depend on the video category", "Enable the audience to choose"	

Table 6.6: Summary of comments on "Please describe the advantage and disadvantage of caption type A and B, respectively" and "How this type of caption could be improved". Similar comments as the first user study on the latter question are not shown here. The number indicates the number of comments mentioned about the category topic. Note that the comment provided by a single participant could include multiple categories.

SPA (Statically Positioned Animation) vs. DPA (Dynamically Positioned Animation without Movement)

147 answers were gained and 95 of them were filtered as valid (64.6%). Figure 6.4 show the histograms of the answers in percentage and table 6.7 shows the mean and sample standard deviation of answers for each question.

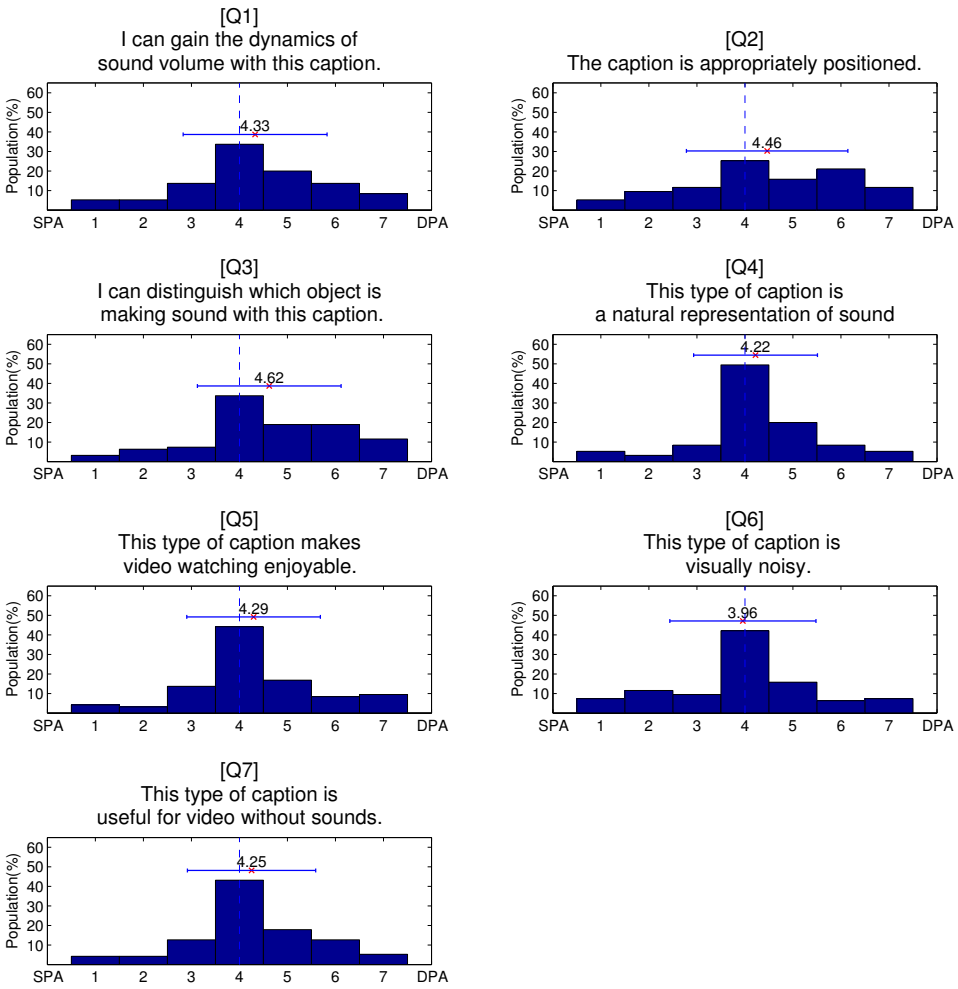

Figure 6.4: Histogram of answers for each question. The red cross shows the mean value and error bar shows the sample standard deviation. All questions were answered with 7-point A/B scale.

No	Question	mean	stdev
Q1	I can gain the dynamics of sound volume with this caption.	4.33	1.50
Q2	The caption is appropriately positioned.	4.46	1.68
Q3	I can distinguish which object is making sound with this caption.	4.62	1.50
Q4	This type of caption is a natural representation of sound.	4.22	1.29
Q5	This type of caption makes video watching enjoyable.	4.29	1.39
Q6	This type of caption is visually noisy.	3.96	1.52
Q7	This type of caption is useful for video without sounds.	4.25	1.34

Table 6.7: The mean and sample standard deviation for each question.

The result shows that the participants tended to think Dynamically Positioned Animation without Movement (DPA) especially superior than Statically Positioned Animation (SPA) in clarifying the sound source object (Q3) with appropriate positioning (Q2). Other differences are minor, but the result also shows the small tendency of participants to think that DPA provided more sound dynamics (Q1), made video watching enjoyable (Q5), and was more natural as a representation of sound (Q4). The difference in terms of visual noisiness was percieved tiny between SPA and DPA (Q6). Overall the participants tend to think that DPA is more useful for video sounds, but the difference is smaller than the preference of SPA to SPC.

Major category	**Minor Category**	**Major comments**
SPA Positive	Less Noisy(12)	"Less stressful to see", "Does not interfere with the visual", "Calm and stable"
	Easier to Follow (7)	"Do not need to being looked for", "Able to concentrate on visual", "Come in to sight naturally"
	Visual Impact (5)	"Sound words are being emphasized", "Has impact on visual"
	Others (6)	, "Easy to understand", "Visually nice"
SPA Negative	Noisy (13)	"Interfere with the visual", "The sound word is too large", "Hard to get into sight"
	Boring (11)	"Tame and boring", "Less vigorous", "Lack of reality"
	Less informative (4)	"Less clear on the sound source position", "Difficult to distinguish the sound source object"
	Others(4)	"The size is not appropriate", "Less expressive"
DPA Positive	Good Positioning(21)	"Easy to distinguish the sound source object", "More expessive to show near the sound source object", "Does not interfere with the visual"
	Visual Impact (17)	"Strong visual impact", "I had the sence of presence", "More vigorous"
	Easier to Follow (4)	"Less need to move my line of sight", "Easier to find next sound word to be appeared"
	Others (11)	"Interesting", "Visually expressive", "I can feel the sound visually", "Easier to imagine the sound"
DPA Negative	Noisy (13)	"Interfere with the visual", "Visually bad", "Makes me feel the sound words is smaller"
	Hard to follow (10)	"Hard to concentrate", "The sound words attract too much attention", "Tired to move my line of sight"
	Less Dynamics (4)	"I feel no sence of presence", "Cannot feel the sense of unity with car"
	Others(5)	"Unstable and I don't feel at ease", "Too much exaggeration", "Some sound words are out of bounds of the video"
Suggestive Comments	"Use different caption type depend on the scene", "Depend on the the preference of the audience", "Should move only on important or dynamic scene", "The sound word could be made as if it's boucing out from the car"	

Table 6.8: Summary of comments on "Please describe the advantage and disadvantage of caption type A and B, respectively" and "How this type of caption could be improved". Similar comments as previous user studies are not shown here.

Table 6.8 shows the comments gained from the participants as same as in SPC vs. SPA. It shows that the participants thought that SPA was better in that it was less noisy and easier to follow but boring, while DPA was hard to follow but good at positioning to distinguish the sound source and had more visual impact. However, some participants thought that DPA was easier to follow than SPA because the sound word appeared near the object (moving car) that got the attention of the participants. Similarly, same number of comments claims the visual noisiness to both SPA and DPA. It seems that the feeling of noisiness and easiness to follow varies among participants.

The suggestive comments were again picked up depending on its relevance to the difference between two methods. We found that same as in SPC vs. SPA, several participants suggested to change the type of annotation depending on the video contents or audience experience. Others suggested to move the sound word less and only for some important or dynamic scenes, which is similar to comments in the first user study that suggested the sound word animation "Should be limited to dynamic scenes".

DPA (Dynamically Positioned Animation without Movement) vs. DPAM (Dynamically Positioned Animation with Movement)

165 answers were gained and 106 of them were filtered as valid (64.2%). Figure 6.5 show the histograms of the answers in percentage and table 6.9 shows the mean and sample standard deviation of answers for each question.

Compared to previous two comparisons, less preferences of participants were found between Dynamically Positioned Animation without (DPA) and with Movement (DPAM). This is probably because the visual difference between the two methods are relatively smaller than previous two comparisons. The major differences between the two methods were that DPA was considered more appropriately positioned and more natural as representation of sound, while DPAM was considered visually noisier. However, this does not largely contributes to the overall usefulness of these annotation methods (Q7).

Table 6.10 shows the comments gained from the participants. The result shows that the participants tended to think that DPA was good with its good readability and easy to follow but was less interesting and boring, while DPAM had a stronger visual impact but was poor at readability and noisy. This is similar as the previous comparison between SPA and DPA, and could be explained by the indication in the suggestive comment "it is a trade-off between visual impact and visual cleanness". Several participants thought that the sound word tended to get out of the bonds of the video in DPAM and suggested to avoid it. Others pointed out that there was little difference between DPA and DPAM and the preference of the audience is rather personal. Again, this indicates that it is important to enable the audience to change between different methods.

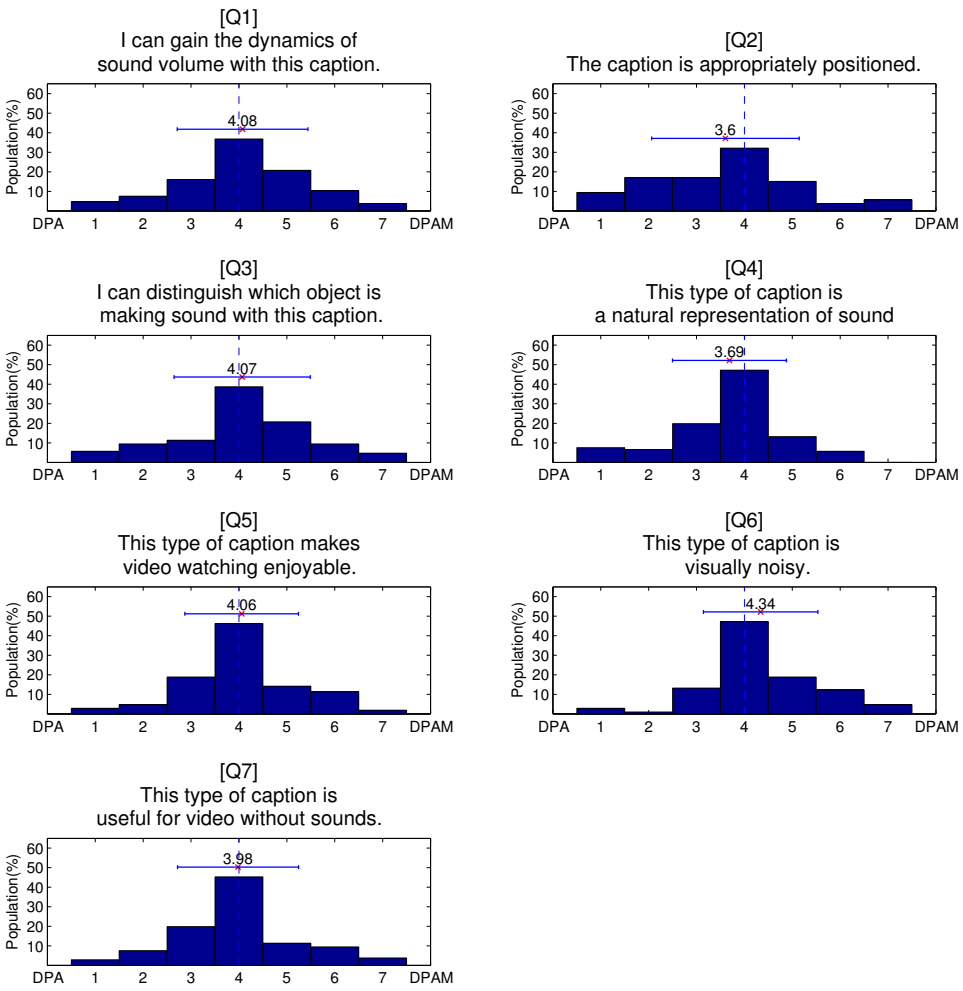

Figure 6.5: Histgram of answers for each question. The red cross shows the mean value and error bar shows sample standard deviation. All questions were answerd with 7-point A/B scale.

No	Question	mean	stdev
Q1	I can gain the dynamics of sound volume with this caption.	4.08	1.36
Q2	The caption is appropriately positioned.	3.60	1.54
Q3	I can distinguish which object is making sound with this caption.	4.06	1.42
Q4	This type of caption is a natural representation of sound.	3.69	1.19
Q5	This type of caption makes video watching enjoyable.	4.06	1.19
Q6	This type of caption is visually noisy.	4.34	1.19
Q7	This type of caption is useful for video without sounds.	3.98	1.26

Table 6.9: The mean and sample standard deviation for each question.

Major category	**Minor Category**	**Major comments**
DPA Positive	Good Readability (18)	"Easy to read", "Simple and clean", "Stable and good for reading"
	Visual Impact (9)	"I can feel the dynamics of the sound", "I had the sense of presence", "Strong visual impact"
	Easier to Follow (6)	"Able to concentrate on visual", "Easier to follow the car with sound words", "Less need to move my line of sight"
	Others (12)	, "Appropriate positioning", "Does not interfere the visual", "Calm and stable", "Visually nice"
DPA Negative	Boring (13)	"I had less sense of presence", "Less visual impact", "Tame and simple"
	Poor Readability (4)	"Hard to see", "Less healthy for my eye"
	Noisy (4)	"Attracts too much of my attention", "Visually noisy"
	Others(4)	"Too much sound words", "Harder to feel the presence of sound"
DPAM Positive	Visual Impact (29)	"I had sense of presence", "Strong visual impact", "Describes the dynamics of movement of car", "Visually attractive"
	Good Positioning (4)	"Easy to distinguish the sound source object", "Easy to understand what the sound is for"
	Good Readability (4)	"I can clearly see the sound words", "Easy to see"
	Others (6)	"Does not interfere with visual", "The timing is good"
DPA Negative	Poor readability (15)	"Hard to see", "Moves too much to read"
	Noisy (12)	"Interfere with visual", "Too noisty", "Noisy when it does not synchronize visual"
	Others(10)	"It is wired", "Moves too much and disgusting", "Sometimes the positioning is nonsense"
Suggestive Comments	"Should not placed out of bonds of video", "Little difference exists and depend on personal preference", "It is a tradeoff between visual impact and visual cleanness","Should reduce the amount of sound words"	

Table 6.10: Summary of comments on "Please describe the advantage and disadvantage of caption type A and B, respectively" and "How this type of caption could be improved". Similar comments as previous user studies are not shown here.

Chapter 7

Discussion

In this section we summarize the result of user study and discuss how different designs of sound word animation effects the audience experience. We also discuss important factors to construct a natural audience experience. Finally, we discuss how to choose from various design candidates for a good visualization of video sounds.

7.1 Design Guideline

The result of user study has revealed important guidelines for designing a sound word animation. These guidelines could be roughly categorized into 1) choice of sound word, 2) font style, 3) positioning style.

7.1.1 Font Style

We chose a very unique font that is similar to those used in contemporal Japanese comic books [8]. However, we found that a number of participants disliked the font design in the user study that compared videos with and without sound word animation. Many people suggested to use more standard fonts. There are mainly two reasons for this. First, since the sound word animation is always moving and each item has a very short duration, it is necessary to be able to grasp its contents in a very short time. Comical fonts tend to be more artistic and represents the characteristic of sound, but is harder to read in such a short time. Second, the combination of comical fonts and a live-action video may seem strange for some audience, and some of them thought that it made the video content "cheap". Indeed, in some existing products such as Manga-camera [9] the captured image is automatically edited to mimic the drawings in comic to provide a sense of unity. It would be interesting to mimic their methods for providing a cartoon-animation-like effects to the video, but this is out of our research focus.

Many participants recommended to add variety to the design. One idea to deal with this problem is to extend the work done by Yamamoto et al. [83] to generate a variety of font styles from the characteristics of the sound. Since the fonts generated by their method seems more "standard" than that used in our user study, it may also satisfy the requirement by some user to use standard fonts. However, as we described in chapter 2, their method need to be largely improved in order to deal with the mixture of sound and sustained sound of the video.

7.1.2 Choice of Sound Word

We have designed our prototype system to generate only two types of sound word based on its category. This is largely due to the limitation of contemporal sound

recognition technique. Not surprisingly, many participants think the generated sound words are monotonous and recommended to add more variety. Again, this could be achieved by extending the work of Yamamoto et al. [83] that has generated various sound word without recognizing semantic sound category (e.g. engine sound). Since their method still has a long way to deal with complex video sound, it would be better to combine parts of their idea with semantic categorization. This could also deal with the claims that the sound word chosen is unnatural, and is one of our important future work.

7.1.3 Positioning Style

The overall result of the user study for comparing different animation styles shows that dynamic positioning without movement were most preferred by the audience. Positioning the sound word near the sound source object seemed to provide a good sense of sound position, and also made watching video enjoyable with stronger visual impact. While some participants think it is harder to follow than statically positioned animation, others think it is easier because the animation always appeared near the audience's line of sight. This made the overall score concerning noisiness to be neutral (Q6 in figure 6.4). On the other hand, Dynamic Positioning with Movement were considered to be noisier, unnatural, and less appropriately positioned. Moving the position of the animation seemed to have a stronger visual impact, but strongly harmed the readability of sound word and perceived noisy.

As suggested by some comments, there are a trade-off between visual impact and visual cleanness. The result shows that the more drastically the position of sound word animation changes, the stronger visual impact it has. In our study the best balancing point between impact and cleanness seems to be dynamic positioning without movement. However, the combination of our method for animation and existing sophisticated techniques for static annotation may change this balancing point, as we mentioned in chapter 2.

7.2 Constructing Natural Audience Experience

The result of the user study that compared videos with and without sound word animation shows that animated sound words successfully provided the dynamics of sound volume to the audience and made the video enjoyable. On the other hand, it was still negatively perceived as a natural representation of sound. This may largely due to the design of fonts and sound word, considering a large number of claims on these elements. Although the design of these elements are not our research focus, a better method for determining these design parameters should be introduced in order to provide a natural audience experience.

We also found that many participants think the sound word should be shown less frequently and be limited to important or most dynamic scenes. This indicates that visualizing all the sound in the video does not much contribute to audience experience, and sometimes even does harm to it. In this thesis, we focus on faithful visualization of the actual sound in video. However, it turned out that for a natural audience experience, we should not mimic the sound as it actually appears in the video. There are mainly two reasons for this. First, the sound is not always synchronized with visual events. When the sound is not available, showing sound words at the point where no apparent visual events exist may be received unnatural. Second, people tend to be more sensitive on visuals than acoustics. Therefore, simply visualizing all the sound in the video may easily be

received as noisy. It would be an important future work to develop a method to filter the sound words based on visual dynamics and semantics of visual scene.

7.2.1 Choosing Suitable Design

Several comments suggested the choice of animation or positioning style should be selected depending on visual contents or the preference of the audience. Indeed, the comments given by between participants often conflict with each other, as mentioned above in design guideline of positioning style. The important thing is therefore not to find out the best choice but provide more options for the audience. Unlike handmade subtitles, our proposed method provides a fully automatic algorithm to visualize the sound. Therefore, various styles of animations can be easily generated by changing parameters in the algorithm. This could address the conflicting claims such as "The sound words are too large / too small", "Need more exaggeration / Need more stability" etc. On the other hand, the parameters to allow the audience to control should be carefully chosen in order not to confuse the audience with an excessive degree of freedom.

Chapter 8

Conclusion and Future Work

In this thesis, we proposed to automatically recognize the non-verbal sound in the video, and visualize the sound with sound words. The sound word is animated based on the volume of the sound it represents. We also proposed a method to dynamically position the generated sound word animation depending on the position of the sound source object. Our user study has shown that animated sound word could effectively visualize the dynamics of sound volume. It also contributes to making video watching enjoyable, and thought to be useful for video without sounds. The proposed dynamic positioning methods can clarify the position of the sound source object and increase the visual impacts.

On the other hand, several design problems such as choice of sound word or font types remain to be solved. As we discussed in chapter 7, a combination of our proposed method and previous work may attack these problems. In order to reduce visual noisiness, the number of sound words displayed and the amount of its dynamics also needs to be controlled by both visual and auditory content analysis. The system should also provide the audience an easy way to control various visualization parameters such as the size of sound words or the amount of dynamics.

One possible application of the proposed method would be video summarization. Various summarization methods have been proposed [72, 12], but none of them were able to summarize the audio in visual. Our method could add comic-like sound word to these visual summarization to provide an audio-visual summary to the user. Currently the prototype system we have implemented can only process car racing video and visualize two categories of sound. On the other hand, the proposed method could be applied to a wider range of video contents as development of visual and auditory recognition technology. We are quite optimistic on this point, given that the speed of improvement of these technologies is rapidly increasing.

References

[1] Amazon mechanical turk. https://www.mturk.com/mturk. [Online; accessed 20-February-2014].

[2] Batman (tv series 1966—1968). http://www.imdb.com/title/tt0059968. [Online; accessed 20-February-2014].

[3] Bergrennen - finger video. http://www.swissrace.ch/shop-items/bergrennen. [Online; accessed 20-February-2014].

[4] Closed captioning standards and protocol for canadian english language television programming services, third edition. http://www.cab-acr.ca/english/social/captioning/captioning.pdf. [Online; accessed 20-February-2014].

[5] Freesound.org. https://www.freesound.org. [Online; accessed 20-February-2014].

[6] Matlab. http://www.mathworks.com/products/matlab. [Online; accessed 20-February-2014].

[7] Yahoo!クラウドソーシング. http://crowdsourcing.yahoo.co.jp. [Online; accessed 25-January-2014].

[8] 荒木飛呂彦 公式サイト [jojo.com]. http://www.araki-jojo.com. [Online; accessed 20-February-2014].

[9] 漫画カメラ. http://tokyo.supersoftware.co.jp/mangacamera. [Online; accessed 20-February-2014].

[10] Rick Altman. *Silent film sound*. Columbia University Press, 2004.

[11] Jimmy Azar, Hassan Abou Saleh, and Mohamad Adnan Al-Alaoui. Sound Visualization for the Hearing Impaired. *International Journal of Emerging Technologies in Learning*, 2(1):1–7, 2007.

[12] Connelly Barnes, Dan B. Goldman, Eli Shechtman, and Adam Finkelstein. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics*, 29(4):89:1–89:9, July 2010.

[13] Jonas Beskow, Olov Engwall, Peter Nordqvist, and Preben Wik. Visualization of speech and audio for hearing impaired persons. *Technology and Disability*, 20(2):97–107, 2008.

[14] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152. ACM, 1992.

[15] Simon Bouvier-Zappa, Victor Ostromoukhov, and Pierre Poulin. Motion cues for illustration of skeletal motion capture data. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '07, pages 133–140. ACM, 2007.

[16] Rui Cai, Lei Zhang, Feng Jing, Wei Lai, and Wei-Ying Ma. Automated music video generation using web image resource. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2 of *ICASSP '07*, pages II–737–II–740. IEEE, April 2007.

[17] Wei Chai and Barry Vercoe. Music thumbnailing via structural analysis. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, MULTIMEDIA '03, pages 223–226. ACM, 2003.

[18] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011.

[19] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. Attribit: Content creation with semantic attributes. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 193–202. ACM, 2013.

[20] M. G. Choi, K. Yang, T. Igarashi, J. Mitani, and J. Lee. Retrieval and visualization of human motion data via stick figures. *Computer Graphics Forum*, 31(7pt1):2057–2065, September 2012.

[21] Selina Chu, Shrikanth Narayanan, C.-c. Kuo, and Maja Mataric. In *Proceedings of 2006 IEEE International Conference on Multimedia and Expo*, ICME '06, pages 885–888. IEEE.

[22] Selina Chu, Shrikanth Narayanan, and C.-C. Jay Kuo. Environmental sound recognition using MP-based features. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–4. IEEE, March 2008.

[23] Selina Chu, Shrikanth Narayanan, and C.-C. Jay Kuo. Environmental Sound Recognition With Time—Frequency Audio Features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1142–1158, August 2009.

[24] Dailymotion. http://www.dailymotion.com. [Online; accessed 20-February-2014].

[25] Alain Dufaux, Laurent Besacier, Michael Ansorge, and Fausto Pellandini. Automatic detection and classification of wide-band acoustic signals. In *Proceedings of the 137 th Meeting of the Acoustical Society of America and Forum Acusticum*, ASA '99, pages 14–19, 1999.

[26] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and Appearance for Mobile Scene Analysis. In *2007 IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 (voc2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. [Online; accessed 20-February-2014].

[28] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The pascal visual object classes challenge 2006 (voc2006) results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf. [Online; accessed 20-February-2014].

[29] Mark Everingham, Luc Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, September 2009.

[30] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–45, September 2010.

[31] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, MULTIMEDIA '99, pages 77–80. ACM, 1999.

[32] Jonathan Foote, Matthew Cooper, and Andreas Girgensohn. Creating music videos using automatic media analysis. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 553–560. ACM, 2002.

[33] Jodi Forlizzi, Johnny Lee, and Scott Hudson. The kinedit system: Affective messages using dynamic texts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 377–384. ACM, 2003.

[34] Federico Girosi. An equivalence between sparse approximation and support vector machines. *Neural computation*, 10(6):1455–1480, 1998.

[35] R.S. Goldhor. In *1993 IEEE International Conference on Acoustics Speech and Signal Processing*, ICASSP '93, pages 149–152. IEEE.

[36] Dan B Goldman, Brian Curless, David Salesin, and Steven M. Seitz. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics*, 25(3):862, July 2006.

[37] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 3–12. ACM, October 2008.

[38] Masataka Goto. Active Music Listening Interfaces Based on Signal Processing. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4 of *ICASSP '07*, pages IV–1441–IV–1444. IEEE, 2007.

[39] R. Gribonval, E. Bacry, S. Mallat, P. Depalle, and X. Rodet. Analysis of sound signals with high resolution matching pursuit. In *Proceedings of 3rd International Symposium on Time-Frequency and Time-Scale Analysis*, TFTS '96, pages 125–128. IEEE.

[40] Richang Hong, Meng Wang, Mengdi Xu, Shuicheng Yan, and Tat-Seng Chua. Dynamic captioning: Video accessibility enhancement for hearing impairment. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 421–430. ACM, 2010.

[41] Xian-Sheng HUA, Lie LU, and Hong-Jiang ZHANG. Automatic music video generation based on temporal pattern analysis. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 472–475. ACM, 2004.

[42] Kazushi Ishihara, Tomohiro Nakatani, Tetsuya Ogata, and HiroshiG. Okuno. Automatic sound-imitation word recognition from environmental sounds focusing on ambiguity problem in determining phonemes. In Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors, *PRICAI 2004: Trends in Artificial Intelligence*, volume 3157 of *Lecture Notes in Computer Science*, pages 909–918. Springer Berlin Heidelberg, 2004.

[43] Junki Ito, Masayoshi Kanoh, Tsuyoshi Nakamura, and Takanori Komatsu. Editing robot motion using phonemic feature of onomatopoeias. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 17(2):227–236, 2013.

[44] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456. ACM, 2008.

[45] Takanori Komatsu. Quantifying Japanese onomatopoeias: toward augmenting creative activities with onomatopoeias. In *Proceedings of the 3rd International Conference on Augmented Human*, AH '12, pages 1–4. ACM, March 2012.

[46] Naoko Kosugi. Misual: Music visualization based on acoustic data. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '10, pages 609–616. ACM, 2010.

[47] Johnny C. Lee, Jodi Forlizzi, and Scott E. Hudson. The Kinetic Typography Engine: An Extensible System for Animating Expressive Text. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, UIST '02, pages 81–90. ACM, October 2002.

[48] Joonhwan Lee, Soojin Jun, Jodi Forlizzi, and Scott E. Hudson. Using kinetic typography to convey emotion in text-based interpersonal communication. In *Proceedings of the 6th Conference on Designing Interactive Systems*, DIS '06, pages 41–49. ACM, 2006.

[49] J. Mantyjarvi, P. Huuskonen, and J. Himberg. Collaborative context determination to support mobile terminal applications. *IEEE Wireless Communications*, 9(5):39–45, October 2002.

[50] Mitsunori Matsushita and Natsumi Imaoka. Kinetic Onomatopoeia Generation System for Creating an Attractive Digital Comic. In *Proceedings of the 25th Annual Conference of the Japanese Society for Artificial Intelligence*, number 072, pages 3–6, 2011.

[51] Tara Matthews, Janette Fong, and Jennifer Mankoff. Visualizing non-speech sounds for the deaf. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '05, pages 52–59. ACM, 2005.

[52] Tomoyasu Nakano, Sora Murofushi, Masataka Goto, and Shigeo Morishima. DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Dance Video Clips on the Web. In *Proceedings of the 8th Sound and Music Computing Conference*, SMC 2011, pages 183–189, 2011.

[53] Suranga Nanayakkara, Elizabeth Taylor, Lonce Wyse, and S H. Ong. An enhanced musical experience for the deaf: design and evaluation of a music display and a haptic chair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 337. ACM, April 2009.

[54] M. Nienhaus and J. Dollner. Depicting Dynamics Using Principles of Visual Art and Narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51, May 2005.

[55] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z. Gajos. Platemate: Crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 1–12. ACM, 2011.

[56] Jason Orlosky, Kiyoshi Kiyokawa, and Haruo Takemura. Dynamic text management for see-through wearable and heads-up display systems. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 363–370. ACM, 2013.

[57] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based organization and visualization of music archives. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 570–579. ACM, 2002.

[58] ProjectM. http://projectm.sourceforge.net. [Online; accessed 20-February-2014].

[59] Raisa Rashid, Jonathan Aitken, and DeborahI. Fels. Expressing emotions using animated text captions. In Klaus Miesenberger, Joachim Klaus, WolfgangL. Zagler, and ArthurI. Karshmer, editors, *Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 24–31. Springer Berlin Heidelberg, 2006.

[60] Vy Quoc. Hunt Richard. Rashid, Raisa. and Deborah I. Fels. Dancing with Words: Using Animated Text for Captioning. *International Journal of Human-Computer Interaction*, 24(5):505–519, June 2008.

[61] Lionel Reveret, Gérard Bailly, and Pierre Badin. MOTHER: A new generation of talking heads providing a flexible articulatory control for video-realistic speech animation. In *International Conference of Spoken Language Processing*, ICSLP '00, 2000.

[62] Edward Rosten, Gerhard Reitmayr, and Tom Drummond. Real-time video annotations for augmented reality. In George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin, editors, *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 294–302. Springer Berlin Heidelberg, 2005.

[63] Stephanie Santosa, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. Direct space-time trajectory control for visual media editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1149–1158. ACM, 2013.

[64] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis. In *2009 IEEE 12th International Conference on Computer Vision*, pages 24–31. IEEE, September 2009.

[65] David A. Shamma, Bryan Pardo, and Kristian J. Hammond. Musicstory: A personalized music video creator. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 563–566. ACM, 2005.

[66] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[67] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180. IEEE.

[68] Statistics-Youtube. http://www.youtube.com/yt/press/statistics.html. [Online; accessed 20-February-2014].

[69] Carlo Strapparava and Alessandro Valitutti. Bringing the text to life automatically. In *Proceedings of the AAAI-2006 Workshop on Computational Aesthetics: AI Approaches to Beauty and Happiness*, 2006.

[70] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[71] Ayaka Terashima, Hiroki Uema, and Mistunori Matsushita. A Method for Generating Kinetic Onomatopoeia based on Lines Drawn to Represent Motion. In *Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence*, pages 63–68, 2012.

[72] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video manga: Generating semantically meaningful video summaries. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, MULTIMEDIA '99, pages 383–392. ACM, 1999.

[73] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 606–613. IEEE, September 2009.

[74] Tobias Höllerer Vineet Thanedar. Semi-automated Placement of Annotations in Videos. Technical report, UC, Santa Barbara, November 2004.

[75] Paul Viola and Michael J Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[76] Watch visualizations while playing your music - Microsoft Windows Help. http://windows.microsoft.com/en-us/windows/watch-visualizations-playing-music. [Online; accessed 20-February-2014].

[77] Sanae H. WAKE and Toshiyuki ASAHI. Sound Retrieval with Intuitive Verbal Descriptions. *IEICE TRANSACTIONS on Information and Systems*, E84-D(11):1568–1576, November 2001.

[78] Jia-Ching Wang, Hsiao-Ping Lee, Jhing-Fa Wang, and Cai-Bei Lin. Robust Environmental Sound Recognition for Home Automation. *IEEE Transactions on Automation Science and Engineering*, 5(1):25–31, January 2008.

[79] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained Linear Coding for image classification. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3360–3367. IEEE, June 2010.

[80] E. Wold, T. Blum, D. Keislar, and J. Wheaten. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

[81] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.*, 5:975–1005, December 2004.

[82] Songhua Xu, Tao Jin, and Francis C. M. Lau. Automatic Generation of Music Slide Show Using Personal Photos. In *2008 Tenth IEEE International Symposium on Multimedia*, pages 214–219. IEEE, December 2008.

[83] Takashi Yamamoto, Masaki Matsubara, and Hiroaki Saito. Visualization of Environmental Sounds using Onomatopoeia and Effective Fonts. *The Journal of the Society of Art ans Science*, 11(1):1–11, 2012.

[84] In-Chul Yoo and Dongsuk Yook. Automatic sound recognition for the hearing impaired. *IEEE Transactions on Consumer Electronics*, 54(4):2029–2036, November 2008.

[85] Youtube. http://www.youtube.com. [Online; accessed 20-February-2014].

[86] Hao Zhang, A.C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136, 2006.

[87] 松下 光範. コミック工学の可能性. In **第** *2***回** *Web***インテリジェンスとインタラクション研究会**, volume 3, pages 63–68, 2011.