UPTEC IT08 004

Examensarbete 30 hp
Februari 2008

Shot Selection Strategies in Video News Story Tracking

Mattis Fjällström

UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Shot Selection Strategies in Video News Story Tracking

Mattis Fjällström

When tracking a news story a user typically chooses a story as input for a query. The computer system then tries to find stories similar to the one used as input. Earlier attempts at video news tracking used all shots in a story, regardless of whether they improved the query or not. Some of those shots are bound to be disruptive to the matching process.

Due to this problem with the current methods, we propose a shot selection method which utilizes the similarity between shots within a story to automatically extract shots that are highly representative of the story content. Specifically, three methods are proposed: a) similar shot exclusion; b) similar shot inclusion; and c) length based shot selection. This paper describes experiments performed on Japanese broadcast news video. We find that being restrictive with what shots to include can improve tracking performance. In certain cases, drastically.

Handledare: Keiichiro Hoashi
Ämnesgranskare: Ivan Christoff
Examinator: Anders Jansson
ISSN: 1401-5749, UPTEC IT08 004
Sponsor: KDDI Research Labs, Saitama, Japan

Tryckt av: Reprocentralen ITC

Layman's Introduction - in Swedish

Då och då ser vi alla någonting på nyheterna som av en anledning eller annan fångar vårt intresse. Det kan handla om de så kallade "tsunami-banden" eller Estonia-färjan, om en katt som sitter fast i ett träd eller om ett krig någonstans långt borta. Vi är nyfikna, vill veta mer. Varje ny detalj är intressant och vi vill inte missa någonting.

Tills för några år sedan var det här mest en fråga om att informationen skulle nå *fram* till oss, via TV- eller radio-nyheterna. För de lite mer tålmodiga kom nyheter via morgontidningen. Problemet för den enskilda individen var då att informationen inte alltid nådde fram i den utsträckning som han eller hon önskade - att tillgodogöra sig det som nådde fram var inte något problem.

I och med dagens informationsteknologiska värld är problemet ett annat. Med kabel-, digital- och satellit-tv har vi massvis av nyhetskanaler. Internet förvandlas allt mer från ett medium som domineras av text till ett fyllt av bilder, ljud och numera video. Problemet, för den som vill följa en story, brukade vara att hitta information - idag är det att att hitta *rätt* information och inte ödsla tid på irrelevanta nyheter.

Vi får all denna data från allt fler källor. Där vi förr bara hade TV och radio, har vi nu också datorer och mobiltelefoner. Vi översköljs av information. Hur kan man följa en story i detta intensiva bakgrundsbrus av nyheter?

Ett sätt är att ta datorn till hjälp. Om vi kan programmera en dator till att känna igen de nyheter vi är intresserade av, så behöver vi ju faktiskt inte se allting - datorn kan titta "åt oss", och spela in de nyheter som vi vill se. Vi, datorns användare, kan sedan "vittja" den och se de nyhetsklipp som intresserar oss.

Hur kan man få datorn till att se på nyheter åt oss? Till viss del handlar det om sökning i video, vilket varit ett "hett" ämne i flera år nu. Men bara till viss del. I princip alla video-sökmetoder som föreslagits till dags dato baseras på att genomsöka en databas. När en dator följer nyheter fungerar det hela lite annorlunda. Video-klippen är ordnade i tiden, och konsumenten är bara intresserade av de allra senaste. Denna begränsning efter tidsaxeln finns inte i vanliga fall. Vidare måste vår sökmetod vara snabb - vi måste hinna med all inkommande data i realtid.

Det finns flera olika sätt att angripa det här problemet. Olika forskningsgrupper har valt olika angreppsmetoder. Somliga fokuserar på ljud, tal, och det som sägs i video-klippen. Somliga söker känna igen ansikten i bilderna, medans andra letar efter texter i bilden. System som märker bilder med taggar hör dock inte riktigt till det här fältet - det kräver förändringar även hos de som sänder bilderna, inte bara hos mottagaren, och dessutom måste ju någon sätta dit alla dessa taggar och göra det på ett sätt som är intuitivt för alla de som skall söka efter innehåll i videoströmmen. Dessutom skulle det vara lätt att "fejka" innehåll för att få fler tittare.

KDDI Labs grupp för text och informationsbehandling har valt som "sitt" forskningsområde bildernas färginnehåll. Under föregående år undersökte och utvecklade Mats Uddenfeldt, också från Uppsala Universitet, ett sätt (1) att jämföra två nyhetsinslag baserat på färginnehåll i enskilda bilder och en jämförelsealgorithm som baserades på den s.k. "Earth Movers Algorithm" (2).

Detta arbete bygger vidare på (1) och undersöker möjligheter att smartare välja bilder ur nyhetsinslagen. Hypotesen är att bilder som inte bidrar positivt till jämförelsen kan sorteras bort på förhand, och på så sätt både snabba upp jämförelseprocessen och förbättra resultatet.

Contents

1 Introduction

1.1 Background

The first thing that appeared online was text. Then came images, then sounds. During the last couple years video has become a common feature of the Internet. One extremely popular website which allows users to upload and view a plethora of short video clips is Youtube. Many TV-stations are making their productions available online. New protocols such as Bittorrent are making downloading of large video files feasible in a way it was not before.

There is a parallel sea change in electronics, giving people the ability to store some of all the data washing over us. Personal video recorders, computers and portable harddrives. DVD-burners. And more. In todays world, a person has access to more media than ever before.

Taken together, these two trends mean that large amounts of video is available in a digital format. There is little reason to believe this trend will subside. Users now face a problem - how to make sense of all this information. An even more basic problem is how to find what they are looking for among all that data.

1.2 General Problem

One subset of this larger problem is tracking news stories. A user scenario is as follows: User sees news story on TV. He wishes to know when more information is available on this topic. However, watching all the news available is not an option - there is just too much being broadcast. Instead, the user should be able to instruct his computer to look for news stories similar to the one he just watched.

This, then, is the problem - given one news story, how to find others like it?

1.3 Video News Tracking

Video News Tracking is a term used to indicate "automatically following a news story from several news outlets over time". Given one news story as input, the news tracking system strives to identify others like it.

In (1), the authors proposes a user feedback adaptive system to track news stories using the Earth Movers Distance algorithm (in the rest of this paper, this will be referred to as EMD) for story to story comparisons.

Each story is composed by a number of shots. From each shot the authors take one frame, from which a histogram is extracted. Thus each story is thus abstracted as a set of histograms.

This representation is compared to subsequent stories using the EMD algorithm. Whenever the distance between the two is found to be below a certain limit (the "EMD Limit"), the new story is considered a match and is presented to the user. The user then provides feedback on whether this was a correct match or not - if not, the story is discarded, if it is, the story is added to the query, and the process is repeated for the next story.

In (1), the authors show that the EMD is a viable metric and highly applicable to the problem at hand, but they also indicated some problems with a video news tracking system that needs to be addressed.

1.4 Specific Problem

With this approach to tracking a news story, the first step is to extract a set of keyframes from the story - one from each shot. Histogram features are then extracted from these frames. This gives a set of histograms (one per shot) representing the story. The same process is applied to each story.

In order to compare two stories to one another, we use the Earth Movers Distance (1, 2). In shot to shot comparisons, we use the regular L1-distance (1).

In (1), the authors used every shot when comparing stories to each other. However, some shots might be better representatives for the story than others - for example, in a story about an earthquake, it is likely that there will be plenty of images depicting dirt and collapsed buildings. Not so in a story about finance. This means that given one story with a set of shots, some will provide better matches than others with other stories on the same topic.

In this paper, we show that under certain conditions, selection of shots before automatic story to story comparison gives better matches and hence improves tracking results.

For example, look at figure 1 and figure 2. These images show the shots included in the first and second Pontiff story, respectively. The system starts out with the first one, and should detect that the second one is about the same topic. There are some similar shots, but there are also several irrelevant shots.

2 Shot Selection Methods

This section outlines three different approaches for filtering shots, and the hypotheses underlying them.

2.1 Longest Shot

This approach - assuming that the longest shot in a story is the most significant - means using only that shot while tracking stories. This leads to a significant reduction in the number of calculations needed, and as a result the tracking should run faster. If the tracking performance is equal or only slightly lower, this might be a price worth paying for lower demands on the hardware.

In order to get the best possible performance, the filtering is done before extracting images from the shots. This means we will only extract frames from the video clip that we know corresponds to the longest shot.

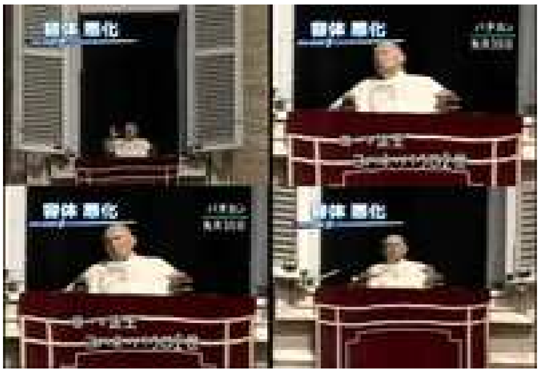

Figure 1: The first Pontiff story, all shots.

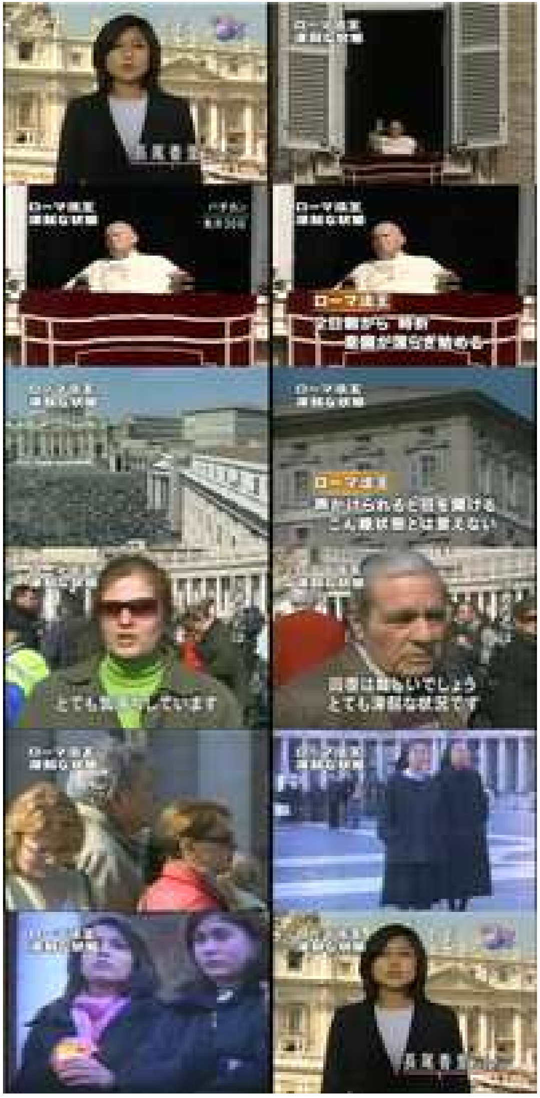

Figure 2: The second Pontiff story, next day, all shots.

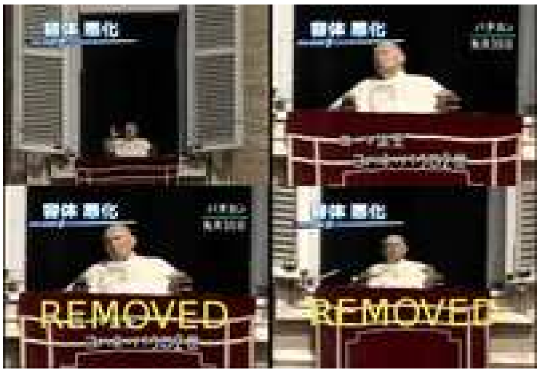

Figure 3: The first Pontiff story, removed shots that were within 0.4 of each other.

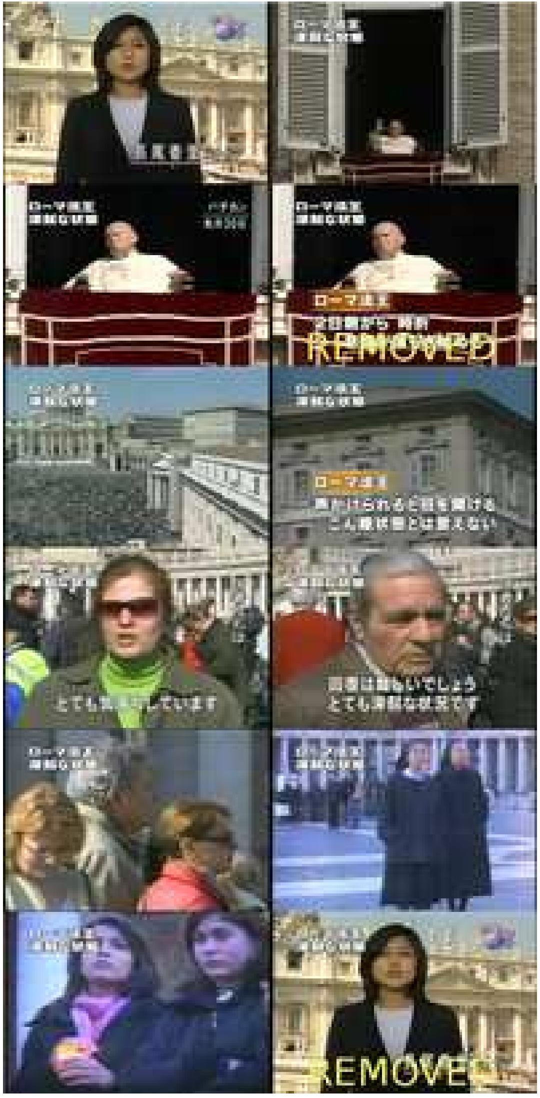

Figure 4: The second Pontiff story, next day. Shots removed for being within 0.4 of each other are marked accordingly.

This reduces both calculation time and memory foot print.

2.2 Similar Shot Exclusion, SSE

This approach eliminates shots that are similar to others in the story. The idea is to avoid duplicates.

Duplicates might detract from the comparison. They do not add new information that's not there already. For a perfect match between two identical stories, they should have the same number of shots with similar looks.

The shot removal algorithm goes through the list of shots that belongs to a story, and compares each with the others. Whenever two shots are within the given limit of each other, we remove the second one.

In figure 3 and figure 4 we have an example of similar shot exclusion, SSE.

2.3 Similar Shot Inclusion, SSI

Now, we postulate another theory. In many news shows, certain shots or images are at the center of the story. E.g., in a story about a plane crash, images of the crashed plane may occur frequently both within the current story and in subsequent reports on the same topic.

If certain shots matches to other stories correctly, and others do not, then it follows that shots within a story can be considered "signal" and "noise". The task then will be to improve the signal to noise ratio.

The algorithm here is very similar to the one used for removing duplicate shots - we do the same sort of iteration and comparison. When a shot has been compared with all other shots, and we found no matches within the given limit, we remove the shot.

There is an extreme case here - when all shots are unique, and no shot looks like any other. In this situation, we would end up without any shots at all. If that happens, we simply use the longest shot within the story as the representative. (This can be seen in the tracking results of some topics - they are virtually identical to those for longest shot tracking).

In figure 5 and figure 6 we have an example of similar shot inclusion. Looking at the similarity between figure 3 and figure 4 compared to figure 5 and figure 6, it seems apparent that the second alternative would result in more successful tracking. However, not all stories are as clean cut as this one, as we shall see.

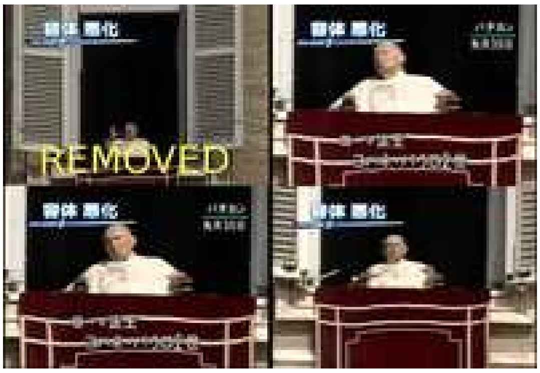

Figure 5: The first Pontiff story. Shots removed that were not within 0.4 of another shot. Shots are marked.

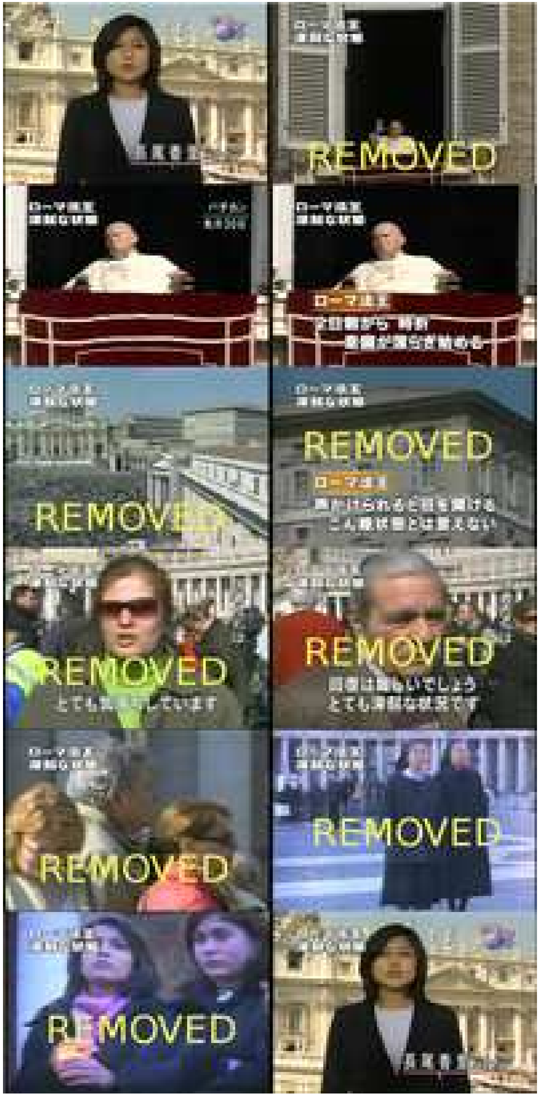

Figure 6: The second Pontiff story, next day. Shots removed that were not within 0.4 of another shot. Removed shots are marked.

3 Methods

3.1 Shot To Shot Metric

In order to be able to compare shots, we need to decide how best to represent those shots. Picking one frame from the middle of each shot and using that image, together with the length of the shot, is a simple but useful abstraction.

The Haar transform is applied in order to arrive at a histogram in the HSV color space. For shot to shot similarity measures, a normalized L1 distance is utilized.

3.2 Story To Story Distance

In order to be able to compare stories we need to solve the same problem - how to get one number that describes the difference between two different stories and qualifies as a metric?

Since each story is represented by a number of shots, and each shot has a weight (its length) and a more detailed description (its histogram), this is a prime candidate for using the EMD, as suggested in (1, 2).

3.3 Earth Movers Distance

To understand the Earth Movers Distance algorithm, picture the following situation.

There are one set of piles of earth, and one set of holes. The question — which the algorithm answers — is this: What is the smallest amount of work needed to fill those holes with earth from those piles? (If there happens to be more holes than piles, one can simply redefine holes and piles before applying the algorithm).

For the purposes of this paper, histograms from shots in one story can be considered holes, and histograms of the shots in another story can be considered piles. This means we can apply the EMD algorithm to find a measure between two different stories.

Further, since the shot to shot cost metric is normalized, the result of EMD will be normalized as well.

For a more in-depth explanation of the workings of EMD, see (6).

3.4 Evaluation Measures

The most used metric for information retrieval is Precision and Recall.

Recall is defined as the number of correctly found stories divided by the total number of correct stories available. A Recall of 1 indicates that all relevant stories were found, while a Recall of 0 indicates that no relevant stories were found. A Recall of 0.5 indicates that half of the relevant stories were found.

Precision is defined as the number of correct stories found, divided by the total number stories found. I.e., a Precision of 1 indicates that relevant stories, and only relevant stories, were found. A Precision of 0.5 says that half the stories found were relevant.

The third standard measure is called the F-measure. It reflects both the Precision and the Recall, thus giving a single metric that can be used to reflect performance of the tracking. F-measure is calculated as shown in equation (1).

$$F = \frac{2 * P * R}{P + R} \qquad (1)$$

$(F = F\text{−}measure, P = Precision, R = Recall)$

However, Precision and Recall has certain problems. Precision is a quota, so if it finds some correct stories, it will never drop to 0 — no matter how badly it performs. Recall has the same problem. Thus, the Precision, Recall and F-measure must be considered together with how often a certain story occurs, and how many other stories are considered. Scores that look rather high can be arrived at by mere statistical probability if a story appears often enough in the news.

In spite of these difficulties, we decided to use these measures. They are, by and large, a standard metric for video database searches.

3.5 Comparing Different Topics

There are some stories that from their first appearance appears so frequently that a user who simply watches everything would reach a "perceived" F-measure of 0.7. This is due to the Recall being 1.0 (the user is not missing anything) and the Precision being around 0.5 (every other story is about this particular topic).

This means that when tracking a topic, it is important to realize why the results are the way they are. Different topics start at different points in time, occur more or less frequently, and are "surrounded" by different stories. Taken together, this means that tracking of each topic is done on its own terms, and the results for tracking one topic cannot be compared to the results of tracking another topic.

As an illustration, consider the Derailment accident topic. From the point this topic is introduced, until the last story in the tracking simulation system, there are 159 stories that the tracking system needs to consider. Out of those 159, 76 are about the derailment accident. In other words - setting the EMD limit to 1.0 and accepting all stories output would give a P value of 0.478 and an R of 1.0. This, in turn, would give an F-measure of 0.6468, which, taken out of context, seems like a rather high score - when in fact 0.6468 should be considered a bare minimum for the tracker while tracking this topic. It reflects the users experience, should he choose not to use an automated tracking system.

Another story, Kokudo, has from its first appearance 1748 other stories to contend with. 21 of those are related to the topic being tracked. Setting the EMD-limit to 1.0 (considering everything a match) gives a Precision of 0.012 (along with a Recall of 1.0). F-measure in this setting becomes 0.0237. Obviously, getting a high F-measure while tracking this story will be much harder than while tracking the Derailment accident.

This means that results for different topics cannot be compared with each other. I will refer to the different settings in which different topics are being tracked as the "topic background".

Different attempts at tracking the same story will, however, occur against the same "topic background" and the results are therefore comparable.

4 Experiments

In order to find out whether only using specific shots improved the results or not, a series of tests were run. The tracking system from (1) was updated with a mechanism that filters shots based on an optional similarity threshold.

For each topic, we first chose a starting story, and then we ran the tracking system as in (1) with this story as input. Since we have data on our video clips and what topic they deal with it is easy for us to compare the results of tracking with the "correct" answers.

This was repeated, using a varying EMD-limit for each run. This gives a set of results depending on what the limit we use to detect similar stories is. This can be plotted as "F-measure" vs. "EMD Limit". For each story, this run was then repeated again and again, but using different shot filter settings each time. All of the resulting curves were then plotted together. The result is set of graphs, one for each topic, which shows the effects of the settings on news story tracking. (E.g. figure 10 for the kokudo story, F-measure as function of EMD limit while keeping similar shots.)

Then the same process was repeated using SSE with different filter settings. Then repeated again, using SSI with different filter settings.

The experiment as described above was run for all stories. In table 1 an overview of the results can be found. Due to space considerations, we have elected not to include all plots in this report.

In the following, general results will first be reported, and then a closer look at one story will be provided.

The data in this paper comes from japanese news broadcasts, from the 1st of March, 2005 until the 30th of April, 2005. Two months of TV programming.

4.1 Results Overview

When reading table 1 it should be understood that "basic" is tracking as performed in (1), using all shots available. This is our reference point, which we are hoping to improve on.

It is then quite apparent that based on our data, SSE does generally not improve tracking results. In 3 cases tracking results improved slightly, in 3 cases the results were worse than basic tracking, and in 2 cases they were basically identical.

It can also be seen that using the longest shot is not a viable approach. In four of the eight topics, the results for longest shot were so bad that they are to be considered meaningless. In two cases the results were viable but really bad, and in one case significantly better than basic. Only one result was really good compared to the basic result.

Similar Shot Inclusion seems to be the most interesting strategy. In five cases out of the eight the results were better than for basic tracking. Two topics got worse results with SSI turned on, and one topic got the same results.

In the following, an analysis will be carried out on the results for one story.

4.2 Case Study Pt. 1 : Kokudo, Similar Shot Exclusion

For closer scrutiny, I have chosen the Kokudo(3) news story.

This news topic is fairly representative for the rest of the topics, for a couple of reasons. First, there are stories about it spread somewhat evenly throughout the recorded period. Second, it is not one of the stories that is tracking exceptionally well (or exceptionally badly).

Experiments as outlined in section 4 were run.

The resulting graphs can be seen in figure 7. The line marked "basic" shows standard tracking results — i.e. all shots used. However, there is one other shot selection method that overall works better for this topic - using only the longest shot.

Also worth noting is that in all these tracking examples a model size of 4 and a timeout of 1 day is used (as per (1)).

Further, it is clear that for this topic strict filtering (0.7, 0.8) generates lower f-measures. Looking at Precision and Recall specifically (figure 8 and figure 9), we find that removing shots which are within a distance of 0.5 or 0.6 of each other gives really good Precision, but significantly better than the basic run.

It's the 0.6 and 0.7 filters. 0.8 cam also be seen, but it is almost identical to basic tracking. Only removing shots if they are not within 0.8 of another shot means that we get to keep almost all of them.

Interesting to note in figure 11 and figure 12, however, is that both the 0.6 and 0.7 run have pretty good Precision and Recall, but not exceptional. The really good result comes from the combination of those two.

5 Conclusions

From the tests performed we can conclude several things.

First, similar shot exclusion, or SSE, does not improve the results of tracking. It performs occasionally better and occasionally worse than using all shots, and no clear trend can be seen.

Second, the longest shot approach is not viable. The F-measure got significantly worse for most stories, for some to the point where

Topic	Best F, SSI	Best F, SSE	Best F, Longest	Best F, Basic
Anti-Japan Demonstration	0.37 @ 0.3	0.36 @ 0.7	useless	0.35
Derailment Accident	0.66 @ 0.4	0.66 @ 0.2	useless	0.66
Earthquake Fukuoka	0.25 @ 0.8	0.26 @ 0.3	0.11	0.28
Earthquake Sumatra	0.17 @ 0.5	0.14 @ 0.2	0.08	0.14
Kokudo	0.26 @ 0.6	0.17 @ 0.5	0.23	0.18
Livedoor	0.27 @ 0.2	0.27 @ 0.3	0.26	0.25
Pirate	0.27 @ 0.4	0.25 @ 0.6	useless	0.30
Pontiff	0.19 @ 0.4	0.16 @ 0.7	useless	0.13

Table 1: SSI = Similar Shot Inclusion, SSE = Similar Shot Exclusion, Longest = Longest shot, Basic = all shots used. The number after the F-measure is the filter setting.

the results are meaningless. These inconsistencies also throw the stories that showed fairly good results in doubt - this could be due to pure luck.

Third, similar shot inclusion, or SSI, seems to improve the results. More testing on more video news topics with more data will be needed to definitely confirm this, but the results of our experiments clearly indicate that this will be a viable approach, with 5 out of 8 stories showing better results than original tracking, one unchanged and two that got slightly worse results.

6 Suggested Improvements

There are several other ways in which news tracking could be improved. Before we decided that the best path to take was to improve shot selection, we had several other options. Those I investigated, and I'd like to say a few words here about them.

6.1 New Performance Metric

As has been mentioned earlier (see Evaluation Measures, above) Precision and Recall might not be entirely appropriate measures for this type of video news tracking. It might be more appropriate to use Utility, a metric used in [4] for a document retrieval system. Utility is more punishing for incorrect matches than Precision / Recall, and video news tracking is in many ways more similar to document filtering than database searching (which is what Precision / Recall was developed for).

6.2 Improving Shot-Shot Matching

In several papers focusing on image retrieval (2, 5) the authors use EMD on a subset of histograms on the image. They find that this gives significantly better results in image-image comparisons, compared to straight histogram comparisons. It would be interesting to use this on our video news tracking system.

6.3 Texture Detection

Currently, we only look at the histogram, that is, the color content of the image. As has been demonstrated (5), it is also possible to use the textures within an image for matching. This information, too, would be most interesting to add to our tracking system.

6.4 Story Focus

We are really searching for a matching story, not a matching shot. This means that looking for a similar story based on the shots could be a blind alley. Disregarding the shots and instead taking a frame out, say, every five seconds, would put the focus on the story instead

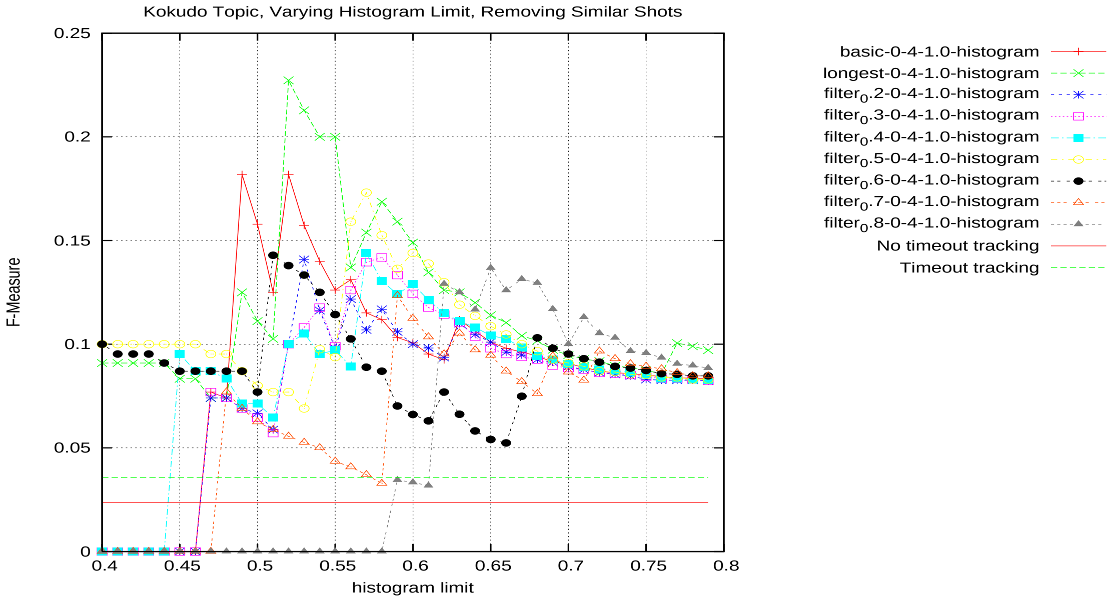

Figure 7: Tracking Kokudo, we've removed shots that are similar. We remove more and more as the filter value goes up. Note the curve for "basic" tracking as well as "longest".

of the shots. A longer shot would automatically get more weight by having more frames in the comparison.

7 Acknowledgements

I would like to thank the Sweden-Japan foundation for partially funding this work, and most importantly, to all the friendly people at KDDI Research Labs in Japan, for having patience with me and helping me out, especially my supervisor Keiichiro Hoashi.

8 References

1. Mats Uddenfeldt, Keiichiro Hoashi, Kazunori Matsumoto, and Fumiaki Sugaya, "Adaptive Video News Story Tracking Based On Earth Movers Distance", ICME 2006

2. Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "The Earth Movers Distance As A Metric For Image Retrieval", Int. J. Comput. Vision, vol. 40, no.2, pp. 99-121, 2000.

3. Kokudo story: http://www.iht.com/articles/2005/03/03/news/japan.php

4. Keiichiro Hoashi, Kazunori Matsumoto, Naomi Inoue, and Kazuo Hashimoto, "Document Filtering Method Using Non-Relevant Information Profile", SIGIR 2000

5. Yossi Rubner and Carlo Tomasi, "Texture-Based Image Retrieval Without Segmentation"

6. Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "A Metric for Distribituions with Applications to Image Databases", IEEE ICCV 1998

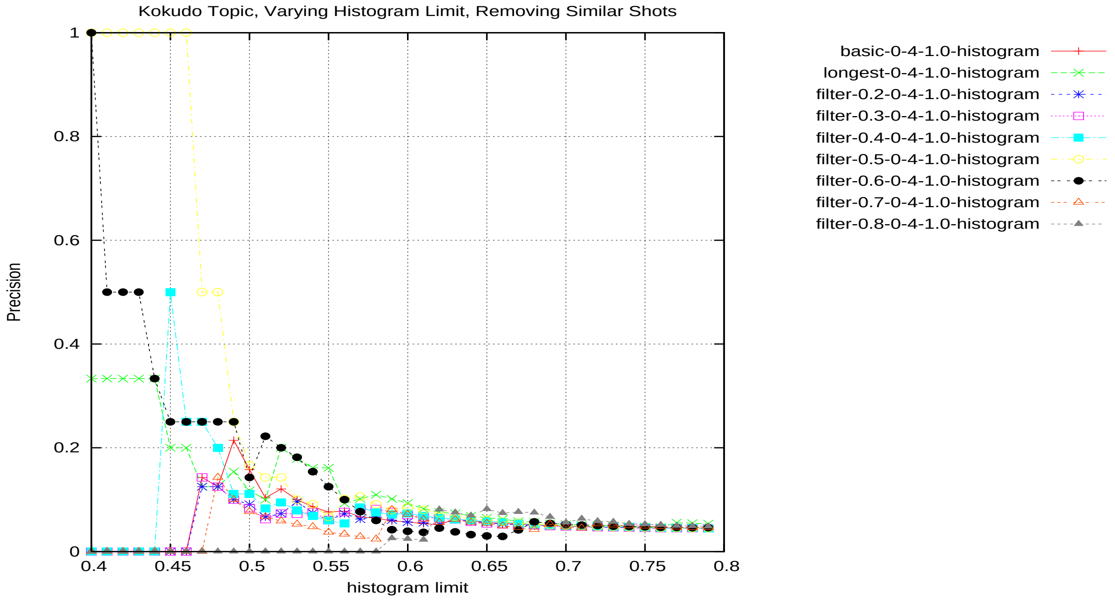

Figure 8: Removing similar shots (SSE). Highest Precision achieved while removing shots that are within 0.5 of each other.

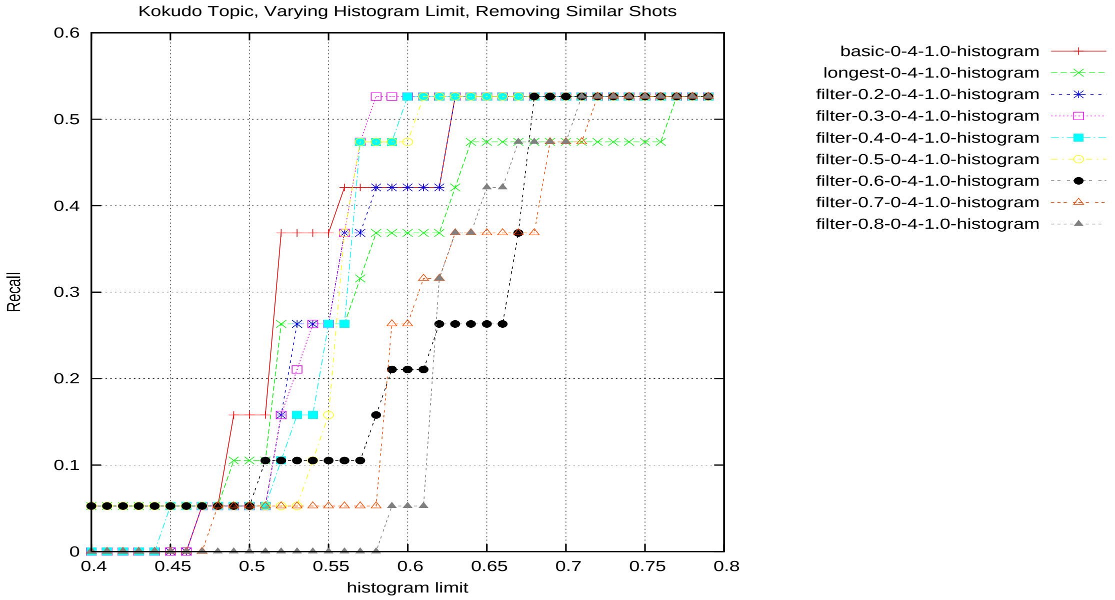

Figure 9: Removing similar shots (SSE). Here, higher scores as far to the left as possible is desirable. Basic tracking gives a Recall curve that's preferable to the others.

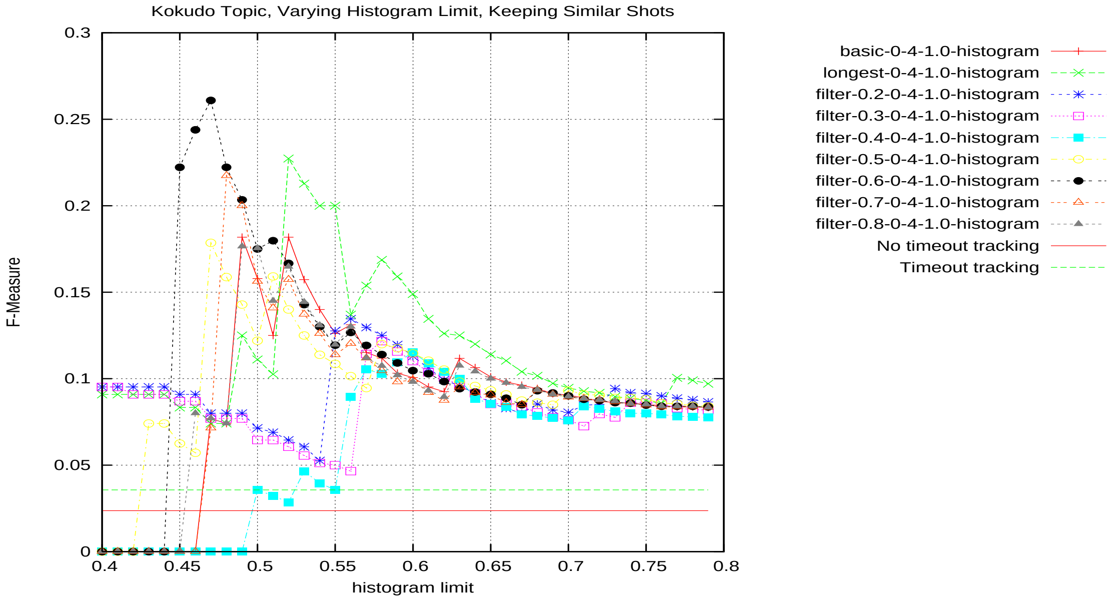

Figure 10: F-measure when keeping shots that have other, similar ones, within the same story. Note the 0.6 and 0.7 plots.

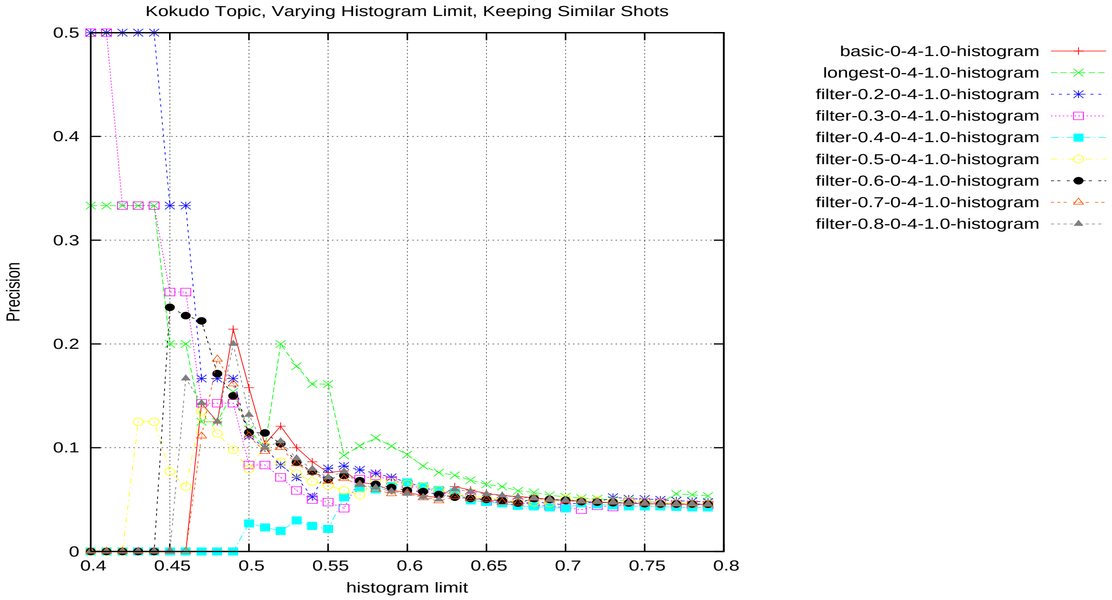

Figure 11: Again, the strictest filtered tracks get the best Precision. However, their F-measure is not very good.

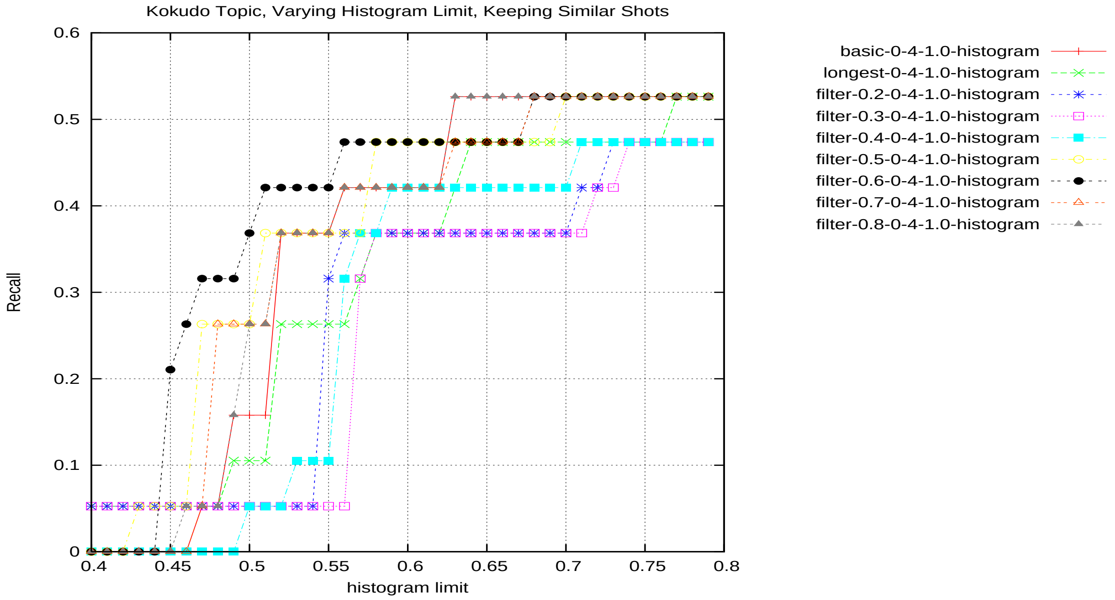

Figure 12: As for Recall, several tracks match or beat regular tracking. Most notably, the run that has been filtered with a filter setting of 0.6 is outperforming most of the other tracking runs.