A System for Adaptive Video News Story Tracking Based on the Earth Mover's Distance

Mats Uddenfeldt

Master's Thesis

Supervisor: Keiichiro Hoashi
Examiner: Tomas Olofsson

Department of Engineering Sciences
Uppsala University
Sweden

February 20, 2006

Abstract

Every day there is an abundance of information broadcasted by all the news networks of the world. An automatic news story tracking system could watch all broadcasted news and track the stories in which we are interested in. In this thesis, a novel approach to news story tracking is investigated. We have designed a system for adaptive video news story tracking based on the Earth Mover's Distance (EMD). The EMD, which was originally presented as a solution to the transportation problem, is used as a similarity measure between the visual features of stories. When an interesting story appears in the news, it is flagged manually as a topic for tracking. Our system will then track the events as they unfold over time and present accumulated results to the user for feedback. This feedback is used to adapt the topic model to changes in the tracked story. The EMD provides the system with a robust way of performing many-to-many matching of news stories independent of the temporal order of their contents. This is particularly suitable in the news genre as stories often are subjected to extensive video editing between shows. Experiments have been run with a range of topics from multiple networks and show promising results.

Contents

Chapter 1

Introduction

There are many news agencies today that broadcast reports on what is happening all over the world. Large networks provide national and global coverage, whereas smaller networks cover more local events. The reports are instant and cover an impressive range of topics, including natural disasters, politics, economics, sports and other events. Due to the vast range of topics and target audiences, the nature of the news stories are very different. In addition to this, reports may be biased and differ depending on the cultural background of the individual network. Although most news networks will claim otherwise, their political standpoint may also influence the reports to some degree. This makes it necessary to watch news from multiple sources, in order to provide us with a more objective view of major stories. There are clearly certain stories in which you, as a viewer, would be more interested in following over others. However, the total number of stories broadcasted every day makes it a daunting, or even impossible, task to watch everything. Therefore we introduce an automatic news story tracking system as a solution to this problem.

A news tracking system could watch all news broadcasted over the world, and return only the relevant stories. This would significantly reduce the number of stories that a viewer would have to watch and make it possible to construct a multifaceted picture of a given event without the need of watching countless hours of unrelated news material. A user would be able to come home and receive the latest updates on the stories, that he or she considers interesting.

Despite that a lot of work has been done in the domain of video information retrieval, the task of tracking news stories presents a new kind of challenge. A news story cannot be represented as a static entity; it will develop over time as the story unfolds in the news. In order to be efficient, it is therefore necessary for the tracking system to adapt to changes in the focus of the story.

In this thesis we investigate a novel approach to news story tracking. An interesting story is chosen by the user and the system will then track similar stories in the news flow. We produce the basis of a news story tracking system and show how we can achieve semantic linkage between stories using only visual feature, even as the focus of the story changes.

1.1 Problem description

The task of finding a way of retrieving relevant news stories from a given news feed is highly challenging. It needs to be effective both in regards to computation cost and in the number of relevant hits. There are many ways in which we can represent the semantic information of a story, so that an automated system can understand it. When such a representation has been found, we will have to find an way to compare the similarity of two stories, to retrieve a value for their degree of semantic matching. In addition to these problems, the news genre present us with yet another problem, as the stories are not static, but change and shift their focus as the story unfolds. This leaves us with the following list of problems:

- How can a representation of a news story be generated?
- How can the similarity between stories be calculated?
- How can we track a story as it changes over time?

1.2 Goals and methods

The main goal of this thesis was to design a system that can track a story as it develops in the news over time. The design would have to be robust in order to work with many different kinds of stories, as well as different networks. It was important that the system could filter out redundant stories from the news feed, in order to reduce the number of stories returned to the user. The work was to serve as a foundation for future experiments and research.

Before we could design such a system, we set upon a theoretical investigation of how we could represent the semantic information of a story and which similarity measures would be appropriate to use in this context. We then proceeded to implement the system to measure the efficiency of the design. Experiments were run to investigate the optimal parameters of the system and to test its robustness. The results of the experiments were evaluated using conventional measures of information retrieval: precision and recall. In addition to this the F-measure, an harmonic mean of precision and recall, was used to provide a single measure of the tracking efficiency.

1.3 Thesis Structure

This thesis is structured as follows. In Chapter 2 we present a theoretical background to the problem of tracking news stories and existing solutions. Our proposed tracking system is described in detail in Chapter 3. Chapter 4 explains how our experiments where set up, and Chapter 5 show the results of our evaluation. The results are analyzed and discussed in Chapter 6. Chapter 7 concludes the paper.

Chapter 2

Background

This chapter serves as a theoretical introduction to the problem of video information retrieval in general, and news story tracking in particular. First, in Section 2.1, we look at how it is possible to generate a representation of a news story. Section 2.2 describes different ways of achieving a similarity measure between different news stories and Section 2.3 describes previous work in news story tracking.

2.1 Understanding news stories

In order to understand news stories and, more interestingly, to provide semantic linkage between news stories, an important starting point is the way a viewer normally perceives the information broadcasted in the news. Our first impression of a news story is, as with most other things we encounter, visual information: we see images describing the events of the story and we recognize locations and faces of the people featured in the story. In addition to the pure visual information, we also acquire language information from the story in the form of audio and closed captions on screen. If we were to watch two news stories run in parallel, we can judge how similar they are based on these characteristics. In other words, semantic linkage can be accomplished using either visual or language information, or a combination of the two.

Correspondingly, there are two major approaches one can take in order to generate a representation of a news story: the text-based approach or the audio/visual-based approach. In the case of the text-based approaches, textual information from the video is used to create a searchable index of stories. The text is extracted from the closed captions on screen, and sometimes automatic speech recognition (ASR) is used to gain access to even more textual information. The audio/visual-based approaches use audio or video features of the video to generate a representation of the story. A distance measure is defined for these features, so that the similarity of two stories can be calculated. The measure is used to detect stories which are close to each other in regards of the featured content.

For query-based story retrieval in the news genre, the text-based approaches have turned out to be quite successful. However, current language information techniques, extending beyond the use of closed-caption text, have to rely on ASR models, which are not very accurate. In this context the audio/visual-based approaches look more promising. Although an individual shot is merely a series of frames with continuous camera motion, a story is a series of such shots, with coherence from a narrative point

of view. Thus, it should be possible to rely on this information to provide semantic linkage between stories.

2.2 Story-based similarity measures

During the past decade numerous techniques have been developed to provide query-based video retrieval. We needed to study these to find a way to provide similarity ranking between news stories, or video clips as they are referred to in a more general context. We will stay with the term *news story* or *story* throughout this thesis.

Most of the previous work focus on the retrieval by a single shot, rather than retrieval by multiple shots, i.e., complete news stories. Existing approaches based on multiple shots either rely on rapid clustering of similar stories without internal order [1, 2] or similarity ranking of stories [3–6]. The fast algorithms suggested in [1, 2] creates signatures, e.g., combined color histograms, to represent the contents of the stories and the similarity depends on the distance between the signatures. These techniques are suitable to group stories with next-to identical content and have been successfully applied to retrieve commercials from large video databases. However, due to the diversity of content in the news genre, the fast algorithms presented in [1, 2] are not useful in our case.

In [3–6], story-based retrieval is built upon the shot-based retrieval. Besides relying on shot similarity, these methods are also dependent on the inter-relationship of shots such as the temporal order, granularity [1] and interference [2] to calculate story similarity. Relying on temporal order for similarity, as in [5], would not be a good idea in the news genre. News stories often exist in several editions, with different internal shot order. If we were to compare two stories, with identical shots but different shot order, we would want to get the same similarity result as in the case of identical shot order.

The more sophisticated methods of granularity and interference look at the level of one-to-one shot matching and the ratio of unmatched shots. When put to use in the news genre these methods run into problems due to video editing effects. In news, the contents of a long shot is often segmented into multiple shorter shots in later editions of a story. Some of these shots are used repeatedly, so that the viewer can recall the events leading up to the current report. Furthermore, errors in shot boundary detection can result in both oversegmentation or the incorrect merging of different shots. This will also result in a mixture of shots, in which one-to-one mapping between shots for similarity would be highly inappropriate.

On the other hand, the method presented in [7] by Peng et al. suggests that one can successfully use the Earth Mover's Distance (EMD) [8] to provide a many-to-many mapping between the shots of two stories. The EMD provides an efficient solution to the well-known transportation problem: to find the least expensive flow of goods from a number of suppliers to a number of consumers. Suppose that we have a given number of suppliers, each with a given amount of supplies, who want to supply a given number of consumers, each with a given limited capacity to accept goods. For each supplier-consumer pair, the cost of transporting a single unit of goods is given as the distance between the supplier and the consumer. The transportation problem is to find the minimum expensive flow of goods, that satisfies the demand of the consumers, from all of the suppliers to all of the consumers. Peng et al. [7] implements a story-based similarity measure, by casting one of the stories as the supplier side and the

[1] The level of one-to-one shot matching.
[2] The ratio of unmatched shots.

other as the consumer side. The EMD is then deployed to compute the minimum cost of transporting frames from one story to the other. This cost can be used as a distance measure between the two stories.

2.3 Tracking news stories

News story tracking is different from news story retrieval, since it needs a way of following the development of a story over time. Despite this, previous solutions to the problem of tracking news stories have usually taken a query-based approach. Given a database of news stories, an index is built using either a text-based or audio/visual-based approach. A user will present the system with a query, which return a list of result in order of relevance.

Zhai et al. [9] proposed an advanced framework using a fusion of both visual and language information to create the story representations for the index. They used two distinct methods for visual data: one specialized for facial extraction and one for use with non-facial keyframes. In addition to this, they extracted closed captions and used ASR to gain access to even more textual information. The combined information was used to compute a similarity mapping to match a given query story with other stories within the video archive.

Ide et al. [10] took a different approach. They presented a database management system for the purpose of structuring a large news video archive. Their goal was not tracking per se, but to provide order in chaos by introducing topic threading between similar stories. The news programs are first segmented into topics by applying morphological and semantic analysis of closed-caption text. These topics are then threaded into the video database in a chronological order, based on their degree of semantic linkage. In their approach the user will then manually track the story by following relevant links in the resulting threaded news archive.

We see the problem of tracking news stories as a problem concerning online learning of user needs. Since the nature of a story changes as it develops over time, the earlier query-based approaches are not sufficient: their scope is static and therefore too limited. We need to find a way in which we can adapt our query based on accumulated results and user feedback. By doing so, we can follow an individual news topic, even though its focus changes over time.

Chapter 3

Proposed Tracking System

In this chapter we will present the details of our proposed adaptive news story tracking system. We discuss the design decisions and provide an overview of the system in Section 3.1. We will then describe the individual pieces of the system, starting with the feature extraction in Section 3.2. Our implementation of a story-based similarity measure is explained in Section 3.3 and Section 3.4 offers a full description of how the system works and how we achieve adaptive tracking.

3.1 Overview

When deciding how to design our tracking system, we weighed the pros and the cons of the different approaches described in Chapter 2. We decided to design our tracking system based solely on the visual features of news stories. We opted out on the text-based approach, because we believe that current language information techniques, relying on closed-caption text or ASR models for textual information, are neither accurate or fast enough to be used effectively in an online system. Although separate shots cannot be used to provide a semantic context, a news story is a series of shots with coherence from a narrative point of view. From an entropy point of view, a news story featuring many different shots holds a relatively high amount of information. Our assumption is that the visual information contained in a single story can be used to provide the necessary semantic linkage between our stories. Besides, we believe that a visual-based approach would be more effective in comparison to running ASR on the stories.

Knowing that our system would have to provide a many-to-many mapping between stories in order to provide good results in the news genre, we chose to implement a story representation along the lines of the one presented by Peng et al. [7]. We model two stories as a weighted graph with two vertex sets. Each vertex represents the keyframe of a shot, and is stamped with its duration in frames as its weight. The cost of transporting a frame is given by a color histogram distance function. The EMD can then be used to determine the distance between two stories. A detailed description of our implementation can be found in Section 3.3.

When designing the tracking part of the system, we had to make sure that we could adapt our query in order to fit the changes in a story over time. We propose a news story tracking system in which we adapt our query based on accumulated results and user feedback. A flow-chart overview of the system can be seen in Figure 3.1 and a brief explanation follows:

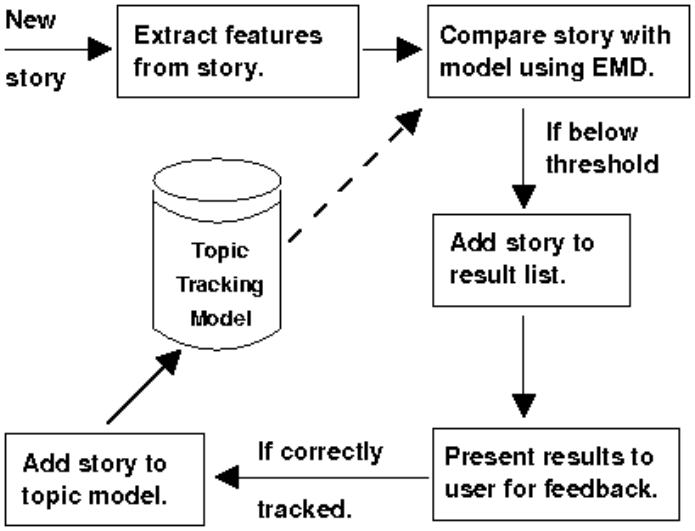


Figure 3.1: A flow-chart overview of the adaptive tracking system.

A single on-topic story is chosen by the user to initialize the system - we will refer to this story as the *init story* - and it is added to the topic tracking model. New stories that enter the system will be compared against the stories of the topic model to judge their relevance to the topic being tracked. The results are supervised by the user, and we will use the correctly tracked stories to adapt our query, by adding them to the topic model. In this way we can follow a story, even though its focus changes over time.

3.2 Feature extraction

In order to create a representation of a story we need to extract its visual features. Since the goal of our work was to design and evaluate a tracking system, we will assume that the news feed has already been segmented into shots and stories. There are many different techniques in which one can achieve this segmentation [11, 12], neither of which is favored by our system.

When fed with a segmented news feed our system detects the boundaries of stories and their shots. We only extract features from non-studio shots from the report segment of each story. To create a representation of the story, we extract features from every shot in the form of a keyframe. The system currently takes a naive approach to keyframe extraction, by simply using the middle frame of every shot. A color histogram is computed from the keyframe and is used to represent the entire shot. This approach is fast and simple, but we cannot be certain that the selected keyframe is a good representative of the entire shot. At the same time using the color histogram without any spatial information means that two frames featuring similar colors but different objects will be considered similar.

We have used Haar encoded color histograms in the HSV color space, as defined for the Scalable Color Descriptor (SCD) in the MPEG-7 Visual Standard [13]. The HSV color space is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color.

3.3 Similarity measure

Existing techniques to provide similarity ranking between stories are investigated in Section 2.2. During our investigation of those techniques, we argued that methods, that enforce a one-to-one mapping between the individual shots, were not appropriate for use in the news genre, mainly due to heavy video editing between the different editions of, and even within, news shows. However, the method presented in [7] by Peng et al. shows that it is possible to use the EMD to provide a many-to-many mapping between the shots of two stories.

To be able to use the EMD as a story-based similarity measure we implemented a story representation along the lines presented in [7]. Each story is represented as a weighted graph composed of its shots. Each shot is represented as a tuple composed of the color histogram feature of the keyframe, as the value, and the duration (number of frames) of the shot, as the weight. In analogy with the original transportation problem, the shots of story A are considered to be the suppliers and the shots of story B the consumers. The cost of transporting a single frame between two shots a_i and b_j is calculated using a histogram distance measure. Here we take a different approach from [7], and use the normalized L1-distance between the two HSV histograms as our cost function. We used the L1 metric because it corresponds to the "histogram intersection", the most commonly used similarity measure for histograms:

$$Dist_{L1norm}(H_a, H_b) = \frac{\sum_{n=1}^{N}(H_{a_n} - H_{b_n})}{\sum_{n=1}^{N} H_{a_n}} \tag{3.1}$$

Given two clips A and B, two weighted graphs G_A and G_B are constructed as follows:

- Let $G_A = \{(a_1, \omega_{a_1}), (a_2, \omega_{a_2}), ..., (a_m, \omega_{a_m})\}$ be the supplier story with m shots, where a_i represents a shot in A and ω_{a_i} is the number of frames in shot a_i.
- Let $G_B = \{(b_1, \omega_{b_1}), (b_2, \omega_{b_2}), ..., (b_m, \omega_{b_n})\}$ be the consumer story with n shots, where b_i represents a shot in B and ω_{b_j} is the number of frames in shot b_j.
- Let $D = \{d_{ij}\}$ be a distance matrix, where d_{ij} is the distance between shots a_i and b_j defined as the normalized L1 distance between their respective keyframes:

$$d_{ij} = Dist_{L1norm}(keyframe(a_i), keyframe(b_j)) \tag{3.2}$$

- We want to find a flow $F = [f_{ij}]$ between G_A and G_B, where f_{ij} is the flow between a_i and b_j, that minimizes [8] the overall cost:

$$WORK(G_A, G_B, F) = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij} \tag{3.3}$$

subject to the following constraints:

$$f_{ij} \geq 0 \text{ , where } 1 \leq i \leq m \text{ and } 1 \leq j \leq n \tag{3.4}$$

$$\sum_{j=1}^{n} f_{ij} \leq \omega_{a_i} \text{ , where } 1 \leq i \leq m \tag{3.5}$$

$$\sum_{i=1}^{m} f_{ij} \leq \omega_{b_j} \text{ , where } 1 \leq j \leq n \tag{3.6}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m}\omega_{a_i}\sum_{j=1}^{n}\omega_{b_j}\right) \tag{3.7}$$

Constraint (3.4) allows moving frames from G_A to G_B and not vice versa. Constraint (3.5) limits the amount of frames that can be sent by the shots in G_A to their weights. Constraint (3.6) limits the shots in G_B to receive no more frames than their weights, and constraint (3.7) forces to move the maximum amount of frames. We call this amount the *total flow*. Once the transportation problem is solved, and we have found the optimal flow F, the EMD is defined as the resulting work normalized by the total flow:

$$EMD(G_A, G_B) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{3.8}$$

We can now use the EMD to find the minimum expensive flow to transport frames between the supplier shots and the consumer shots. The normalization factor is the total weight of the smaller story as indicated in Constraint (3.7). The output of Equation (3.8) is thus normalized in the range of [0, 1]. This makes it possible for us to define the similarity between two stories, A and B, as:

$$Sim_{story}(G_A, G_B) = 1 - EMD(G_A, G_B) \tag{3.9}$$

Equation (3.9) is normalized in the range of [0, 1], with the higher values representing more similar stories.

3.4 Adaptive tracking of news stories

We propose an adaptive news story tracking system in order to be able to follow a story as it develops in the news over time. We see the problem of tracking news stories as a problem concerning online learning of user needs, in this case the desire to follow a particular story in the news feed.

The system is initiated when the user manually flags the init story, which represents the topic in which he or she is interested in tracking. The init story is used as the first story of the topic tracking model. When a new story enters the system, it will be compared against all stories of the topic model. If it is considered similar to at least one of the stories, it will be considered a hit and added to the list of results. The nature of the story will undoubtedly change over time, but if the changes occur in controlled increments we should be able to adapt our model to the changes. The actual update is supervised by the user, by giving feedback on the accumulated list of results after a given time interval. A detailed description of the steps shown in the flow-chart diagram of Figure 3.1 follows below:

0. Init story : The user initializes the tracking system, by flagging a single interesting story for tracking. A weighted graph represent-ation of the init story is created using the technique described in Section 3.3. The init story is added to the topic model, and thus defines the story which we are about to track.

1. New story : A new story enters the topic tracking system from the news feed.

2. Extract features from story : For every shot in the new story, a keyframe is selected and its visual features are extracted in order to create a weighted graph representation of the story.

3. Compare story with model using EMD : The new story is compared against all the stories of the topic using the EMD as defined in Equation (3.8). The most similar result is compared against a defined threshold for least matching distance. If the story is considered a match, we add it to the result list. If not, it is discarded and the system will wait for the next new story.

4. Present results to user for feedback : The accumulated list of results is presented to the user after a regular time interval. The user will go through the list of result and flag the correct results. Incorrectly tracked stories are discarded.

5. Add story to topic model : The correctly tracked stories reflect the latest development in the tracked story, and are therefore used to update the topic model. This is the adaptive step of our tracking system, in which the topic model changes based on the user feedback.

The topic model is initially composed only of the manually flagged story, but as time progresses the model is updated with more recent stories. The model is operated as a FIFO (First In, First Out) list of the latest correctly tracked stories. Tracking will cease when a given time-out period has passed without a single correct result being found on the list of accumulated results. This indicates either that the story has disappeared from the news, or that the system has failed to update the topic model according to the changes in the story. The system can be controlled through the use of three parameters:

Topic model size : The topic model size, N, is the number of stories to hold in the topic model. It decides how far back the adaptive memory of the system stretches.

Similarity threshold : The similarity threshold, S, controls how many of the observed stories end up on the result list. It defines the maximum distance that can exist between two matching stories.

Time-out period : The time-out period, T, defines how long we should wait with no matching results before we consider a story dead, and cease to track a story. It is given in number of days.

Chapter 4

Experiments

This chapter provides the details on the experiments we have conducted with the implementation of our news tracking system. The data set used in our experiments is described in Section 4.1. The method in which we conducted our experiments can be found in Section 4.2 and the measures used for evaluation in Section 4.3.

4.1 Experiment data

To evaluate the performance of the proposed system, we have prepared a video database which consists of approximately 530 hours of Japanese news shows, recorded over a two-month period between March and May of 2005. The video was recorded in MPEG-1 format and stored as individual full-length files. In total the news shows make up ten different programs spread over five separate news networks. A list of the recorded shows along with some of their characteristics can be seen in Table 4.1.

The preprocessing of the experiment data includes shot segmentation, story segmentation, story labeling, keyframe extraction and similarity measure calculation. The material was subjected to automatic shot boundary detection using the techniques presented in [14]. We then let human subjects go through the material to perform story boundary detection and manually label all the stories in the database according to type (introduction, report, weather, sports) and topic. The story types are general descriptors of the story contents and the story topics indicate major stories that occurred within the period. During the two months a total of ten "hot" topics dominated the news feed. The topic labels were used as reference data in our experiments, and the top ten can be seen together with a brief description in Table 4.2. The video was annotated with according to the MPEG-7 standard [13]. Story boundaries were set using the *MediaTimePoint* element and story types and labels were set using the *FreeTextAnnotation* element. An example of an annotated file can be found in Appendix A. Furthermore, because the data had been annotated with story type, we could limit our experiments to the non-studio shots only, i.e., video footage such as reports from the site of the story. These parts of the stories, from now on referred to as the *report segments*, provide visual information, which can be used to compare different stories. The report segments were annotated using the characters *VTR*, an acronym which stands for Video Tape Recorder.

When calculating similarity between stories based on visual features, it is inappropriate to use studio shots. The studio shots are next to identical between different stories on the same network, and can even be similar across different networks. As

Recorded show	Length	Stories / Show	Shots / Story	Seconds / Shot
Asahi #1	2h	21.9	37.6	4.9
Asahi #2	1h 10m	14.5	36.3	5.0
Fuji #1	2h	21.8	37.4	4.9
Fuji #2	25m	6.0	24.6	5.3
NHK #1	30m	15.4	12.4	6.8
NHK #2	50m	20.6	19.4	6.4
NTV #1	1h 30m	20.1	31.9	4.8
NTV #2	30m	6.2	30.6	5.2
TBS #1	1h	19.6	20.5	4.9
TBS #2	50m	15.3	28.2	4.2

Table 4.1: Summary of the recorded shows in the experiment data set. Length is the entire length of the show. *Stories/Show* is the average number of stories per show. *Shots/Story* and *Seconds/Shot* were calculated using only the report segments.

Topic	Description
Anti Japan Demonstration	Anti-Japan demonstrations and riots in China.
Derailment Accident	Derailment accident of Japanese train in Amagasaki (4/25).
Earthquake, Fukuoka	Major earthquake at Fukuoka, Japan (3/21).
Earthquake, Sumatra	Major earthquake at Sumatra (3/28).
Expo 2005	World Exposition 2005 held in Aichi, Japan.
Kokudo	Arrest of Kokudo president Yoshiaki Tsutsumi.
Livedoor	The M&A of Livedoor, a Japanese IT company, and Fuji TV.
North Korea	Stories about North Korea related incidents.
Pirate	Pirate assault of Japanese tug boat in Malaysia.
Pontiff	Death of Pope John Paul II (4/2), election of Pope Benedict XVI (4/19).

Table 4.2: Description of labeled stories in experiment data set.

previously mentioned, our experiments have been conducted including only the report segment of each story. Imposing this restriction on our experiment data set leaves us with approximately 309 hours of video, spread over 487 shows, and divided into 7944 stories. A summary of the ten dominant topics, representing 1907 stories in total, is shown in Table 4.3.

Before we could start our experiments, we had to solve a few problems with our data set, as there were discrepancies between the automatic shot segmentation and the manual story segmentation. This made deciding which shots that belonged to which story, or particularly which shots represented the first and last shot of a story, difficult. We solved this using restrictive rules for inclusion: it is better to exclude a relevant shot than to include a potentially non-relevant shot, mainly because the EMD would produce "false" results if we would happen accidentally include studio shots. Another problem was with oversegmentation due to camera flashes in news conferences. This resulted in stories with several extremely short shots, which should have been only one. We resolved this issue by merging all sequential sub-second shots up to a duration of

Topic	Duration	Shows	Stories
Anti Japan Demonstration	26 days	111	260
Derailment Accident	6 days	55	394
Earthquake, Fukuoka	30 days	37	68
Earthquake, Sumatra	51 days	41	72
Expo 2005	57 days	62	78
Kokudo	41 days	58	88
Livedoor	59 days	286	584
North Korea	59 days	129	194
Pirate	34 days	56	88
Pontiff	23 days	40	81
Non-labeled stories	62 days	487	6037

Table 4.3: Summary of labeled stories in the entire data set (March to May 2005): *Duration* is the time between first and last appearance. *Shows* is the number of shows in which the story appears and *Stories* is the total number of appearances.

Topic	Duration	Shows	Stories
Anti Japan Demonstration	21 days	28	131
Derailment Accident	6 days	11	79
Earthquake, Fukuoka	30 days	14	29
Earthquake, Sumatra	40 days	15	30
Expo 2005	54 days	24	29
Kokudo	39 days	11	20
Livedoor	48 days	52	123
North Korea	59 days	15	22
Pirate	8 days	13	32
Pontiff	22 days	14	40
Non-labeled stories	62 days	103	1272

Table 4.4: Summary of labeled stories in the NHK data set (March to May 2005).

one second. By doing this we reduced the effect oversegmentation had on the efficiency and results of the EMD calculation.

4.2 Method

To test the efficiency of the design, we have implemented a simulation of our adaptive topic tracking system using C++. We implemented the system using the Xerces-C++ parser library to access our annotated data and the FFmpeg library to extract keyframes from the video. Yossi Rubner's EMD code [8] served as the foundation of our EMD implementation. It computes the EMD between two weighted arrays of features. For the purpose of feature extraction, we quantize the HSV color space into 256 distinct color sets: hue quantized into 16 bins, saturation and value into 4 bins respectively.

The first chronological appearance of a labeled story was used to initialize the system. If inspection showed that the keyframes of this story failed to summarize the story, the next appearance was chosen instead. User feedback was simulated using the provided topic story labels, and was given after every individual story processed by the system. This system was subjected to two separate types of experiments:

Single network : Using a subset of the database, namely all the stories from the NHK network, we ran experiments to discover trends and to determine optimal parameters for our system. The NHK data set covers approximately 100 hours of video, spread over 103 shows with 1807 stories. A summary of these stories can be found in Table 4.4. The tracking system was first tested with the following parameters: Topic model size $N = \{1, 3, 5, 7\}$, Similarity threshold $S = [0.50 \,..\, 0.60]$, and Time-out period $T = \{1, 3, 5, 7\}$ days.

Multiple networks : We ran additional experiments on the entire database, composed of multiple networks, to test the robustness of the system. In particular we were interested in how the system behaved with a larger data set and how the results were affected by using init stories from the different networks.

4.3 Evaluation measures

The results of the experiments are evaluated by conventional measures of information retrieval: precision and recall. We used the following definitions:

A result is every story which meets the threshold S for the EMD comparison. The ground truth for a given topic consists of all the stories, after the original story, with an identical topic label. In [9] the concept of a "somehow relevant" hit is introduced to boost the results, e.g., a summary of all the stories of a show would be considered a match to all of the stories. We do not agree; a story is either relevant or not. If a potential result has a label which does not exactly match the given topic, it is considered a false alarm. This strict limitation should be noted when comparing our results.

Based on these measures we can calculate precision and recall after the system has finished running. Furthermore, we calculated the F-measure, as a harmonic mean of precision and recall, based on the following formulas:

$$P = \frac{\text{number of relevant results}}{\text{number of results}} \tag{4.1}$$

$$R = \frac{\text{number of relevant results}}{\text{ground truth}} \tag{4.2}$$

$$F = \frac{2PR}{P+R} \text{ , where } P \text{ is precision and } R \text{ is recall} \tag{4.3}$$

Chapter 5

Results

In this chapter we present the results from our experiments. The implementation of our tracking system was subjected to two different types of experiments. At first we ran experiments with a subset of the data set to discover trends and to determine trends regarding the parameters of our system. These results can be found in Section 5.1. The results from tracking in the full data set can be found in Section 5.2.

5.1 Tracking in a single network

As explained in Chapter 4 all the experiment data was subjected to keyframe extraction and EMD calculation before running the simulations, in order to facilitate running multiple experiments with different parameters. The stories from the NHK subset, summarized in Table 4.4, was used for this purpose and the stories were compared with all stories aired at a later date. The EMD values for this data set, i.e., the results from Equation (3.8), are normally distributed with a mean of 0.654 and a standard deviation of 0.088. A histogram plot of the distribution can be found in Figure B.1 of Appendix B. We proceeded to evaluate the system to discover which parameters lead to the best results.

Using a similarity threshold of 0.50 causes the system to fail tracking 4 out of the 10 stop stories, due to being too restrictive. A threshold that low will only match a little more than 5% of the material. We therefore exclude this threshold from our results. The mean F-measures for $S = \{0.55, 0.60\}$, $N = 7$ and $T = 1$ are 0.23 and 0.22 respectively, with increasingly higher recall (0.39, 0.50) and lower precision (0.18, 0.15). In order to show the trends for an increasing similarity threshold, the mean precision, recall and F-measure for the thresholds $S = [0.51 \; .. \; 0.60]$ are plotted in Figure 5.1. It can be seen, that increasing the threshold leads to increasing recall at the cost of decreasing precision. However, the number of returned results outgrows the number of correct results, which can be seen in the increasingly poor precision. The trend is the same for all N and all T in our experiments. By observing the F-measure, we can see that the increase in recall does not weigh up the decrease in precision. We definitely want to put emphasis on a high precision, since our system relies on user feedback from the list of accumulated results. If the list becomes uncontrollable in size, no user could be expected to provide the necessary feedback. Hence, $S = 0.53$ appears to provide an optimum threshold for our data set and the individual results further enforce this assumption.

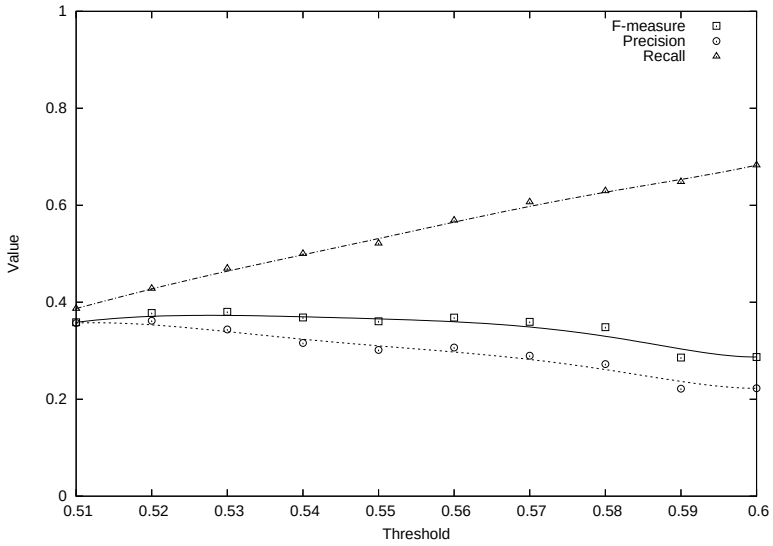


Figure 5.1: Examining trends for decreasing or increasing the threshold.

Similarly, increasing the time-out value results in increasing recall and decreasing precision. $T = \{1, 3, 5, 7\}$ with $S = 0.55$ [1] and $N = 3$ results in the following F-measure means: 0.23, 0.19, 0.17 and 0.16 with steadily decreasing precision (0.21, 0.17, 0.16 and 0.15). Using a long time-out period might reduce the chance of ending the news story tracking prematurely, but it also extends the time we continue to track a canceled story. Major news stories, as the ones we are interested in tracking, are very likely to appear on a daily basis until they are canceled, which is why $T = 1$ show the best results.

The most interesting parameter turns out to be the size of the topic model, N. A topic model consisting of a single story means that only the last correctly detected story is used to find the next matching story. This discards all references to older footage in the topic model and leads to very poor results; the mean F-measure is as low as 0.10, with 2 stories failing to be tracked at all (F-measure = 0.00). The results for $S = 0.55$ [2], $T = 1$ and $N = \{3, 5, 7\}$ can be seen in Table 5.1. We can see different trends in the precision, recall and F-measure as N changes. Some of the stories give a better result when tracked with a larger model, and some with a smaller one. However, it appears as if $N = 5$ provides a stable middle ground in most of the cases and in the case of the *Pirate* story it produces the best result. Comparing the results of Table 5.1, where *Derailment Accident* is in the top, with the *Shows* column of Table 4.4, we can see that the more frequently a story appears, the better results we get from a high N. With less frequent stories, the opposite appears to be true. We have yet to find a conclusive reason to this - it could simply be that a bigger topic model leads to a wider definition of the scope of the story. That would explain why more frequent stories, which represent

[1] S = 0.53 was not included in the study of the time-out value, T.

[2] S = 0.53 was not included in the study of the topic model size, N.

Topic	N	P	R	F
Anti Japan Demonstration	3	0.38	0.52	0.44
	5	0.35	0.62	0.45
	7	0.35	0.71	0.47
Derailment Accident	3	0.78	0.44	0.56
	5	0.63	0.56	0.59
	7	0.60	0.66	0.63
Earthquake, Fukuoka	3	0.18	0.50	0.26
	5	0.17	0.54	0.25
	7	0.15	0.54	0.24
Kokudo	3	0.21	0.44	0.29
	5	0.15	0.50	0.23
	7	0.14	0.56	0.22
Livedoor	3	0.19	0.44	0.26
	5	0.16	0.48	0.24
	7	0.17	0.59	0.27
Pirate	3	0.14	0.31	0.20
	5	0.18	0.54	0.27
	7	0.17	0.54	0.26

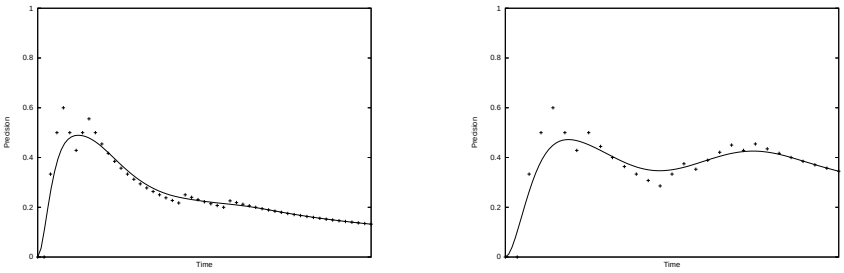
Table 5.1: Results with $N = \{3, 5, 7\}$, $S = 0.55$ and $T = 1$.

Reason	Topic	P	R	F
Too general	Expo 2005	0.03	0.07	0.04
	North Korea	0.10	0.15	0.12
Dual events	Earthquake, Sumatra (both, 3/2)	0.03	0.03	0.03
	Earthquake, Sumatra (single, 3/22)	0.14	0.57	0.22
	Pontiff (both, 4/2)	0.03	0.08	0.05
	Pontiff (single, 4/19)	0.13	0.36	0.20

Table 5.2: Results for failed stories, along with the reasons ($N = 5$, $S = 0.55$, $T = 1$). Improved results, after modified initialization date, shown for the *Dual events*.

a larger portion of the news feed, benefits from an increasing N.

Four stories were omitted from Table 5.1, due to their poor results (F-measure $<=$ 0.12). It was important for us to analyze the reasons behind the failed results, as they could provide interesting pointers on how to the results could be improved. Inspection shows that these stories are not individual stories, but rather groups of stories. Further analysis shows that the failed stories can be divided into *too general* stories and stories which concern *dual events*. The *Expo 2005* topic is a loosely connected group of stories concerning the different events and displays of the World Exhibition held in Aichi and the *North Korea* topic is similarly covering a wide range of stories involving North Korea in some way. The dual events are not one but two stories, with a period of silence in between: the *Pontiff* topic is split up into the death of Pope John Paul II and the election of Pope Benedict XVI, and the *Earthquake, Sumatra* topic is split up in reports on the reconstruction after the earthquake of December 2004 and the new earthquake of March 2005. We ran additional experiments to track these stories, using only the second of the two events. As can be seen in Table 5.2, the results improved significantly after this change. The results for the *too general* stories could not be



(a) Kokudo (b) Pirate

Figure 5.2: Plotting precision over time for (a) Kokudo and (b) Pirate.

Topic	P	R	F
Anti Japan Demonstration	0.40	0.57	0.47
Derailment Accident	0.64	0.51	0.57
Earthquake, Fukuoka	0.20	0.54	0.29
Kokudo	0.13	0.39	0.20
Livedoor	0.20	0.42	0.27
Pirate	0.34	0.45	0.39

Table 5.3: Final results with optimal parameters ($N = 5$, $S = 0.53$, $T = 1$).

improved in this way, due to being too scattered in their contents.

In an attempt to visualize the tracking of a story, and at the same time show what happens when the topic model fails to update in accordance to the changes in the story, we provide Figure 5.2. In this figure we plot the changes in the precision over time for two of the stories, *Kokudo* and *Pirate*. In the case of the *Kokudo* story, we begin tracking the story and, as can be seen from the increasing precision, we retrieve more and more correctly matched stories from the news feed. However, the precision drops quickly and, excluding two more hits which can be seen from the singular increases in precision in the middle of Figure 5.2 (a), we loose track of the story and cease tracking. The *Pirate* story, in Figure 5.2 (b), suffers a similar drop in precision as the story changes focus. The difference here is that we successfully include this new development in the topic model and stay on track for the duration of the story.

Overall the experiments show that the proposed method is capable of achieving news story tracking over time. The final results of the tracking on the NHK data set can be seen in Table 5.3. These results were achieved with the optimal parameters $S = 0.53$, $T = 1$ and $N = 5$. Increasing either of the parameters leads to higher recall at the cost of lower precision. Although there are some types of topics that fail to be tracked, it is to too wide topics rather than to the method itself. We also note that we can improve the results, by a more careful selection of init stories.

5.2 Tracking in multiple networks

After performing the experiments on the smaller data set we wanted to see how the system would behave in the entire data set, composed of all ten programs from the five

Topic	Init story	P	R	F
Anti Japan Demonstration	ASAHI #1	0.15	0.65	0.24
	FUJI #1	0.09	0.59	0.15
	NHK #2	0.11	0.66	0.20
	NTV #1	0.15	0.66	0.25
	TBS #2	0.15	0.64	0.24
Derailment Accident	ASAHI #1	0.57	0.46	0.51
	FUJI #1	0.56	0.47	0.51
	NHK #1	0.55	0.49	0.52
	NTV #1	0.54	0.48	0.51
	TBS #1	0.55	0.47	0.51
Pirate	ASAHI #1	0.10	0.48	0.17
	FUJI #1	0.10	0.43	0.16
	NHK #2	0.08	0.43	0.14
	NTV #1	0.09	0.41	0.14
	TBS #1	0.09	0.42	0.15

Table 5.4: Results for tracking in the entire data set with init stories from different networks ($N = 5$, $S = 0.53$, $T = 1$).

different channels. The same method to perform keyframe extraction and EMD calculation was used with this larger data set resulting in EMD values, which were normally distributed with a mean of 0.627 and a standard deviation of 0.087. A histogram plot of the EMD distribution for the entire data set can be found in Figure B.2 of Appendix B. That the mean distance was lower in the case of all different networks might be considered a surprise. Intuitively, including stories from many different networks should result in an increasing mean distance because of greater variation of content, but this was not the case.

We ran our experiments using the top three stories from the single network experiment: *Anti Japan Demonstration*, *Derailment Accident*, and *Pirate*. These three topics were subjected to tracking using an init story selected from each of the five networks. This gave a good idea about the robustness of our system. Although the results in themselves may not look that impressive, let us review the conditions of this experiment:

- We are now tracking in an extremely large data set. Every day our system is processing an average of 161.4 stories in almost 5 hours of news programs.
- We are no longer using optimal parameters. The optimal parameters were discovered by running extensive experiments on the NHK data set. The entire data set has a different distribution mean of the EMD values, which will inevitably lead to worse results.

Despite these limitations we are able to produce results for the three selected stories. The recall is the same or higher compared with the results from the single network experiment, but the precision is lower. The reason for this can be found in our analysis of the threshold parameter, S, in Section 5.1 and particularly in the trends shown in Figure 5.1. By increasing the threshold we gained higher recall at the cost of lower precision. The gain in recall, however, was not enough to compensate for the loss of precision. Since the entire data set has a lower mean, we draw the conclusion that the similarity threshold $S = 0.53$ is now too high. Regrettably, due to lack of time it was

not possible for us to run the same extensive experiments on the entire data set in the scope of this thesis. It would be very interesting to see what kind of results could be achieved with an optimal threshold.

We fully expected our results in the multiple network scenario to be worse than in the single network scenario, due to the increased amount of noise causing more false alarms. What should be noted, however, is that the system is robust in the sense that we can pick init stories from either of the networks and receive similar results in tracking. This enforces our original assumption; the semantic information contained in the keyframes of these stories are enough to perform tracking across multiple networks, even though the flow from the news feed to the system more than quadrupled in quantity.

Chapter 6

Discussion

Although various tracking systems using visual and language information have previously been designed, our method is unique because the nature of the query is adaptive. Instead of a simple query based system for retrieval out of a database, we provide an online system which can be set to watch any news stream and track the topics flagged by a user. The advantages of our system is that it requires no training and that it relies on low-level video features. It is therefore suitable for use in an online tracking environment, where real-time requests to handle incoming data come from the user.

We have conducted our evaluation with a strict idea about truth. The semantic context of a story has to match the tracked topic exactly to be considered correct. Despite this strict definition, we are able to track several types of news stories, using only visual features. Studying the results, it is clear that we succeeded in significantly reducing the amount of irrelevant stories presented to the user. We track in a vast pool of data, but still managed to achieve a mean precision of 0.31 at the same time as we maintained a relatively high degree of recall, between 0.39 and 0.57. Since the nature of the involved stories changed significantly over the recorded period, we regard this as a success. In contrast to this, we noticed that there are some types of stories which we could not track. The failed stories from our experiments were analyzed and put into two groups: the *too general* topics and the *dual events* topics. In Section 5.1, we showed that the results for the topics in the later group could be improved by limiting the tracking to one of the included events. The first group, however, remains too scattered to track. It is also clear that our system relies on tracking major "hot" topics and cannot be put to use on stories which appear with too low frequency.

In our initial investigation of what provides a good story representation, we argued that visual features should be able to provide semantic linkage between stories. This appears to be true, but the quality of the keyframes is very important. The system currently takes a naive approach to keyframe extraction, by simply using the middle frame of every shot. An inspection of a subset of the extracted keyframes show that, even though it is a successful approach in most of the cases, a portion of the keyframes have little or no value. Being able to eliminate redundant shots in the init story, or using more advanced keyframe selection method, should definitely lead to better results.

What comprises a good init story is still somewhat hard for us to quantify. The results of our experiments are inconclusive in regards to correlation between the results and the number of shots included in or the length of a story. However, we did notice that init stories which failed to summarize the desired topic visually did lead to worse results. Since we rely on the keyframes to track the topic, it is important that the

init story contains as much information as possible. Using a short "highlight" story to initialize tracking fails to capture the full semantic context of the topic. It is also important that the init story does not contain irrelevant shots. Fortunately, our design relies on the user to pick the init story, so it should be possible to manually exclude shots which can be deemed irrelevant to the desired topic.

The trends surrounding the parameters we used to modify the behavior of our system are interesting. In regards to the time-out period, T, we could quickly draw the conclusion that our stories appear frequently enough for an automatic time-out of a single day. This means that if a matching story does not appear in any of the channels for the duration of a full day, we stop tracking it. Although this parameter was important for our simulation experiments on the system, a real implementation could rely on the user to decide when to stop tracking: either when the user is no longer interested in the story or when it is obvious from the context of the story that it has been canceled. Leaving the choice of this parameter to the user, would be a wise choice.

Adjusting the similarity threshold, or the highest EMD value between two stories for them to be considered matching, has great impact on our results. If we choose a too high value for S we will return far too many results to the user, and a too low value we will return too few. Our experiments show that we need to adjust the threshold value according to the distribution of the data set we want to track in. Although the characteristics of the distribution cannot be known before we commence tracking, we can tune the threshold as more and more stories have been handled by the system. Values at one or more standard deviation below the mean appear to be good candidates. Our experiments on the NHK data set lead to an optimal threshold of 0.53, which is 1.41 standard deviations ($1.41 * 0.088$) below the mean of 0.654. In a real scenario we cannot calculate an optimal similarity threshold in advance. Instead it would be constantly trimmed using the assumption above and the mean and standard deviation of all the stories investigated by the system up to this point.

Despite varying the model size, N, appeared to give different results depending on how frequently the story appears, we were able to find an optimal value in 5 for our NHK data set. A small topic model will cause us to loose references to older material, which might cause us to miss stories. If we set N too high, the scope of our tracking will be too wide. It would be interesting to investigate if the optimal model size depends on the amount of stories or networks in the news feed. Our simulation system currently gives user feedback after every story, and not every day as would be more appropriate. Changing this will also change the optimal model size.

Further investigating the behavior of the parameters in the entire data set is an important task for future work. With our current video database there is a vast information to dig into. Closer analysis needs to be done on the stories used for initialization, as well as the ones only being used for tracking. One interesting task would be to investigate if the topic model converges even though we use different init stories. Another, more radical idea, is to change the way the topic model is composed. Instead of using a FIFO list, we could extend the way we model our story representations to modeling a topic representation, i.e., a weighted graph comprised of the shots of all the stories included in the topic model. This would radically change the behavior of our system, but would result in a more efficient way to calculate the similarity of new stories entering the system, as they would only be compared once against the topic model representation instead of N times against the stories of the topic model list.

Chapter 7

Conclusions

In this thesis we have described a system to provide adaptive tracking of news stories based on the Earth Mover's Distance (EMD). A single on-topic story is chosen by a user to initialize the system. This story is added to a topic tracking model, against which new stories entering the system are compared to judge their relevance to the tracked topic. The results are supervised by the user, and the correctly tracked stories are used to adapt the topic model to the changes in the story. The behavior of the system can be modified through three parameters: the topic model size, the similarity threshold, and the time-out period.

We have implemented a simulation of the system and have run extensive experiments on a video database composed of approximately 530 hours of news programs from five Japanese broadcasting networks. With our experiments we have shown the feasibility of our system and investigated which parameters produce the best results.

Our work has yielded a deeper understanding of the problem of tracking news stories and a substantial experimentation platform on which to continue exploring the design of a system for adaptive tracking of news stories. Our implementation has shown that it is possible to track news stories using only visual features. The future direction of our work will be to investigate how the system can be improved and to further analyze the results of our experiments. A long term goal is to couple the system with the story segmentation methods presented in in [12] in order to achieve an even more autonomous system.

Acknowledgments

I would like to thank all the members of the Text Information Processing group of KDDI R&D Laboratories in Fujimino, Japan. In particular thanks goes to Keiichiro Hoashi, who supervised my work and introduced me to the world of video information retrieval, and to Tadashi Yanagihara, who has guided me through my daily life and provided inspiration for many interesting discussions.

I also want to thank my brother, Björn, for visiting over the New Year, and my girlfriend Virginie Delporte, for our wonderful vacation in the fall of 2005 and her never ending support from afar - I am coming home soon, I promise.

This work was partially funded by a scholarship from the Sweden-Japan Foundation (http:www.swejap.a.se).

Appendix A

Annotation data (MPEG-7)

This is an example of the annotation data used to describe the story segmentation and labeling. The annotation is in XML format and follows the MPEG-7 standard [13]. A description of the elements used in our experiments follows:

- *VideoSegment* annotates a story, or video clip.
- *MediaTimePoint* is used for the start time of each video clip.
- *FreeTextAnnotation* is used to label:
  - Story segment type: introductory, VTR (report) , weather, sports.
  - Story topic: Anti Japan Demonstration, Derailment Accident, etc.

Below is an excerpt of the annotation data file for the NHK #1 show of 4/2 2005:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaLocator>
          <MediaUri>20050402_01_NHK.mpg</MediaUri>
        </MediaLocator>
        <TemporalDecomposition>
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>Story Boundary</FreeTextAnnotation>
              <FreeTextAnnotation>Program Start</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:29:27000F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>Story Boundary</FreeTextAnnotation>
              <FreeTextAnnotation>Introductory</FreeTextAnnotation>
              <FreeTextAnnotation>Society</FreeTextAnnotation>
            </TextAnnotation>
```

```
            <MediaTime>
              <MediaTimePoint>T00:01:50:12300F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>VTR</FreeTextAnnotation>
              <FreeTextAnnotation>Society</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:02:10:22800F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>Story Boundary</FreeTextAnnotation>
              <FreeTextAnnotation>Introductory</FreeTextAnnotation>
              <FreeTextAnnotation>Society</FreeTextAnnotation>
              <FreeTextAnnotation>International</FreeTextAnnotation>
              <FreeTextAnnotation>Pontiff</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:07:32:12600F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>VTR</FreeTextAnnotation>
              <FreeTextAnnotation>Society</FreeTextAnnotation>
              <FreeTextAnnotation>International</FreeTextAnnotation>
              <FreeTextAnnotation>Pontiff</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:07:51:29100F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
```

[Cut here to be able to show the end of the file.]

```
          <VideoSegment>
            <TextAnnotation type="scene" relevance="1" confidence="1">
              <FreeTextAnnotation>Story Boundary</FreeTextAnnotation>
              <FreeTextAnnotation>Weather_End</FreeTextAnnotation>
              <FreeTextAnnotation>Program End</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:30:35:15000F30000</MediaTimePoint>
              <MediaIncrDuration mediaTimeUnit="PT1001N30000F">0</MediaIncrDuration>
            </MediaTime>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Appendix B

EMD distribution plots

B.1 NHK data set

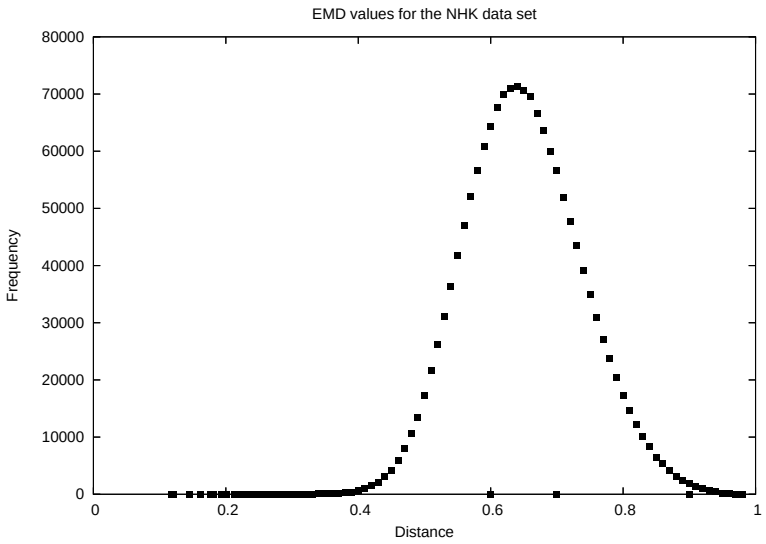


Figure B.1: Histogram plot of the EMD values for the NHK data set: normally distributed with mean 0.654 and standard deviation 0.088. The plot shows the results of comparing all 1807 stories to all stories appearing at a later date in the data set giving a total of 1633528 results.

B.2 Entire data set

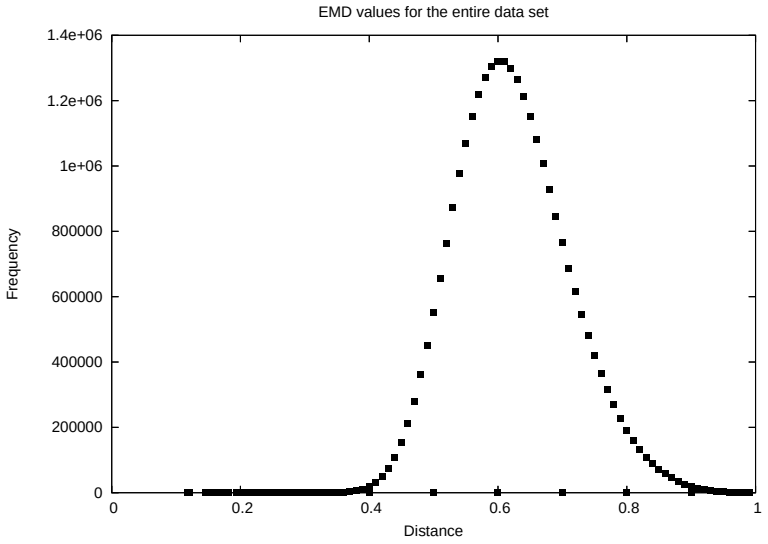


Figure B.2: Histogram plot of the EMD values for the entire data set: normally distributed with mean 0.627 and standard deviation 0.087. The plot shows the results of comparing all 7944 stories to all stories appearing at a later date in the data set giving a total of 31557540 results.

Bibliography

[1] S. Cheung and A. Zakhor, "Fast similarity search and clustering of video sequences on the world-wide-web," *Multimedia, IEEE Transactions on*, vol. 7, no. 3, pp. 524–537, 2005.

[2] Junsong Yuan, Ling-Yu Duan, Qi Tian, and Changsheng Xu, "Fast and robust short video clip search using an index structure," in *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, New York, NY, USA, 2004, pp. 61–68, ACM Press.

[3] Liping Chen and Tat-Seng Chua, "A match and tiling approach to content-based video retrieval," in *ICME*, 2001.

[4] Yi Wu, Yueting Zhuang, and Yunhe Pan, "Content-based video similarity model," in *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, New York, NY, USA, 2000, pp. 465–467, ACM Press.

[5] Anil K. Jain, Aditya Vailaya, and Xiong Wei, "Query by video clip," *Multimedia Syst.*, vol. 7, no. 5, pp. 369–384, 1999.

[6] Yuxin Peng and Chong-Wah Ngo, "Clip-based similarity measure for hierarchical video retrieval," in *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, New York, NY, USA, 2004, pp. 53–60, ACM Press.

[7] Yuxin Peng and Chong-Wah Ngo, "Emd-based video clip retrieval by many-to-many matching," in *CIVR*, 2005, pp. 71–81.

[8] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[9] Yun Zhai and Mubarak Shah, "Tracking news stories across different sources," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, pp. 2–10, ACM Press.

[10] Ichiro Ide, Hiroshi Mo, Norio Katayama, and Shin'ichi Satoh, "Topic threading for structuring a large-scale news video archive," in *CIVR*, 2004, pp. 123–131.

[11] Chong-Wah Ngo, Ting-Chuen Pong, Roland T. Chin, and HongJiang Zhang, "Motion-based video representation for scene change detection," in *ICPR*, 2000, pp. 1827–1830.

[12] Keiichiro Hoashi et al., "Video story segmentation and its application to personal video recorders," in *CIVR*, 2005, pp. 39–48.

[13] Thomas Sikora, "The mpeg-7 visual standard for content description-an overview," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 11, no. 6, pp. 696–702, 2001.

[14] Keiichiro Hoashi et al., "Shot boundary determination on mpeg compressed domain and story segmentation experiments for trecvid 2004," in *TRECVID*, 2004, pp. 39–48.