Very quick audio searching

~ introducing global pruning to the time-series active search ~

Akisato Kimura, Kunio Kashino,
Takayuki Kurozumi and Hiroshi Murase

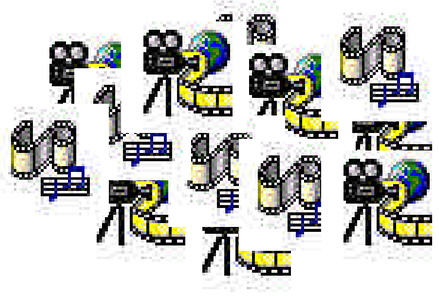
NTT Communication Science Labs.

1

Backgrounds

a 24-hour media information

a month or a year
media information



a quick search method
for media information

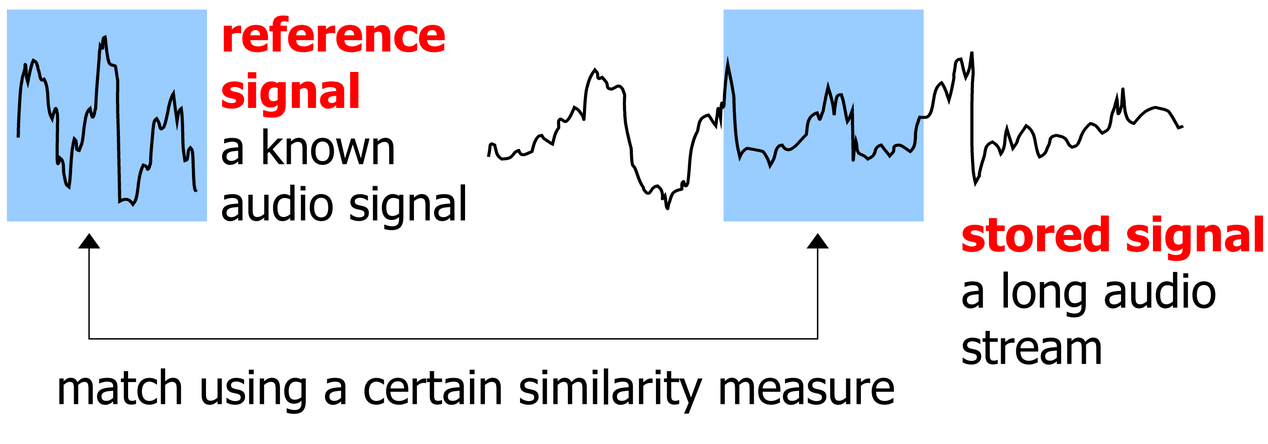
Time-series
active search
[Kashino at el,1999]

an even faster method
for media information

a target of
this work

2

Objectives

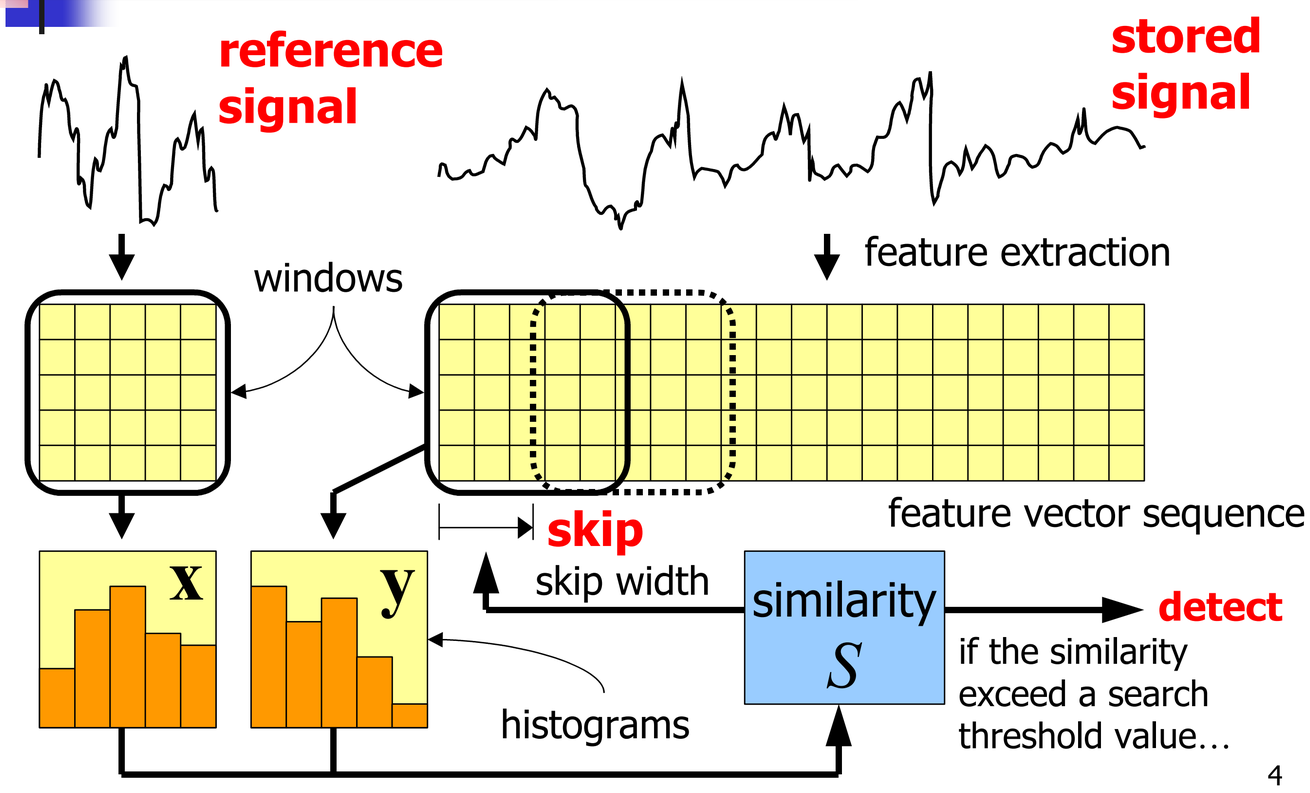


- detect and locate a reference signal
- the signal segments to be detected preserve the same spectrum pattern as the reference signal, except for minor distortions or noises

3

Time-series active search (TAS)

[Kashino et al. 1999]



4

Time-series active search (TAS)
[Kashino et al. 1999]

- takes less than 1 second to detect a 15 second reference signal in a 24-hour stored signal
- guarantees search accuracy with respect to the similarity equivalent to L_1-distance

$$S_1(\mathbf{x},\mathbf{y}) = \frac{1}{D}\sum_{i=1}^{L}\min(x_i, y_i)$$ (histogram intersection)

↕ equivalent

$$d_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{L}\left|x_i - y_i\right|$$ (L_1-distance)

$$\mathbf{x} = \{x_i\}_{i=1}^{L},\ \mathbf{y} = \{y_i\}_{i=1}^{L}$$ (a reference histogram and stored histograms)

5

Approach

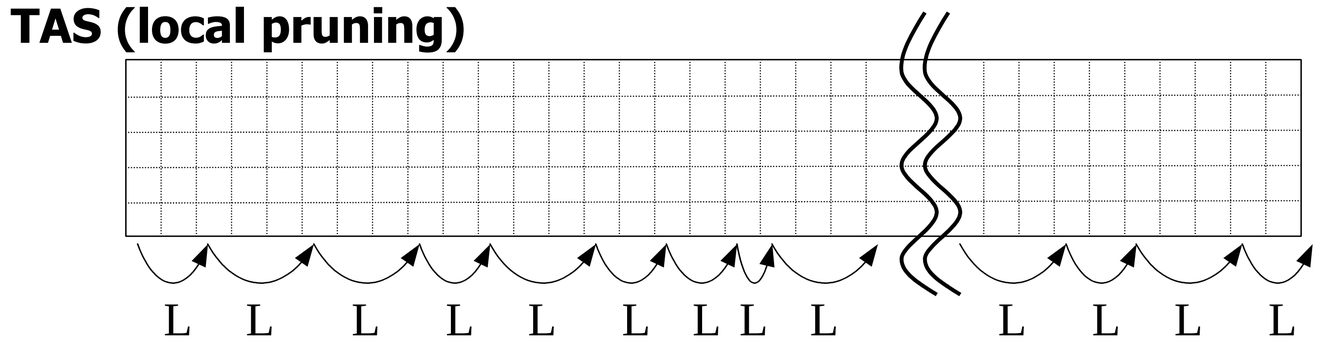
<TAS method>
local pruning (=skipping)
<proposing method>
global pruning +local pruning

- guaranteeing search accuracy with respect to a certain similarity standard

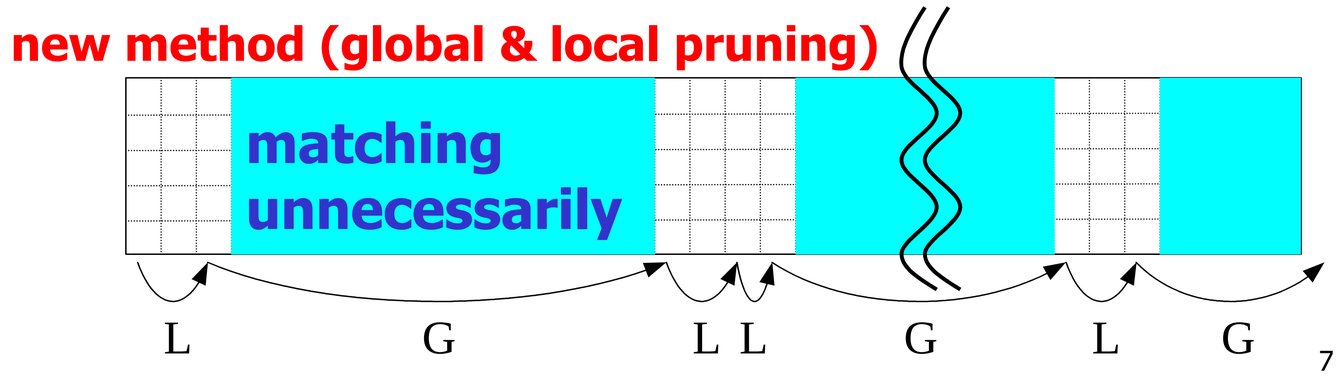
6

Global pruning

TAS (local pruning)



new method (global & local pruning)



7

Procedure of TAS

<preprocessing stage>

1. extract features from the stored signal

<search stage>

1. extract features from the reference signal and create histograms

2. match histograms

8

Procedure of proposed method

<preprocessing stage>

1. extract features from the stored signal **and create histograms**
2. **divide the histogram space**
3. **classify the histograms**

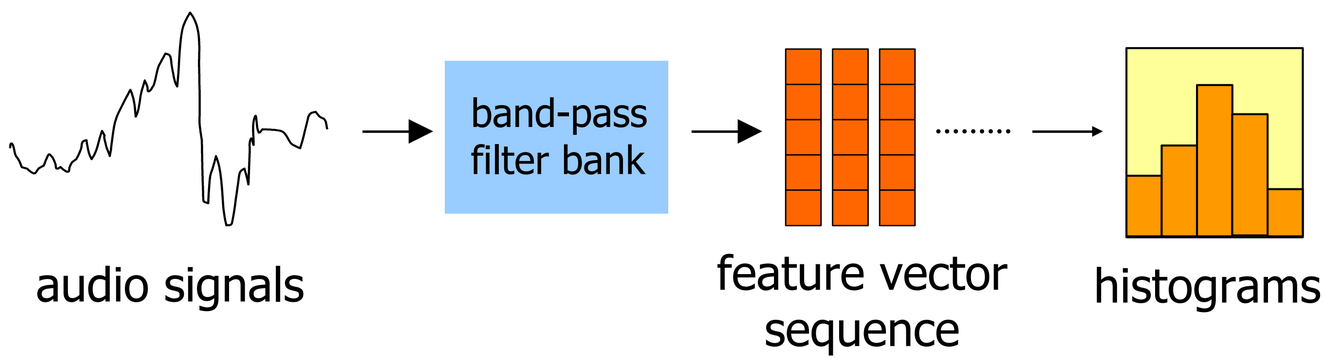
<search stage>

1. extract features from the reference signal and create a histogram
2. **perform global pruning**
3. match histograms

9

Extract features from the stored signal and create histograms <pre 1>

- extract features by using a band-pass filter bank
- create histograms by classifying the feature vectors over the window



10

Divide the histogram space <pre 2>

- each division is used as a unit to determine whether matching is needed
- use a codebook learning algorithm for vector quantization (VQ) (ex. LBG algorithm)
- distance measure : L_2-distance (Euclid distance)

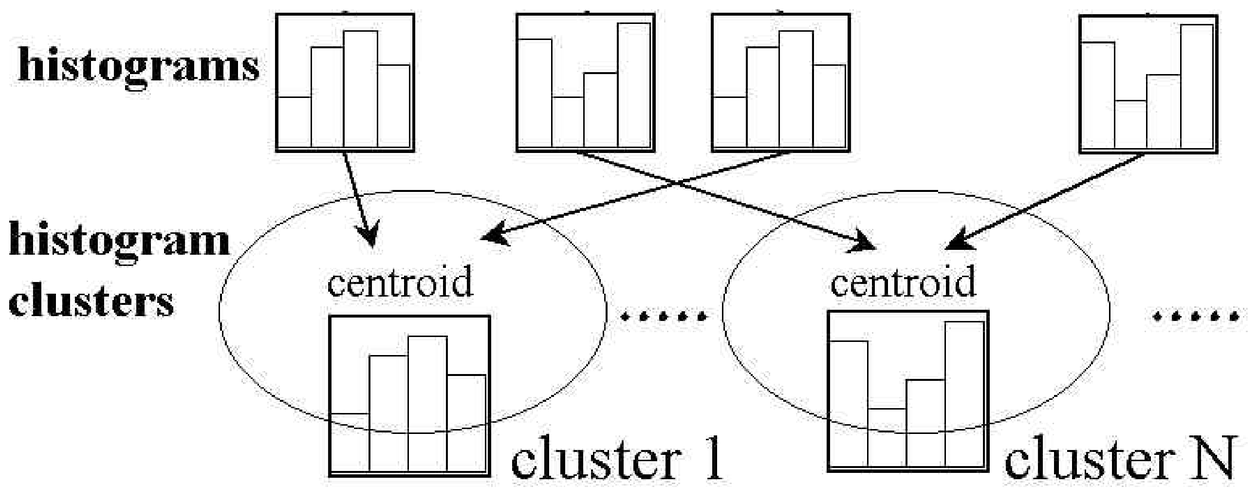
$$d_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{L}(x_i - y_i)^2}$$

- histogram clusters :
  the divisions of the histogram space

11

Classify histograms <pre 3>

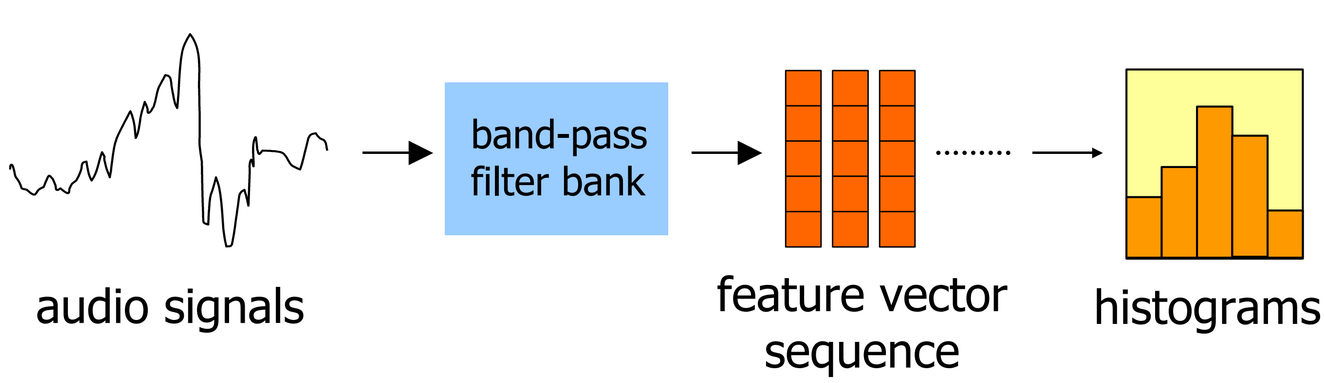
- classify histograms based on L_2-distance and using a certain VQ algorithm



12

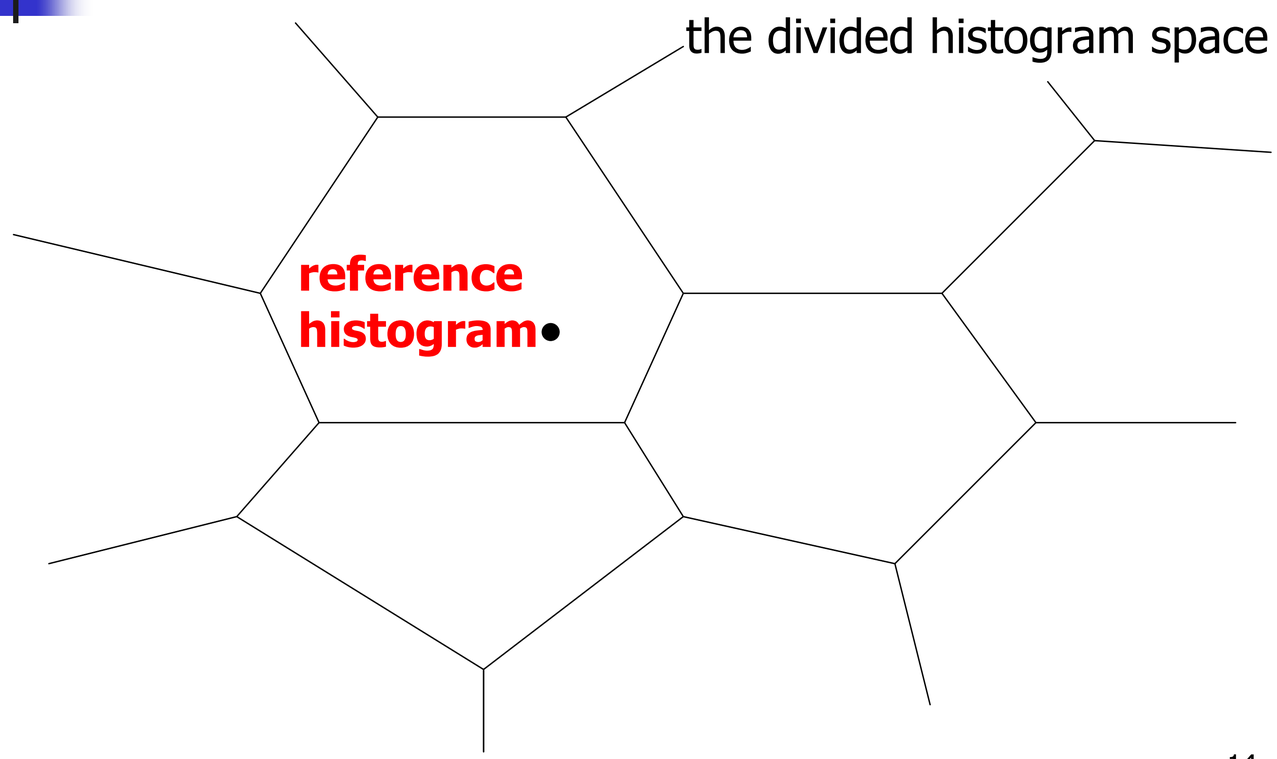
Extract features from the reference signal and create a histogram
<search 1>

- extract features using a band-pass filter bank
- create histograms by classifying the feature vectors



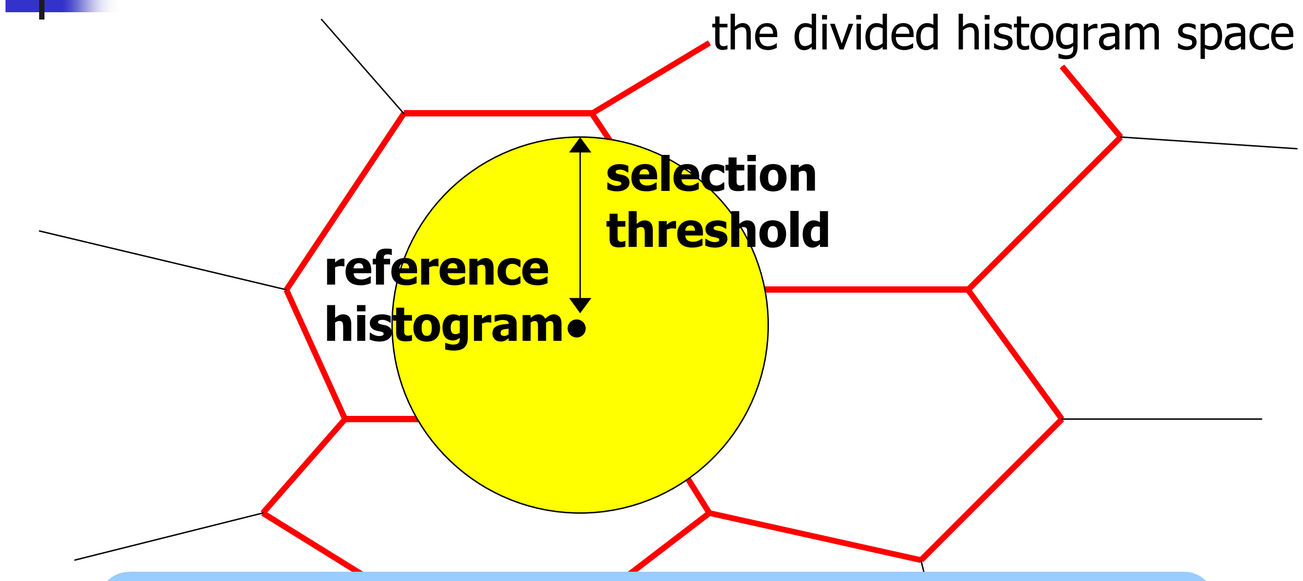
13

Perform global pruning
<search 2>



14

Perform global pruning
<search 2>



select clusters that may contain a histogram whose distance from the reference histogram is smaller than the selection threshold

15

Experiments

- **stored signal** :
  a recording of
  real 24-hour TV broadcasting
- **reference signal** :
  12 sections of 15 seconds
  randomly chosen from the stored signal
- **number of histogram bins** : 512
- **number of histogram clusters** : 512

16

Review : procedure of proposed method

<preprocessing stage>

1. extract features from stored signal and create histograms
2. divide the histogram space
3. classify the histograms

<search stage>

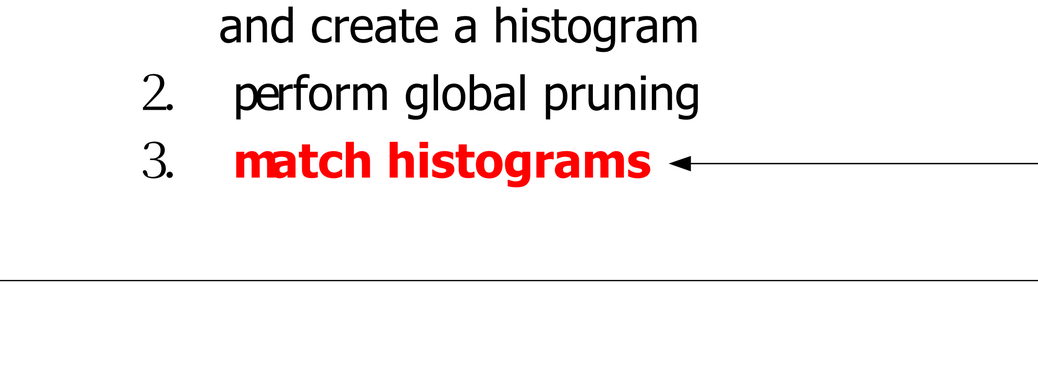
1. extract features from reference signal and create a histogram
2. perform global pruning
3. **match histograms**

search time includes only the time for this step

17

Results : search time

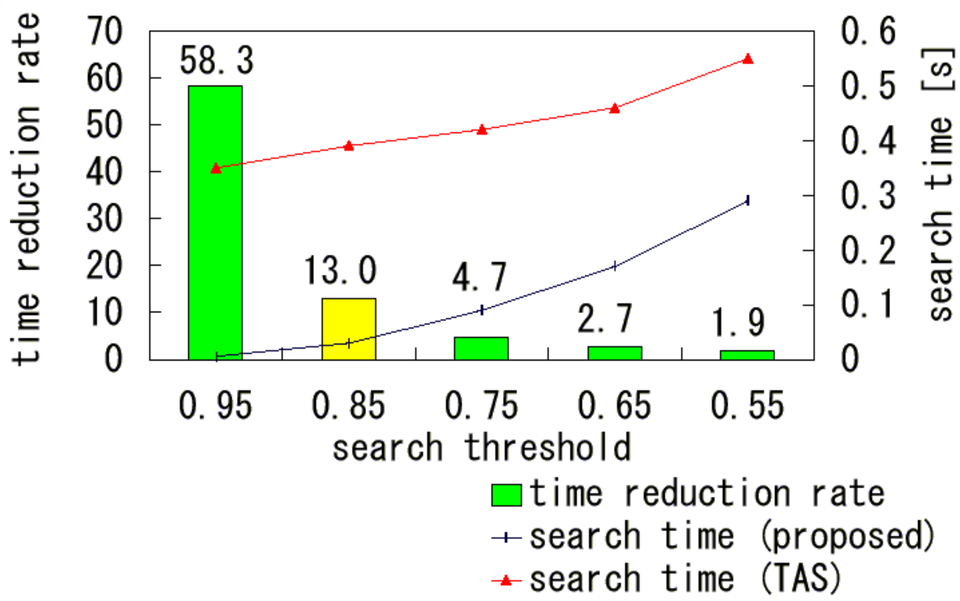
- search time includes only the time for the histogram matching step



(when the search threshold is 0.85)

18

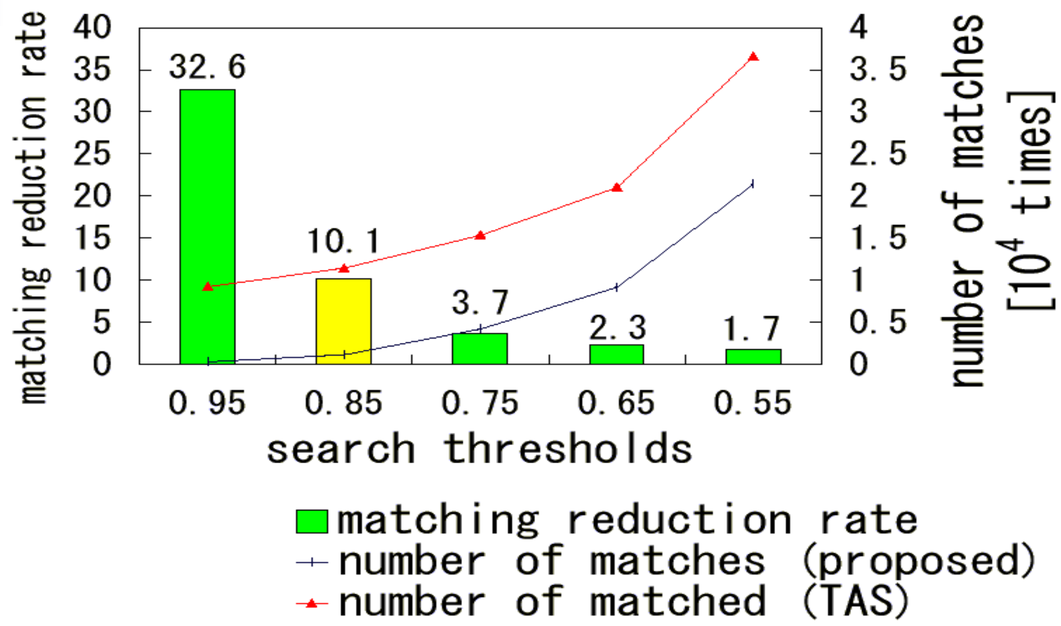
Results : search time



- detect the segments 13 times faster than TAS (search threshold = 0.85)

19

Results : number of matches



- reduces the number of matches to 1/10 (search threshold = 0.85)

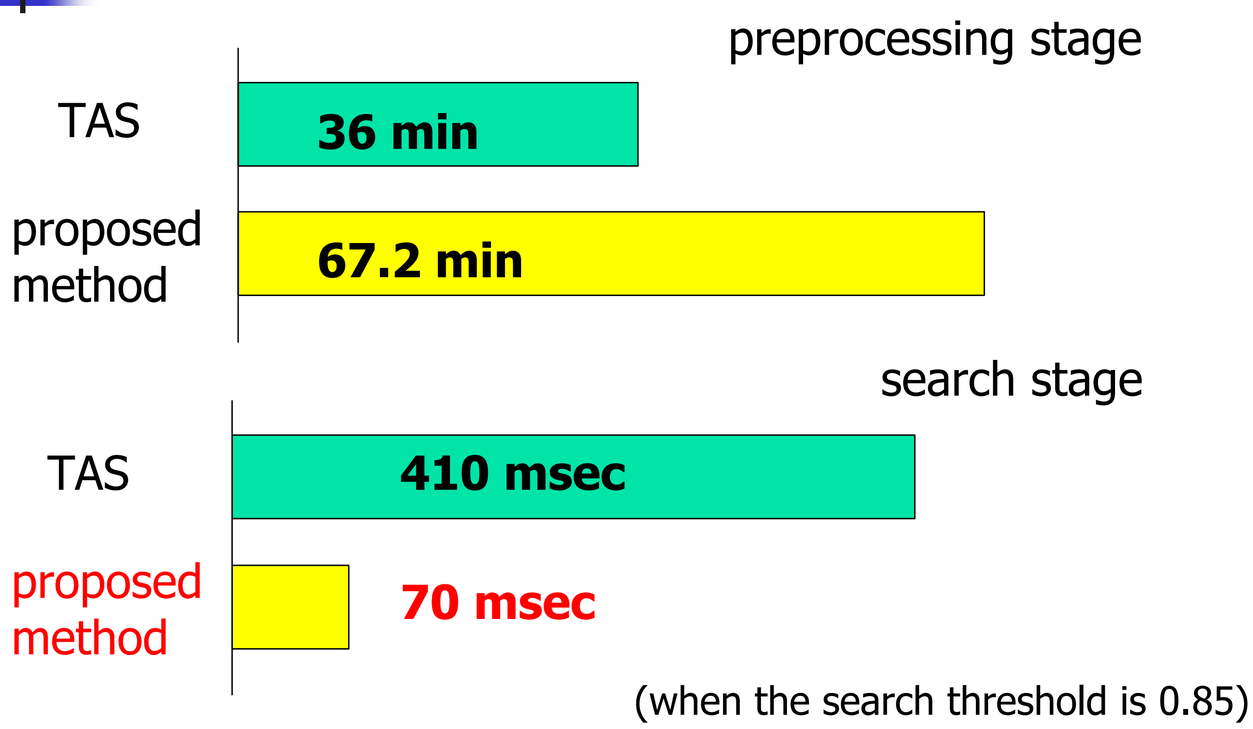
20

Conclusions

- We have proposed a global pruning method that enables a very quick search of audio signals
- It reduces the number of matches to 1/10 (search threshold = 0.85)
- Extension to video search is planned

21

Results : all the processing time (example)



(when the search threshold is 0.85)

22