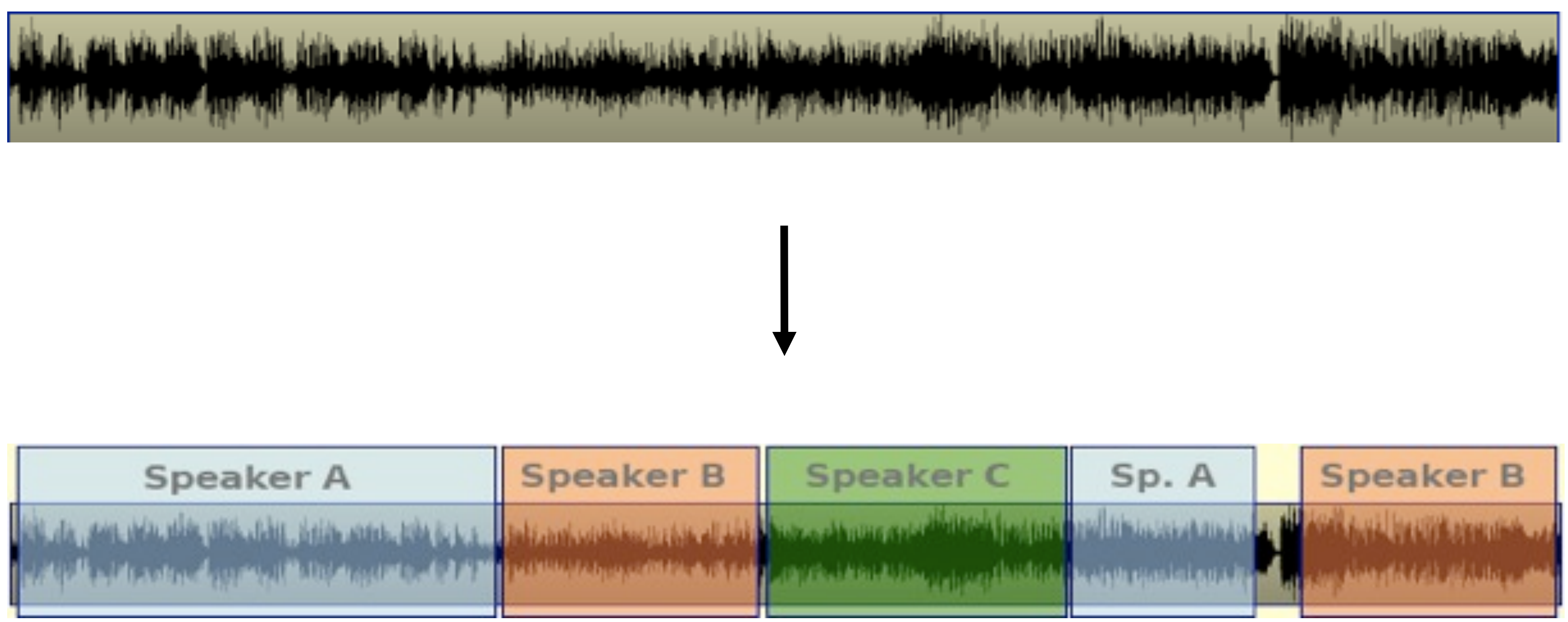
Introduction to Speaker Diarization



Dr. Gerald Friedland
International Computer Science Institute
Berkeley, CA
friedland@icsi.berkeley.edu

Speaker Diarization...

➡tries to answer the question: **“who spoke when?”**

➡using a single or multiple microphone inputs

➡without prior knowledge of anything (#speakers, language, text, etc...)

Visualization

Audiotrack:

Estimate “who spoke when” with no prior knowledge of speakers, #of speakers, words, or language spoken.

Visualization

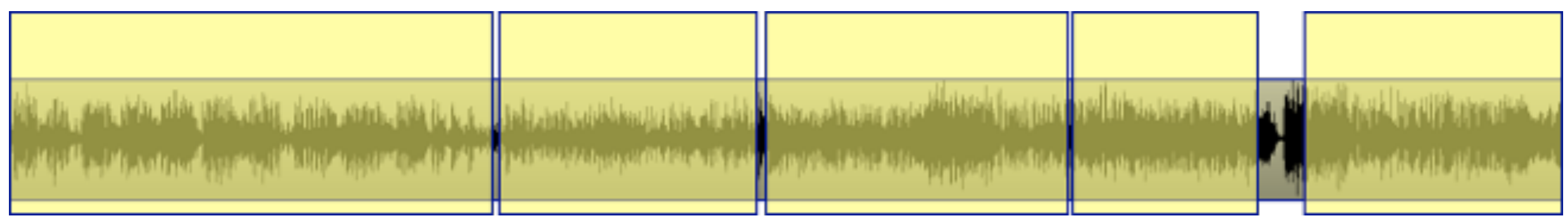
Audiotrack:

Estimate “who spoke when” with no prior knowledge of speakers, #of speakers, words, or language spoken.

Visualization

Audiotrack:

Segmentation:

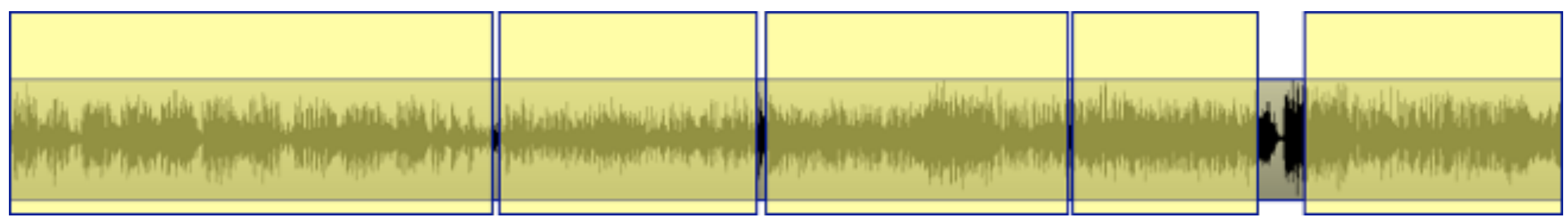


Estimate “who spoke when” with no prior knowledge of speakers, #of speakers, words, or language spoken.

Visualization

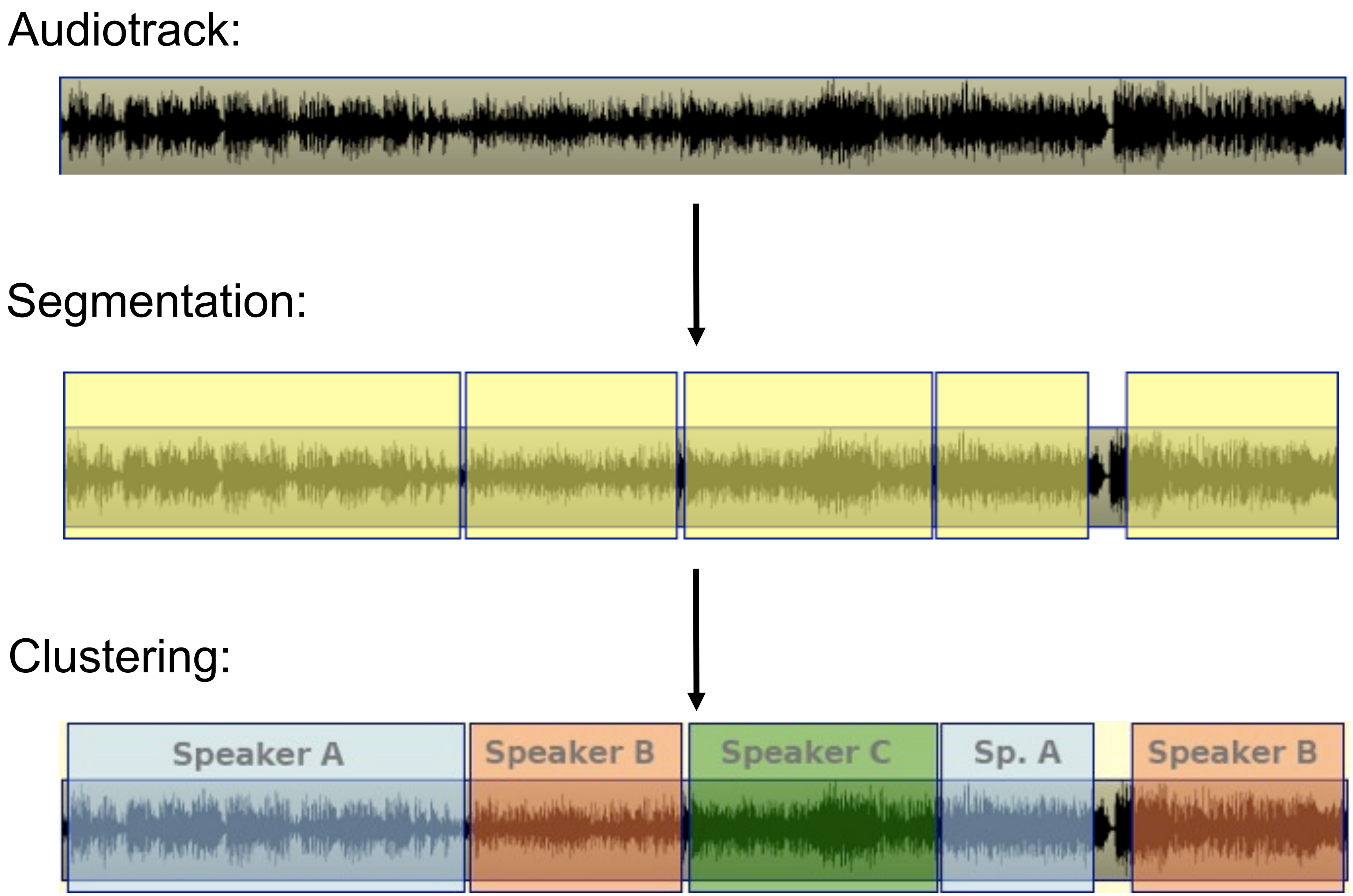
Audiotrack:

Segmentation:



Estimate “who spoke when” with no prior knowledge of speakers, #of speakers, words, or language spoken.

Visualization



Estimate “who spoke when” with no prior knowledge of speakers, #of speakers, words, or language spoken.

Speaker Diarization is NOT

Speaker Diarization is NOT

- Speaker ID (Speaker ID is supervized and needs prior training)

Speaker Diarization is NOT

- Speaker ID (Speaker ID is supervized and needs prior training)
- Speaker Verification (is supervized and returns yes/no answer)

Speaker Diarization is NOT

- Speaker ID (Speaker ID is supervized and needs prior training)
- Speaker Verification (is supervized and returns yes/no answer)
- Beamforming (as this requires multiple mics, even though beamforming can be used to support diarization)

Why Diarization?

Why Diarization?

- Important basic technology for various semantic audio analysis tasks

Why Diarization?

- Important basic technology for various semantic audio analysis tasks

- Meeting retrieval, video conferencing, speaker-adaptive ASR, video retrieval, etc...

Why Diarization?

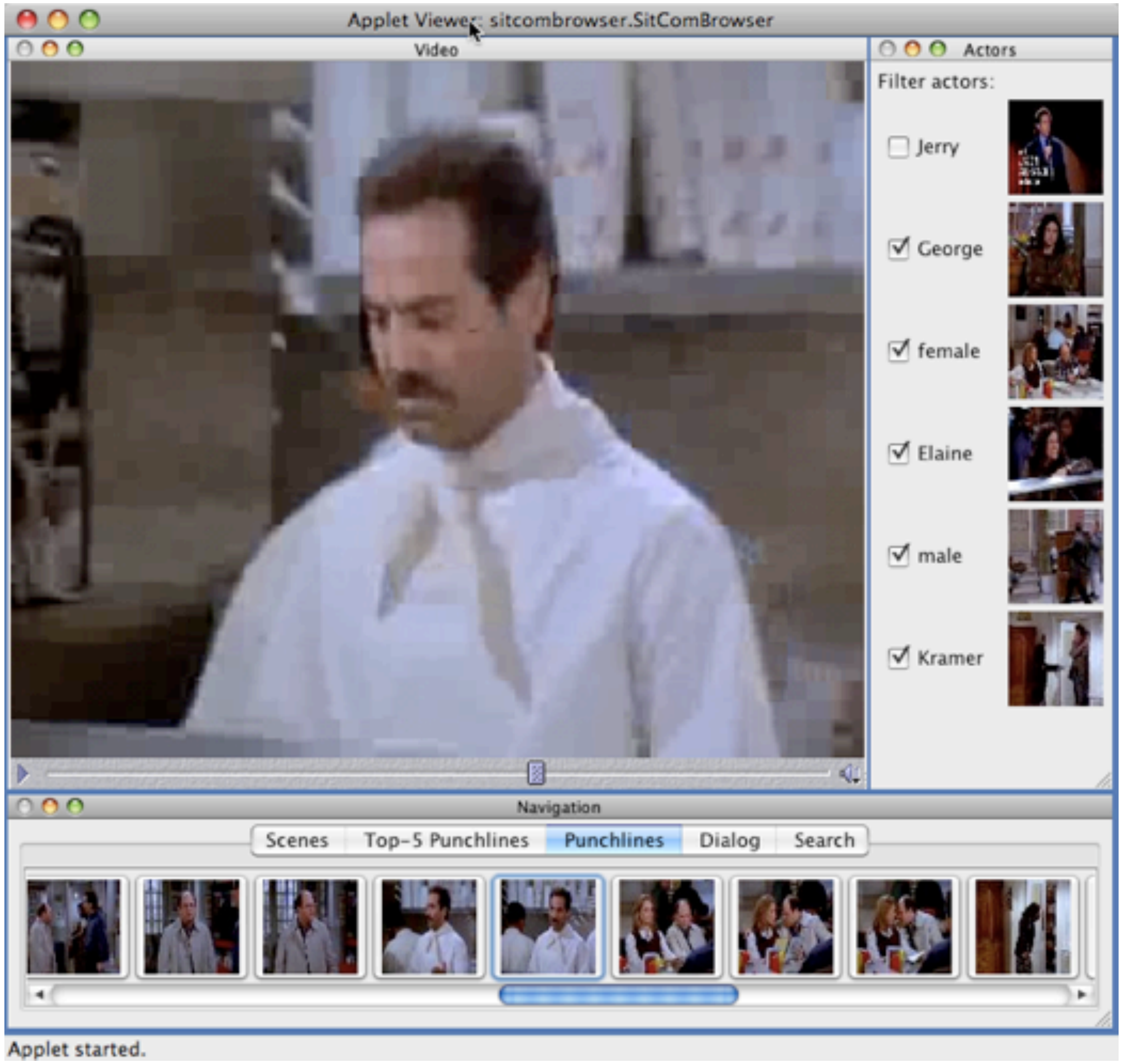
- Important basic technology for various semantic audio analysis tasks

- Meeting retrieval, video conferencing, speaker-adaptive ASR, video retrieval, etc...

- Let's take a look at some examples

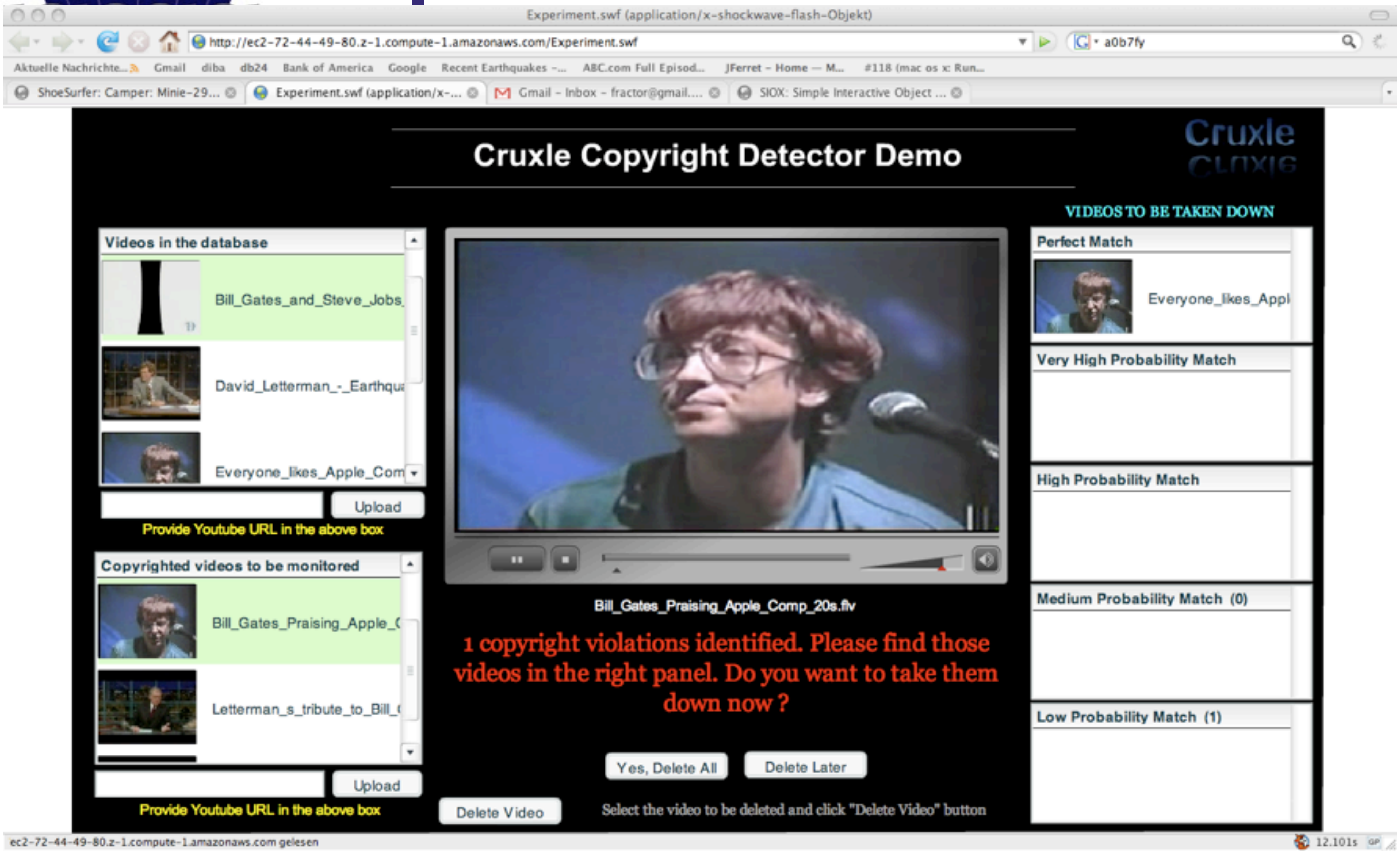
Application: Meeting Browsing

Application: Semantic Navigation



G. Friedland, L. Gottlieb, A. Janin: "Joke-o-mat: Browsing Sitcoms Punchline by Punchline", Proceedings of ACM Multimedia, Beijing, China, October 2009.

Application: Video Duplicate Detection



Other Applications

(Speaker) Diarization is often used as underlying support for...

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming
- Visual Localization

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming
- Visual Localization
- Video Analysis: Object Detection, Event Detection, Scene Detection

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming
- Visual Localization
- Video Analysis: Object Detection, Event Detection, Scene Detection
- behavior-level analysis tasks, such as dominance detection

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming
- Visual Localization
- Video Analysis: Object Detection, Event Detection, Scene Detection
- behavior-level analysis tasks, such as dominance detection
- Robotics Applications (e.g. addressing people)

Other Applications

(Speaker) Diarization is often used as underlying support for...

- Beamforming
- Visual Localization
- Video Analysis: Object Detection, Event Detection, Scene Detection
- behavior-level analysis tasks, such as dominance detection
- Robotics Applications (e.g. addressing people)
- Support for adaptive speech recognition

Main Drive: NIST RT Eval

Main Drive: NIST RT Eval

- Speaker Diarization was evaluated as part of the NIST Rich Transcription Evaluation (since about 2002)

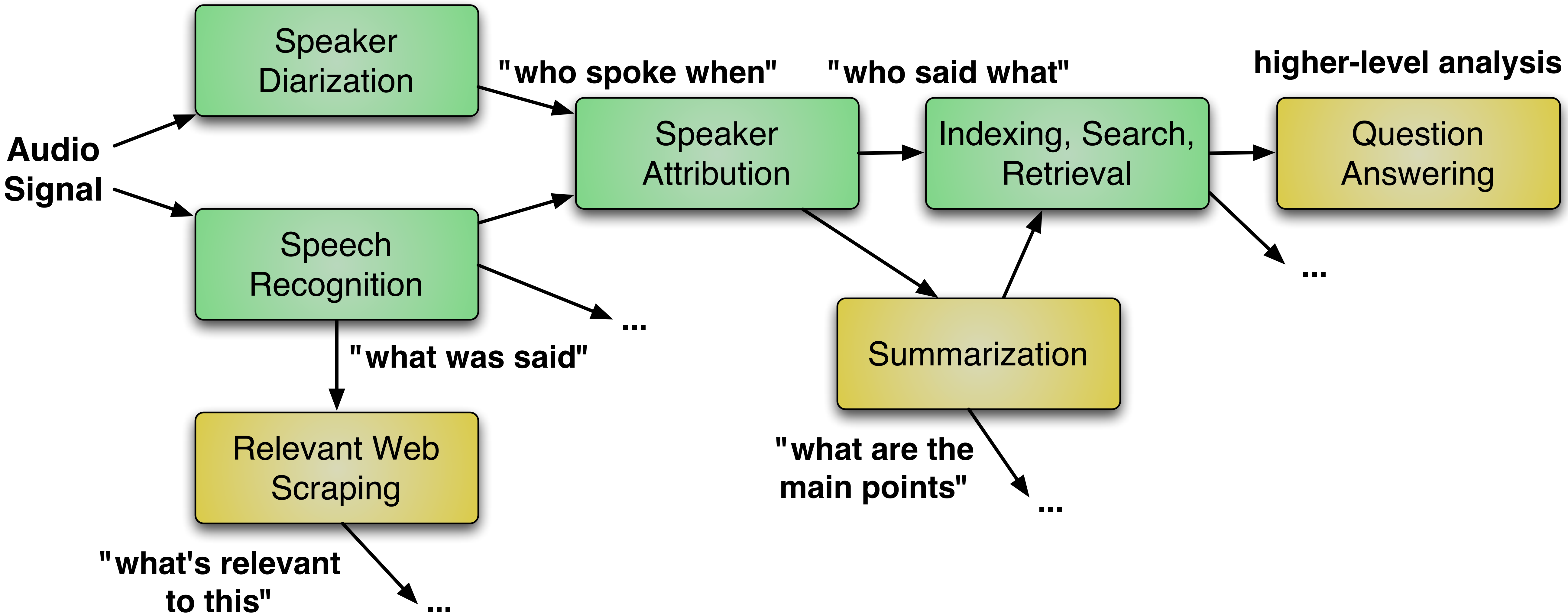
Main Drive: NIST RT Eval

- Speaker Diarization was evaluated as part of the NIST Rich Transcription Evaluation (since about 2002)
- Idea: Create "Rich Transcripts" of broadcast news, later meetings.

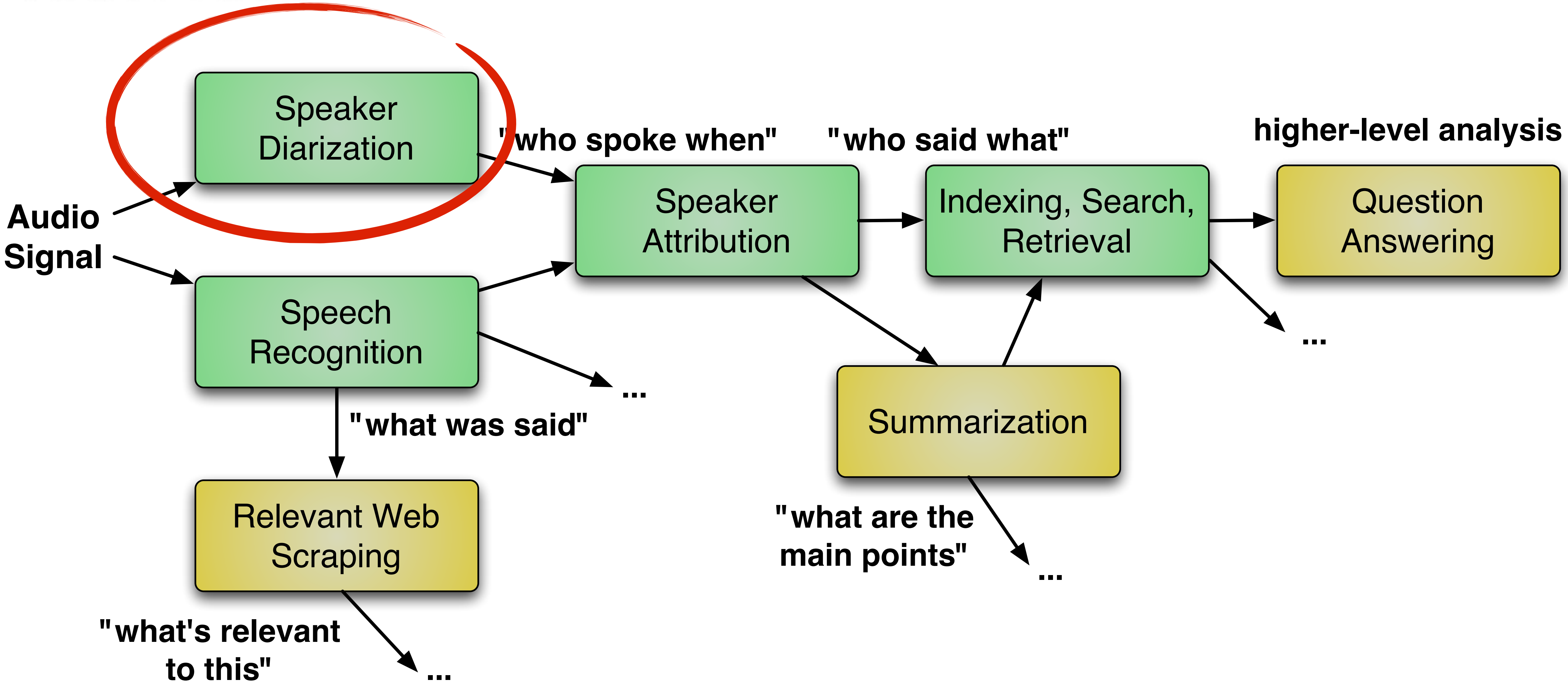
Main Drive: NIST RT Eval

- Speaker Diarization was evaluated as part of the NIST Rich Transcription Evaluation (since about 2002)
- Idea: Create "Rich Transcripts" of broadcast news, later meetings.
- Evaluated on Real-World data

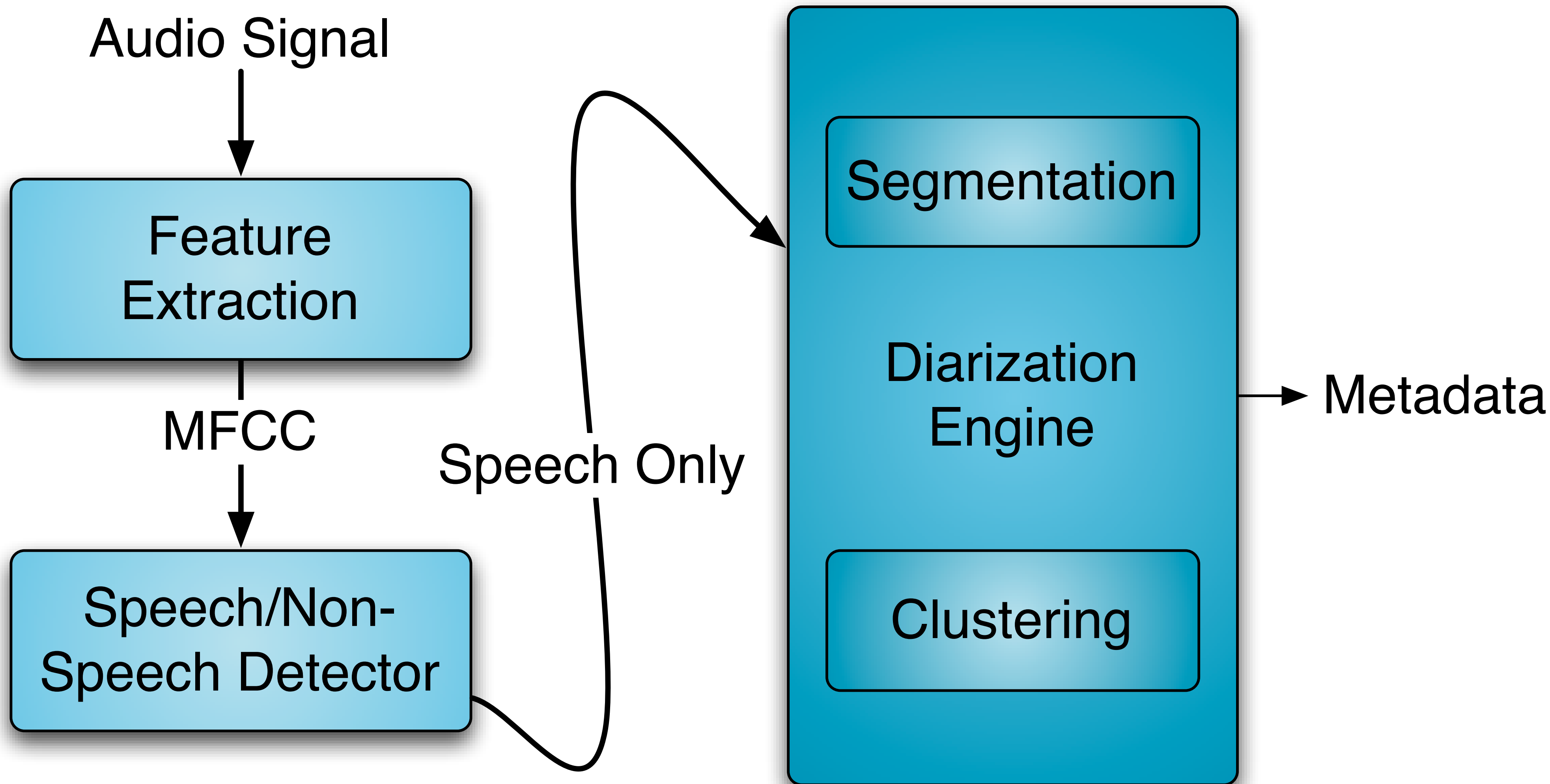
Typical Component Composition for RT



Typical Component Composition for RT



Speaker Diarization: General Overview



Output Format of Diarization

Output Format of Diarization

- RTTM files (as defined by NIST)

Output Format of Diarization

- RTTM files (as defined by NIST)
- Example:

Output Format of Diarization

- RTTM files (as defined by NIST)

- Example:

```
SPEAKER soupnazi 1 40.0 2.5 <NA> <NA> George <NA>
```

Output Format of Diarization

- RTTM files (as defined by NIST)

- Example:

```
SPEAKER soupnazi 1 40.0 2.5 <NA> <NA> George <NA>
SPEAKER soupnazi 1 42.5 2.5 <NA> <NA> Jerry <NA>
```

Output Format of Diarization

- RTTM files (as defined by NIST)

- Example:

```
SPEAKER soupnazi 1 40.0 2.5 <NA> <NA> George <NA>
SPEAKER soupnazi 1 42.5 2.5 <NA> <NA> Jerry <NA>
SPEAKER soupnazi 1 45.0 2.5 <NA> <NA> female <NA>
```

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Output Format of Diarization

- RTTM files (as defined by NIST)

- Example:

```
SPEAKER soupnazi 1 40.0 2.5 <NA> <NA> George <NA>
SPEAKER soupnazi 1 42.5 2.5 <NA> <NA> Jerry <NA>
SPEAKER soupnazi 1 45.0 2.5 <NA> <NA> female <NA>
```

Output Format of Diarization

- RTTM files (as defined by NIST)
- Example:

```
SPEAKER soupnazi 1 40.0 2.5 <NA> <NA> George <NA>
SPEAKER soupnazi 1 42.5 2.5 <NA> <NA> Jerry <NA>
SPEAKER soupnazi 1 45.0 2.5 <NA> <NA> female <NA>
```

- Large amount of tools available to deal with these files.

Error Measurement

Error Measurement

- US NIST defines error metrics and is evaluating speaker diarization on a regular basis

Error Measurement

- US NIST defines error metrics and is evaluating speaker diarization on a regular basis
- Error metrics is called 'Diarization Error Rate' (DER)

Error Measurement

- US NIST defines error metrics and is evaluating speaker diarization on a regular basis

- Error metrics is called ‘Diarization Error Rate’ (DER)

- All tools available open source

Error Measurement

$$\mathrm{DER} = \frac{T_{FA} + T_{MISS} + T_{SPK}}{T_{SPEECH}}$$

DER = The amounts of time a speaker has been assigned wrongly, missed, assumed when there is none, or assumed solely when there is more than one relative to the length of the audio.

Segmentation & Clustering

Segmentation & Clustering

- Originally: Segment first, cluster later

  Chen, S. S. and Gopalakrishnan, P., "Clustering via the bayesian information criterion with applications in speech recognition," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2001, Vol. 2, Seattle, USA, pp. 645-648.

Segmentation & Clustering

- Originally: Segment first, cluster later
  Chen, S. S. and Gopalakrishnan, P., “Clustering via the bayesian information criterion with applications in speech recognition,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2001, Vol. 2, Seattle, USA, pp. 645-648.

- More efficient: Top-Down and Bottom-Up Approaches

Segmentation: Secret Sauce

Segmentation: Secret Sauce

- How do you distinguish speakers?

Segmentation: Secret Sauce

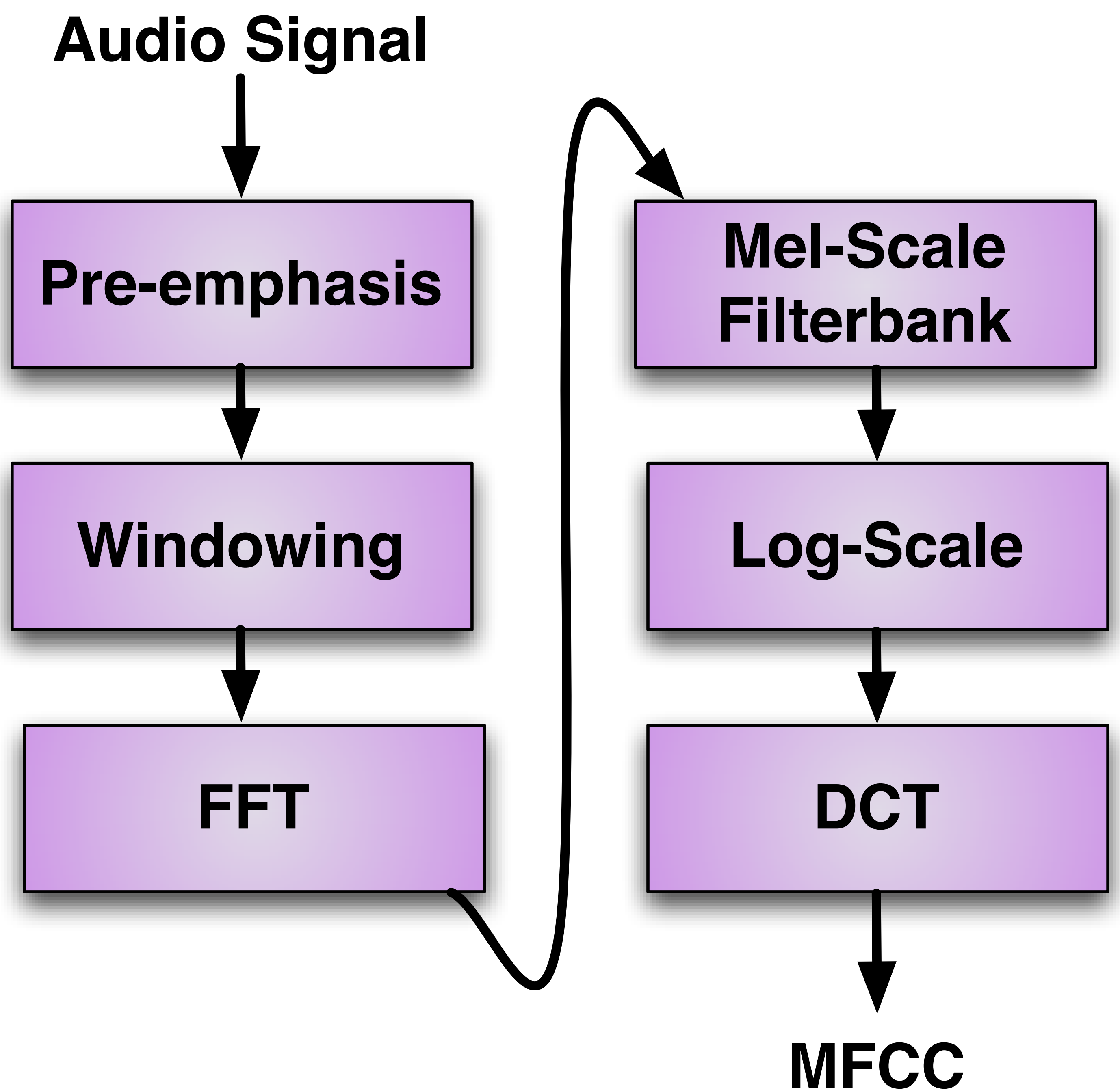
- How do you distinguish speakers?

- Combination of MFCC+GMM+BIC seems unbeatable!

Segmentation: Secret Sauce

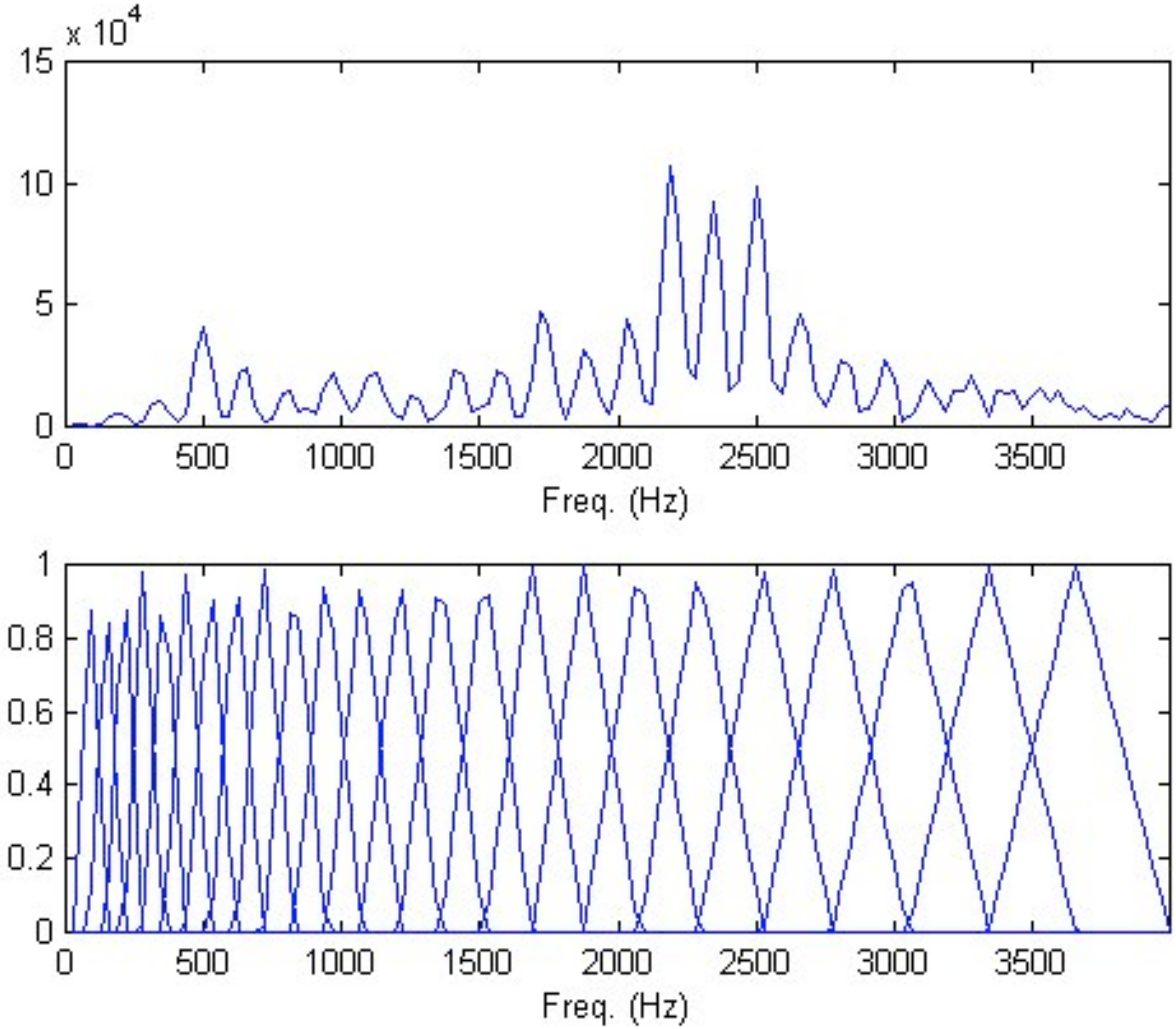
- How do you distinguish speakers?
- Combination of MFCC+GMM+BIC seems unbeatable!
- Can be generalized to Audio Percepts

MFCC: Idea

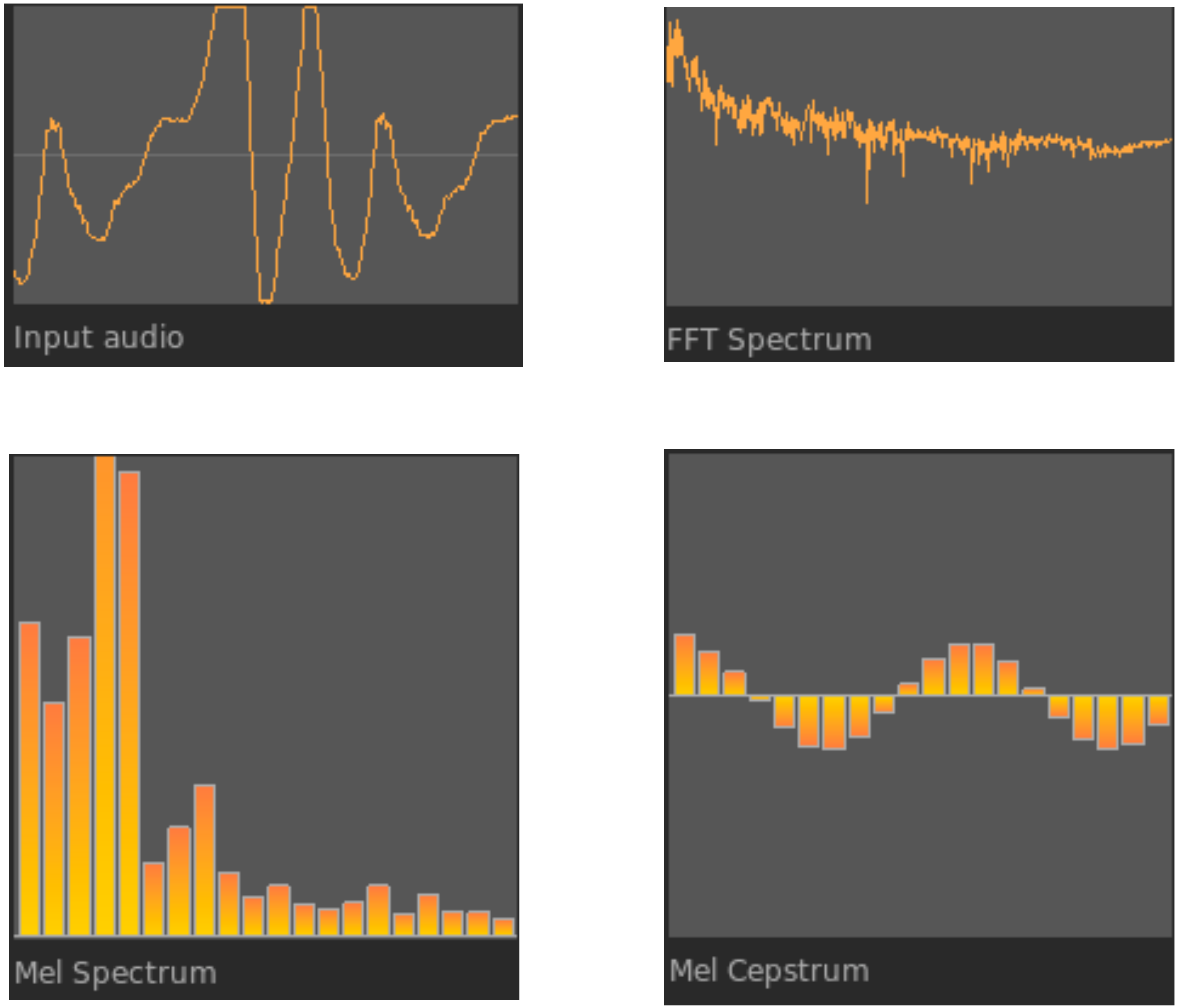


power cepstrum of signal $= \left|\mathbf{F}\left\{\log(|\mathbf{F}\left\{\text{the signal}\right\}|^2)\right\}\right|^2$

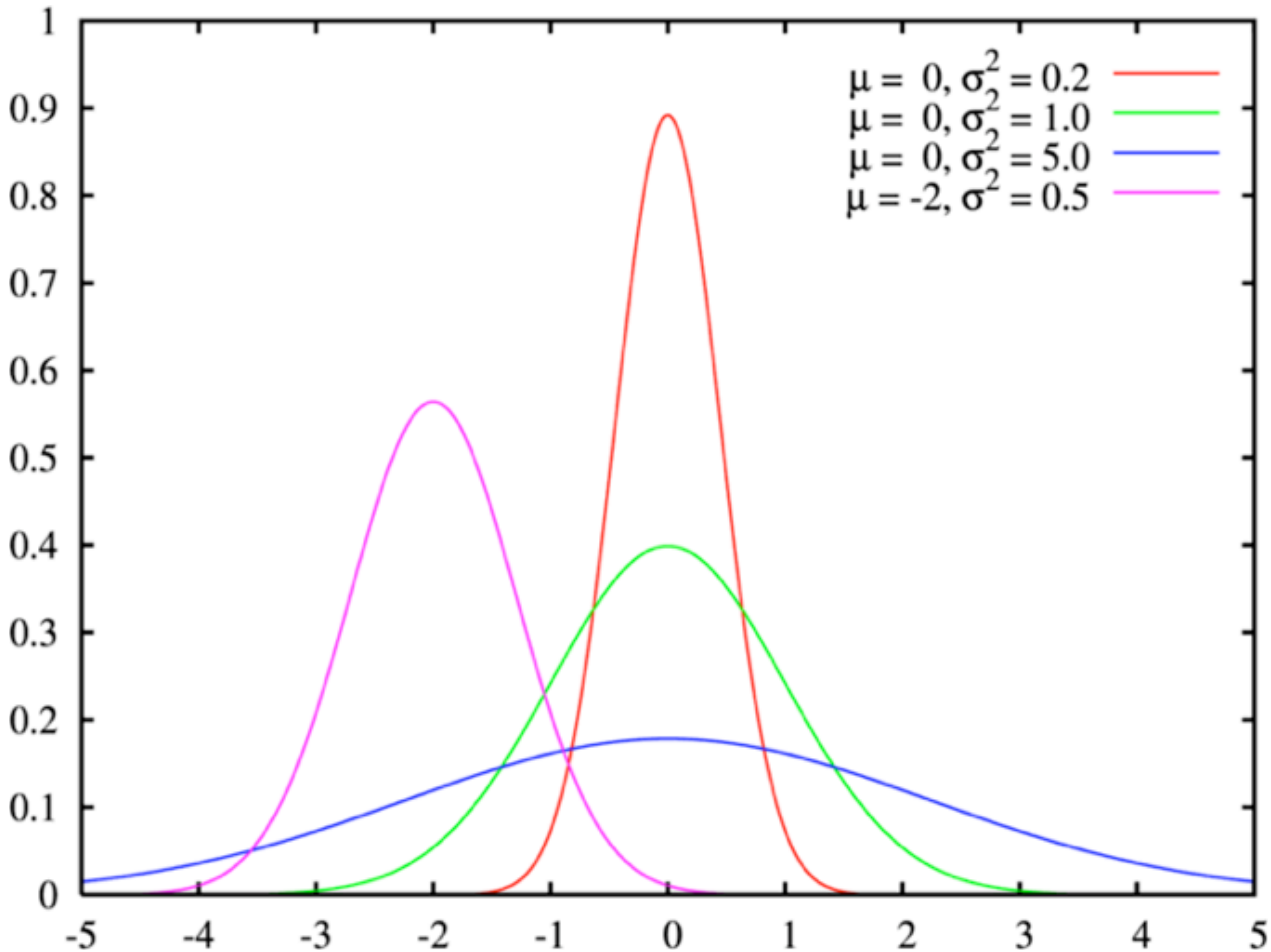
MFCC: Mel Scale



MFCC: Result



Gaussian Mixtures



Training of Mixture Models

Goal: Find a_i for $$f_X(x) = \sum_{i=1}^{n} a_i f_Y(x;\theta_i).$$

Expectation: $$y_{i,j} = \frac{a_i f_Y(x_j;\theta_i)}{f_X(x_j)}.$$

Maximization: $$a_i = \frac{1}{N}\sum_{j=1}^{N} y_{i,j}$$

$$\mu_i = \frac{\sum_j y_{i,j} x_j}{\sum_j y_{i,j}}.$$

Bayesian Information Criterion

$$\mathrm{BIC} = \log p(X|\Theta) - \frac{1}{2}\lambda K \log N$$

where
X is the sequence of features for a segment,
Θ are the parameters of the statistical model for the segment,
K is the number of parameters for the model,
N is the number of frames in the segment,
λ is an optimization parameter.

Bayesian Information Criterion: Explanation

Bayesian Information Criterion: Explanation

- BIC penalizes the complexity of the model (as of number of parameters in model).

Bayesian Information Criterion: Explanation

- BIC penalizes the complexity of the model (as of number of parameters in model).
- BIC measures the efficiency of the parameterized model in terms of predicting the data.

Bayesian Information Criterion: Explanation

- BIC penalizes the complexity of the model (as of number of parameters in model).
- BIC measures the efficiency of the parameterized model in terms of predicting the data.
- BIC is therfore used to choose the number of clusters according to the intrinsic complexity present in a particular dataset.

Bayesian Information Criterion: Properties

Bayesian Information Criterion: Properties

- BIC is a <u>minimum description length</u> criterion.

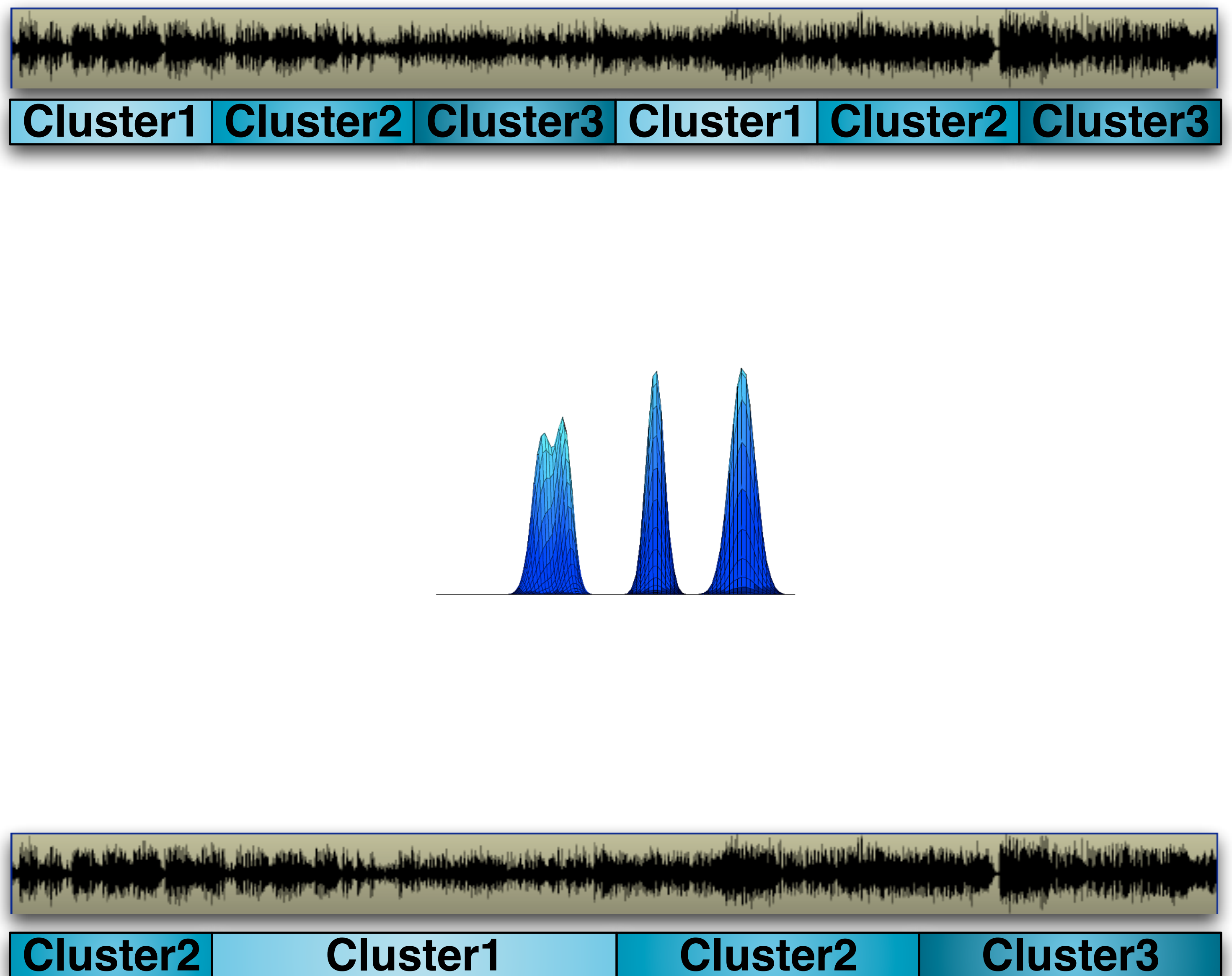
Bayesian Information Criterion: Properties

- BIC is a minimum description length criterion.
- BIC is independent of the prior.

Bayesian Information Criterion: Properties

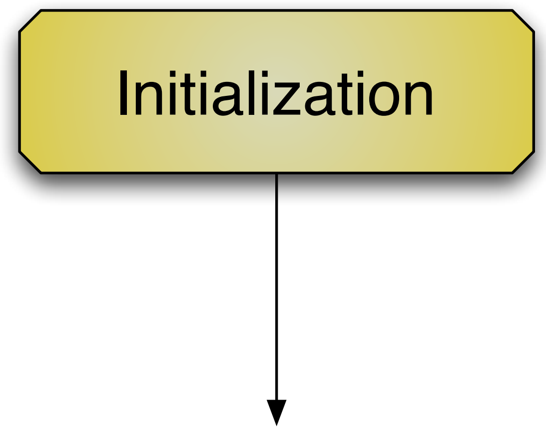
- BIC is a minimum description length criterion.
- BIC is independent of the prior.
- It is closely related to other penalized likelihood criteria such as RIC and the Akaike information criterion.

Bottom-Up Algorithm



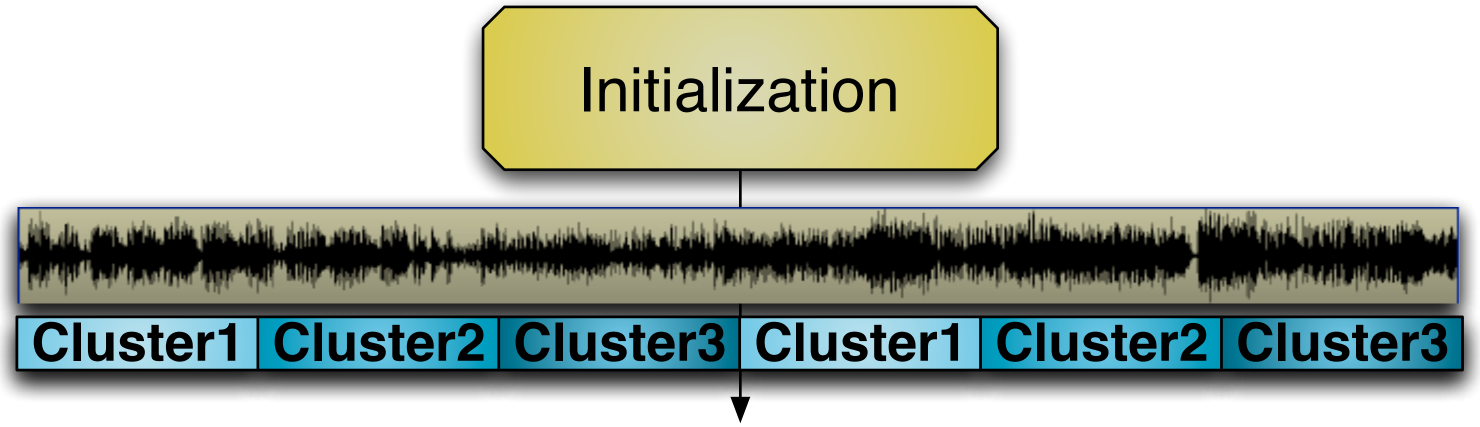
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



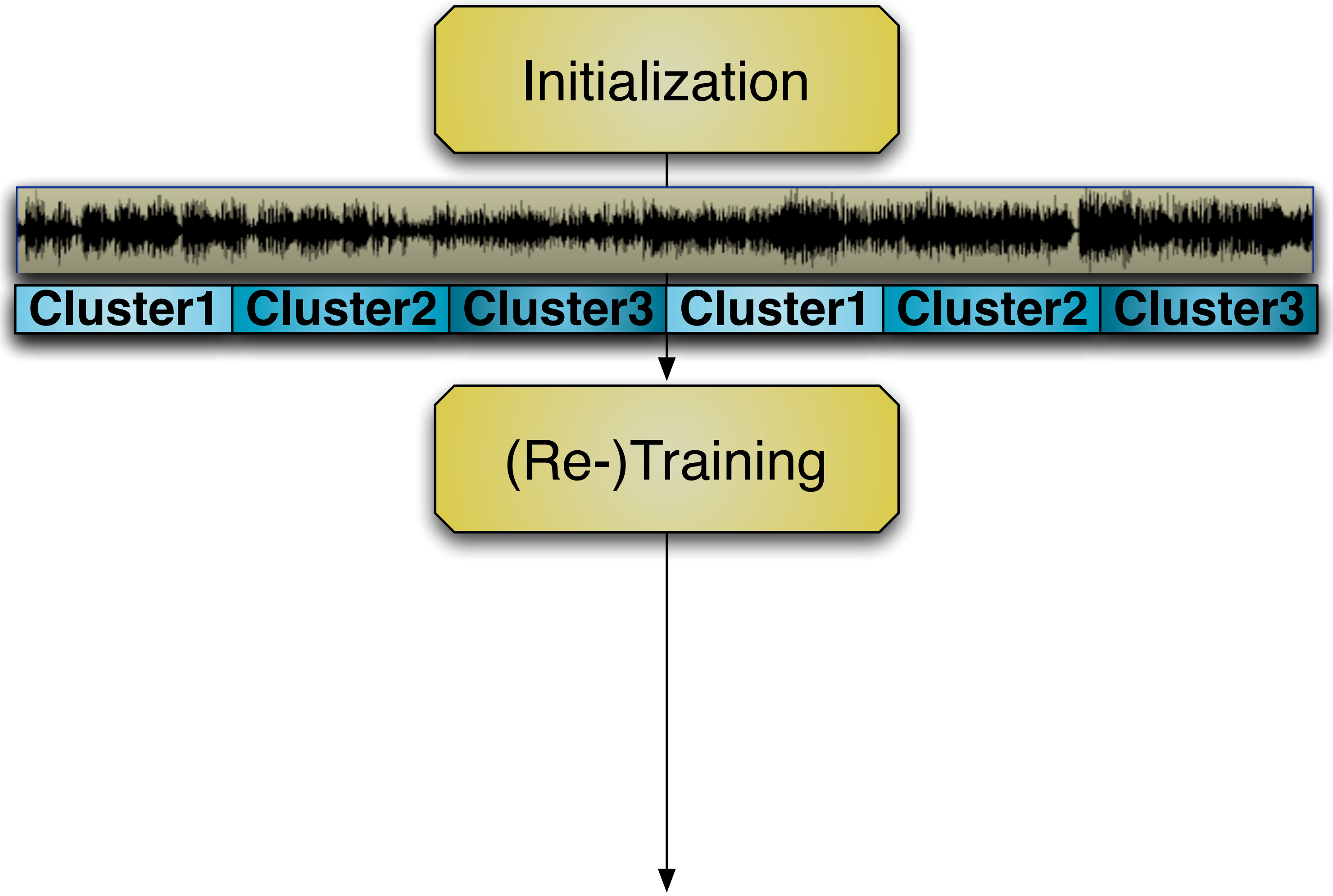
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



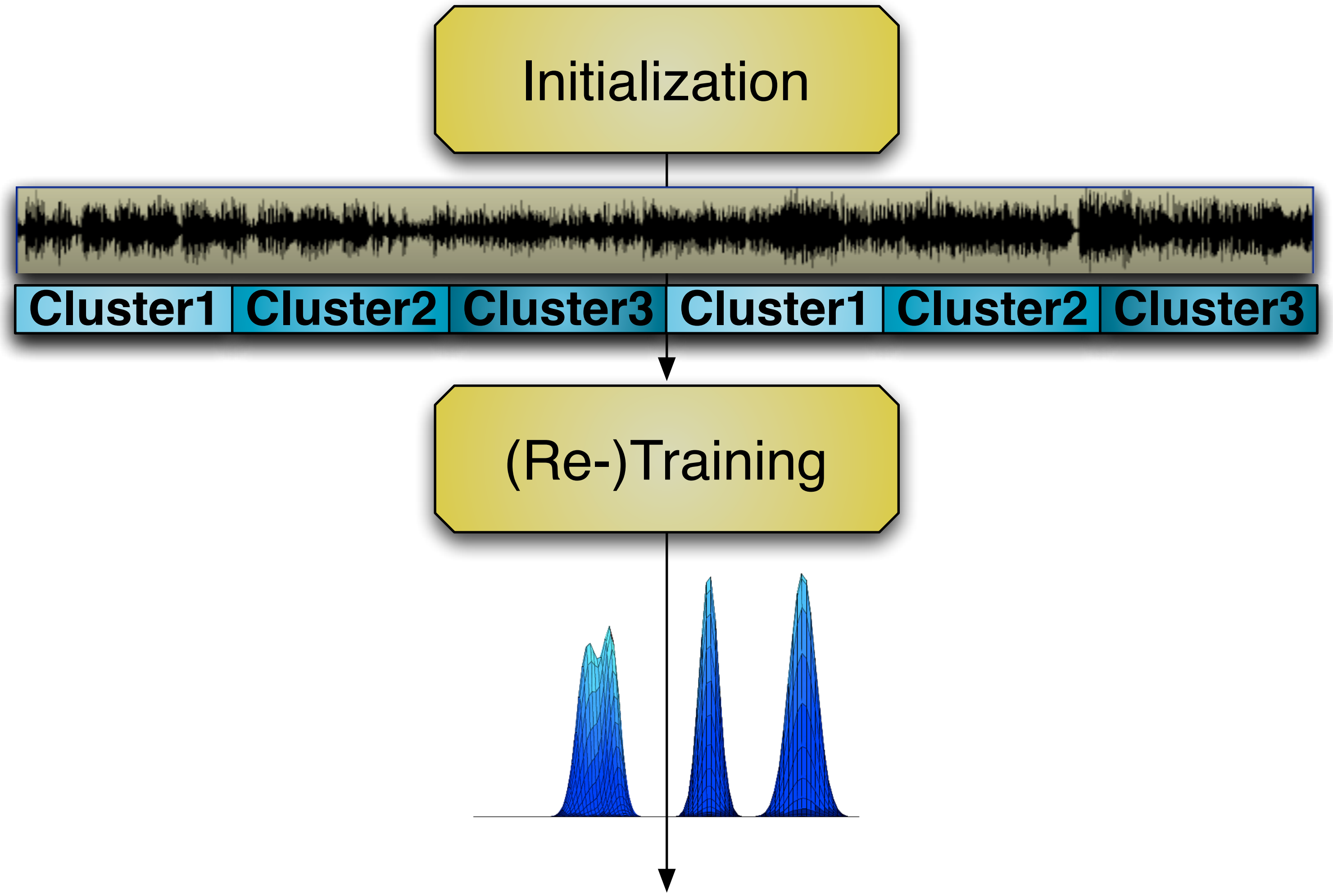
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



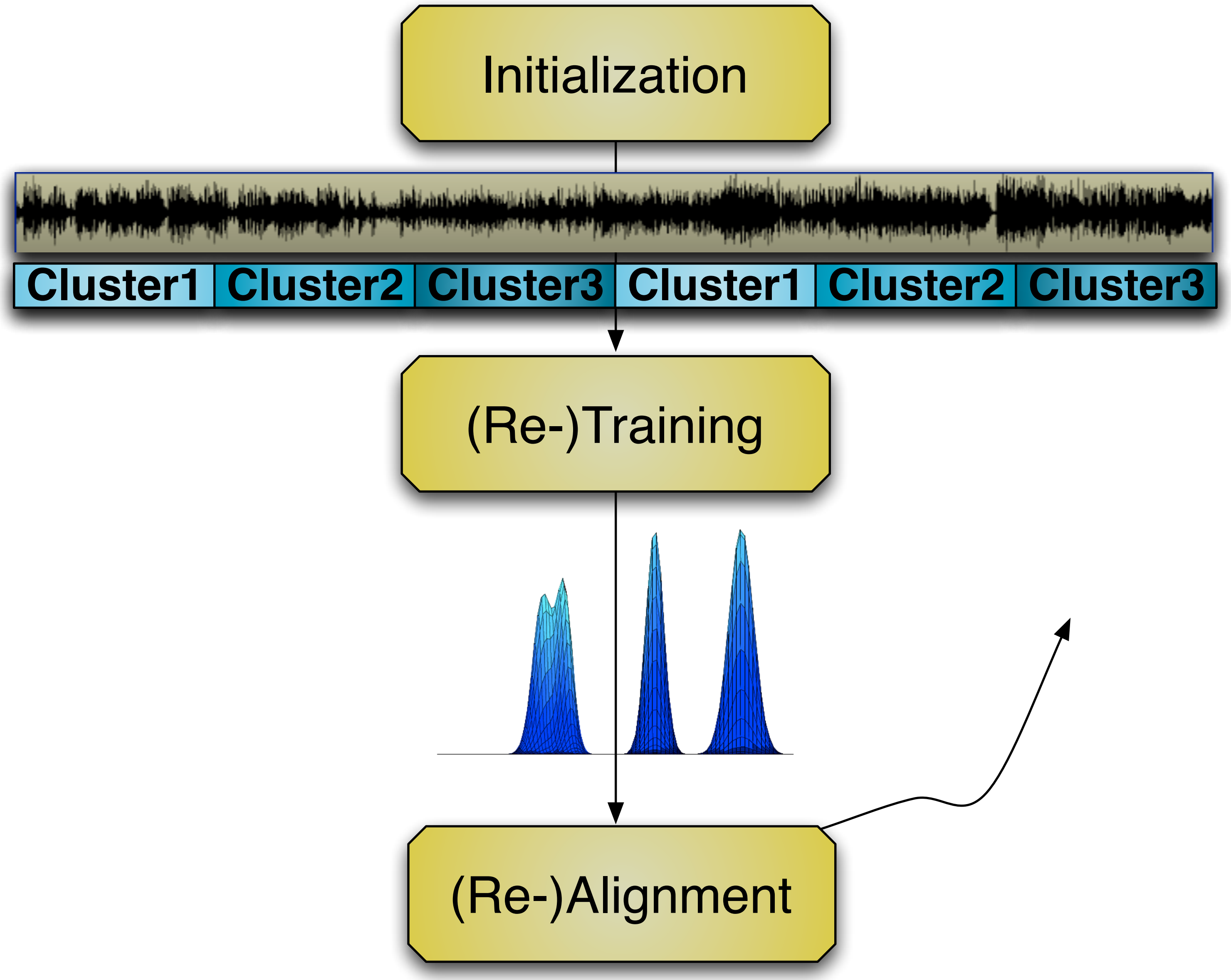
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



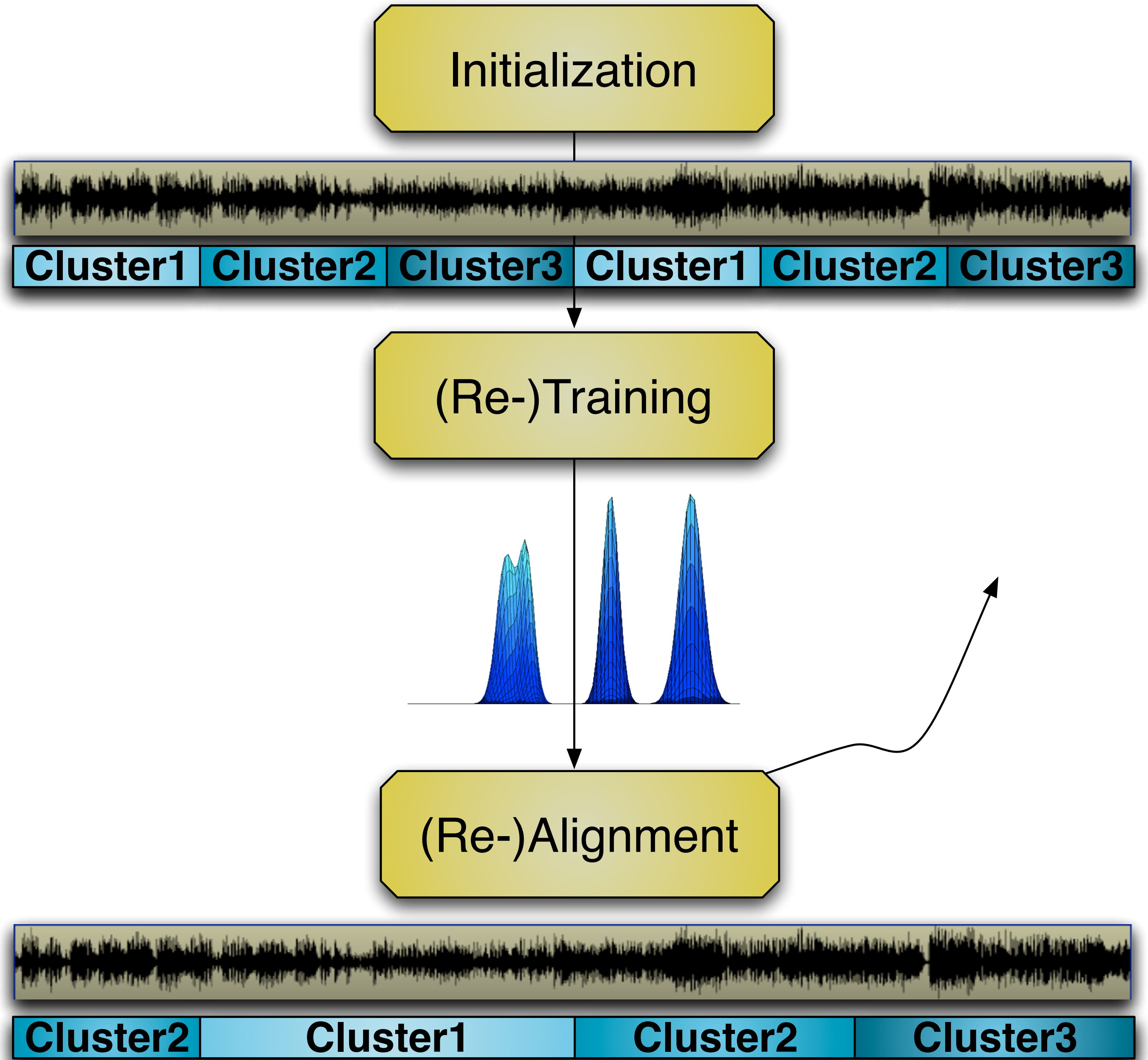
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



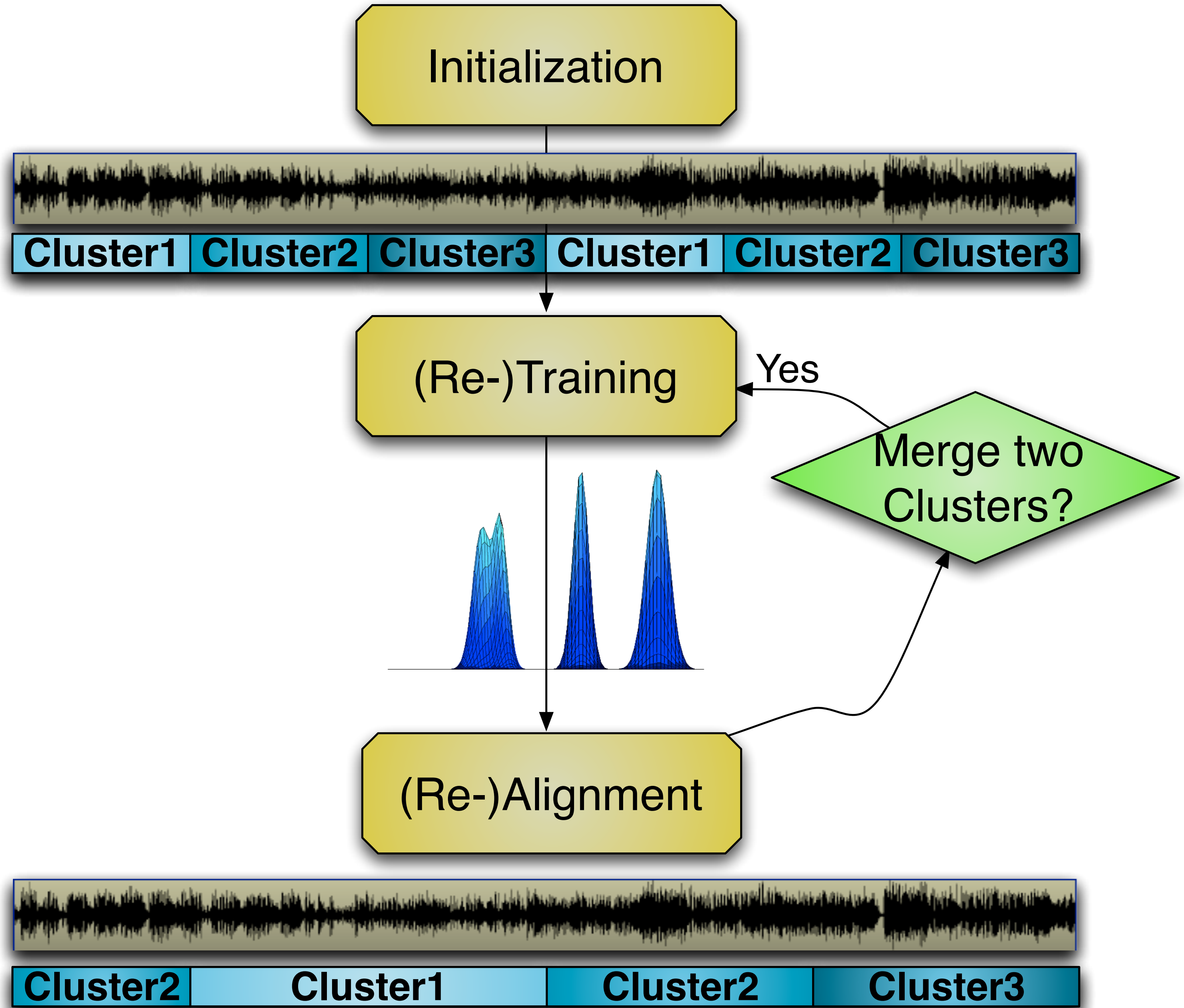
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



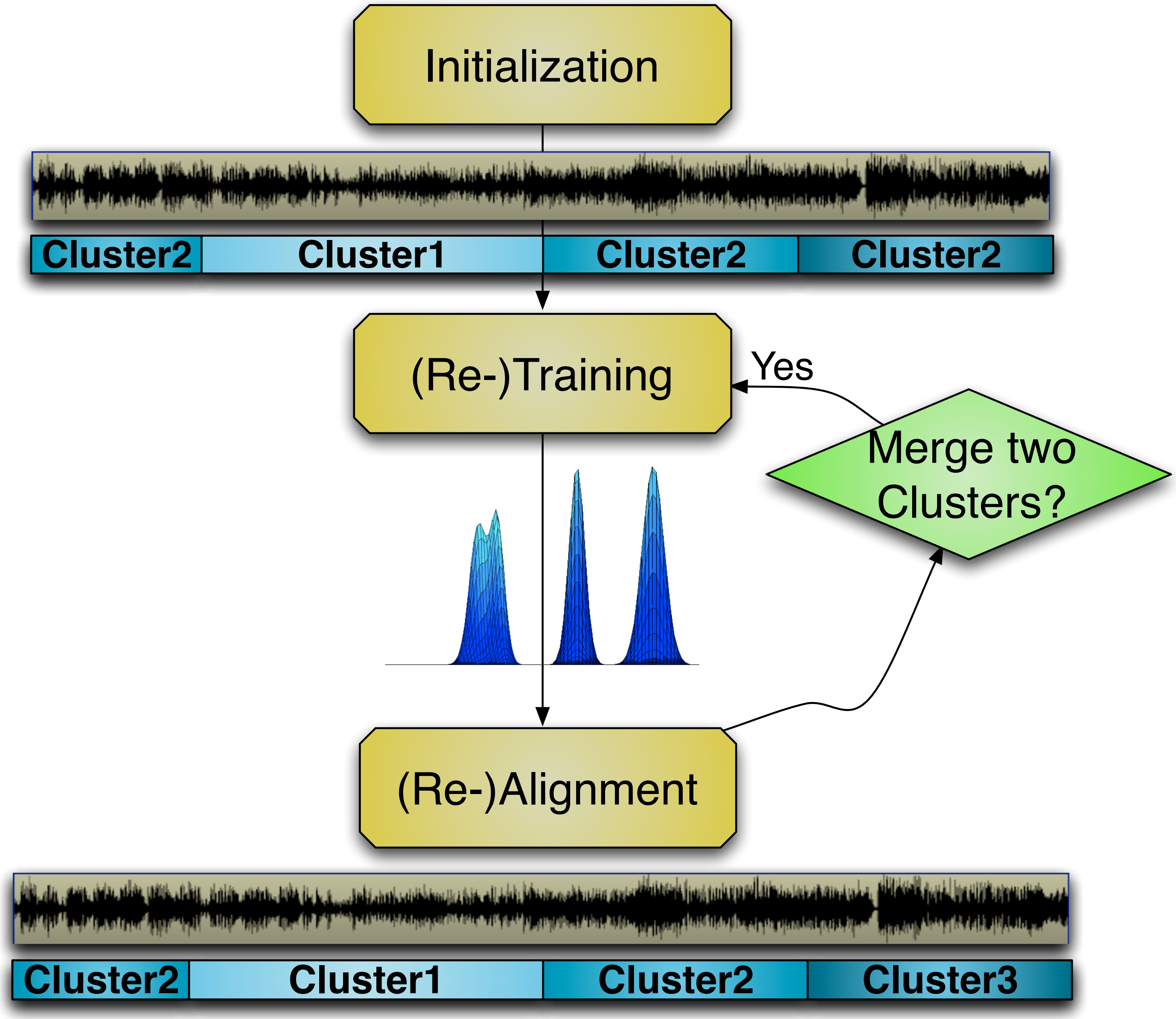
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



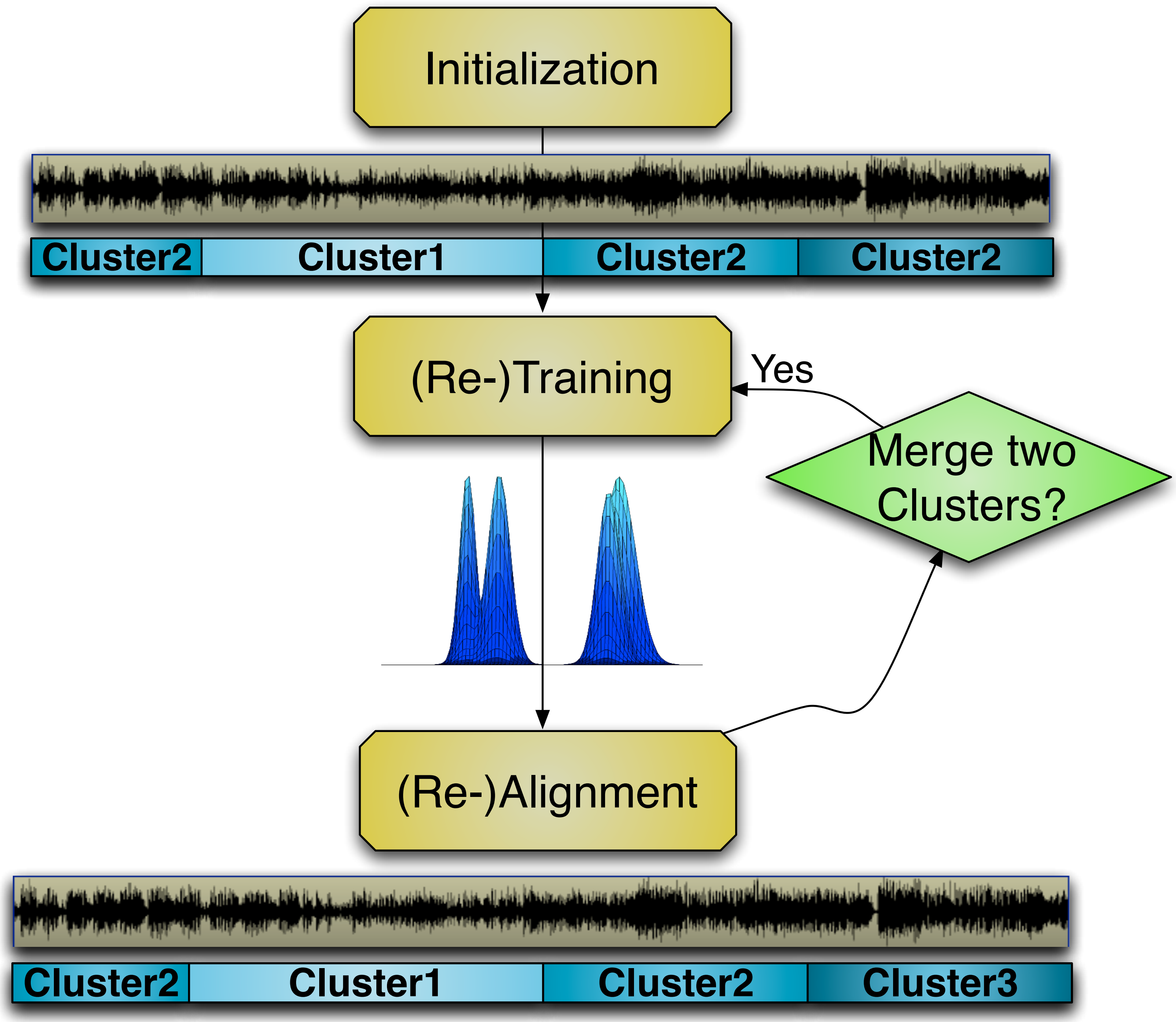
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



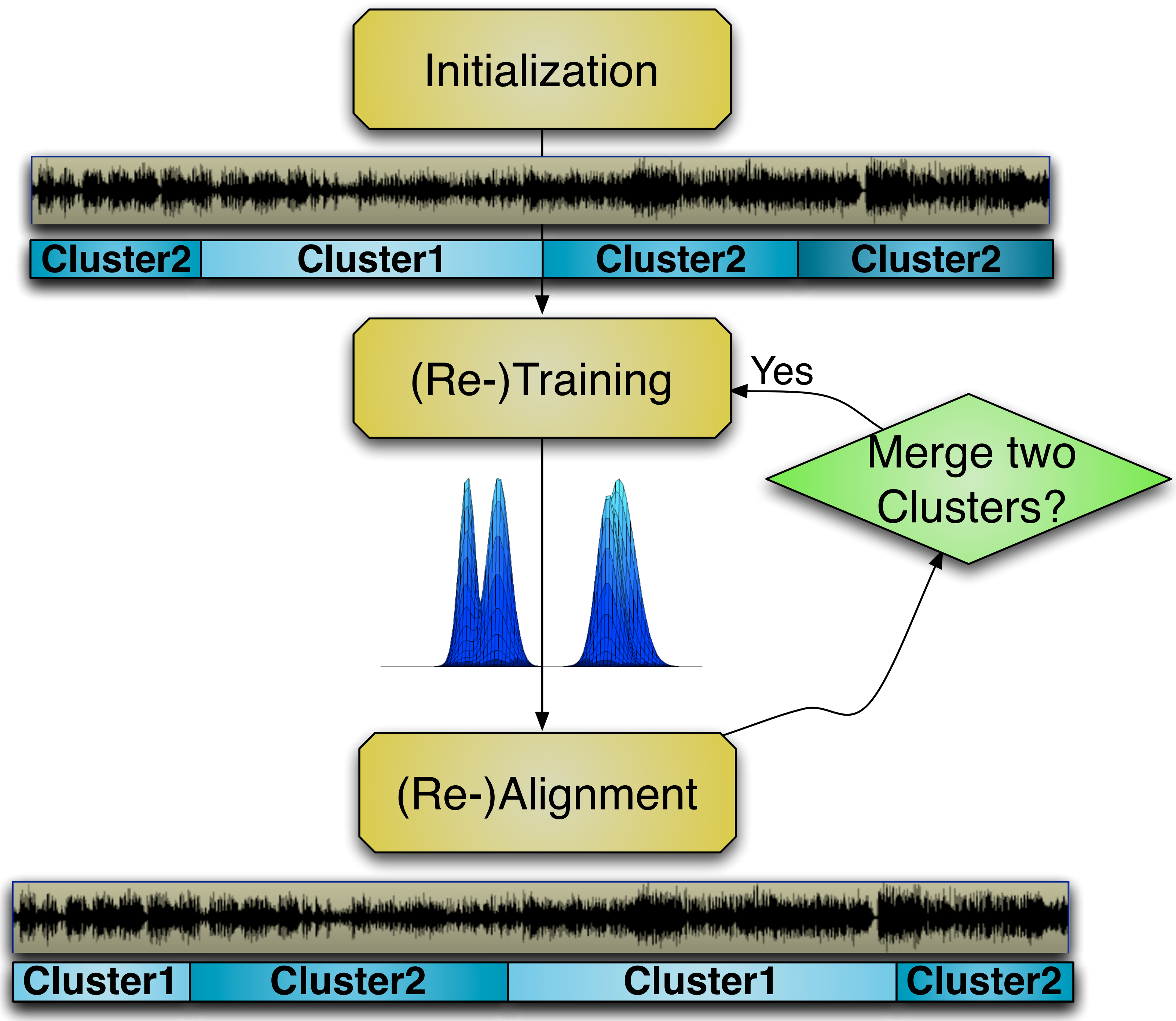
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



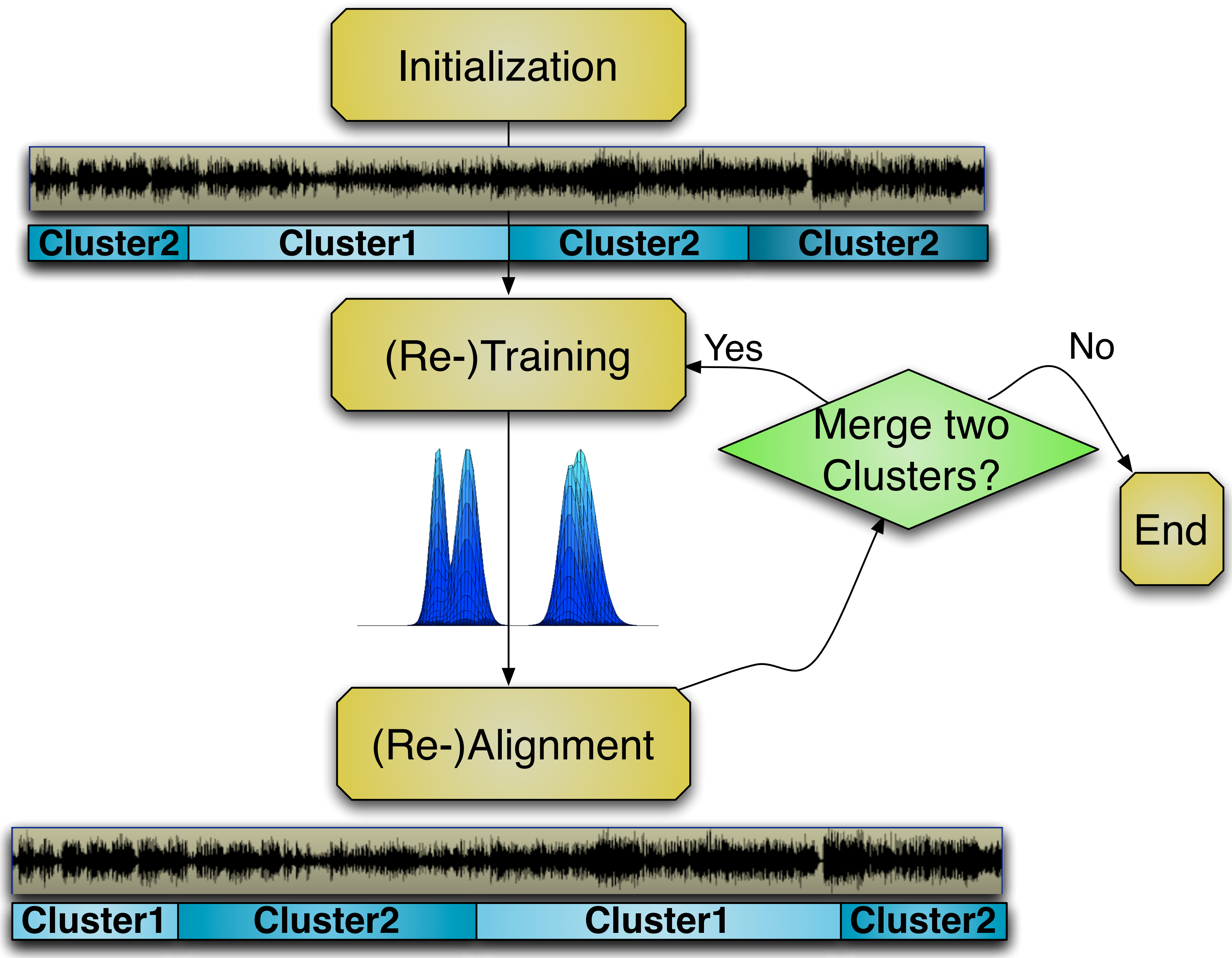
- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

Bottom-Up Algorithm



- Start with too many clusters (initialized randomly)
- Purify clusters by comparing and merging similar clusters
- Resegment and repeat until no more merging needed

ICSI's Speaker Diarization

ICSI's Speaker Diarization

- Speaker Diarization research @ ICSI since 2001

ICSI's Speaker Diarization

- Speaker Diarization research @ ICSI since 2001
- Various versions of Diarization Engines developed over the years

ICSI's Speaker Diarization

- Speaker Diarization research @ ICSI since 2001
- Various versions of Diarization Engines developed over the years
- Status: Research code but stable for some applications that are error tolerant

ICSI's Speaker Diarization Engine Variants

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)
- Fast (single mic, multiple CPU cores)

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)
- Fast (single mic, multiple CPU cores)
- Super fast (single mic, multiple GPUs)

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)
- Fast (single mic, multiple CPU cores)
- Super fast (single mic, multiple GPUs)
- Accurate but slow (multi mic, additional preprocessing)

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)
- Fast (single mic, multiple CPU cores)
- Super fast (single mic, multiple GPUs)
- Accurate but slow (multi mic, additional preprocessing)
- Audio/Visual (single and multi mic, for localization)

ICSI's Speaker Diarization Engine Variants

- Basic (single mic, easy installation)
- Fast (single mic, multiple CPU cores)
- Super fast (single mic, multiple GPUs)
- Accurate but slow (multi mic, additional preprocessing)
- Audio/Visual (single and multi mic, for localization)
- Online (single mic, "who is speaking now")

Basic Speaker Diarization: Facts

Basic Speaker Diarization: Facts

- Input: 16kHz mono audio

Basic Speaker Diarization: Facts

- Input: 16kHz mono audio
- Features: MFCC19, no delta or deltadelta

Basic Speaker Diarization: Facts

- Input: 16kHz mono audio
- Features: MFCC19, no delta or deltadelta
- Speech/Non-Speech Detector external

Basic Speaker Diarization: Facts

- Input: 16kHz mono audio
- Features: MFCC19, no delta or deltadelta
- Speech/Non-Speech Detector external
- Runtime: ~ realtime (1h audio needs 1h processing on a single CPU, excluding speech/non-speech)

Multi-CPU Speaker Diarization: Facts

Multi-CPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization

Multi-CPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: Dependent on number of CPUs used.
  Example: 8 cores runtime = 14.3 x realtime, i.e. 14minutes of audio need 1 minute of processing.

Multi-CPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: Dependent on number of CPUs used.
  Example: 8 cores runtime = 14.3 x realtime, i.e. 14minutes of audio need 1 minute of processing.
- Runtime bottleneck usually: Speech/ Non-Speech Detector

GPU Speaker Diarization: Facts

GPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization

GPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: 250 x realtime, i.e. 1h of audio is processed in 14.4sec!

GPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: 250 x realtime, i.e. 1h of audio is processed in 14.4sec!
- Uses current CUDA NVidia Framework as backend.

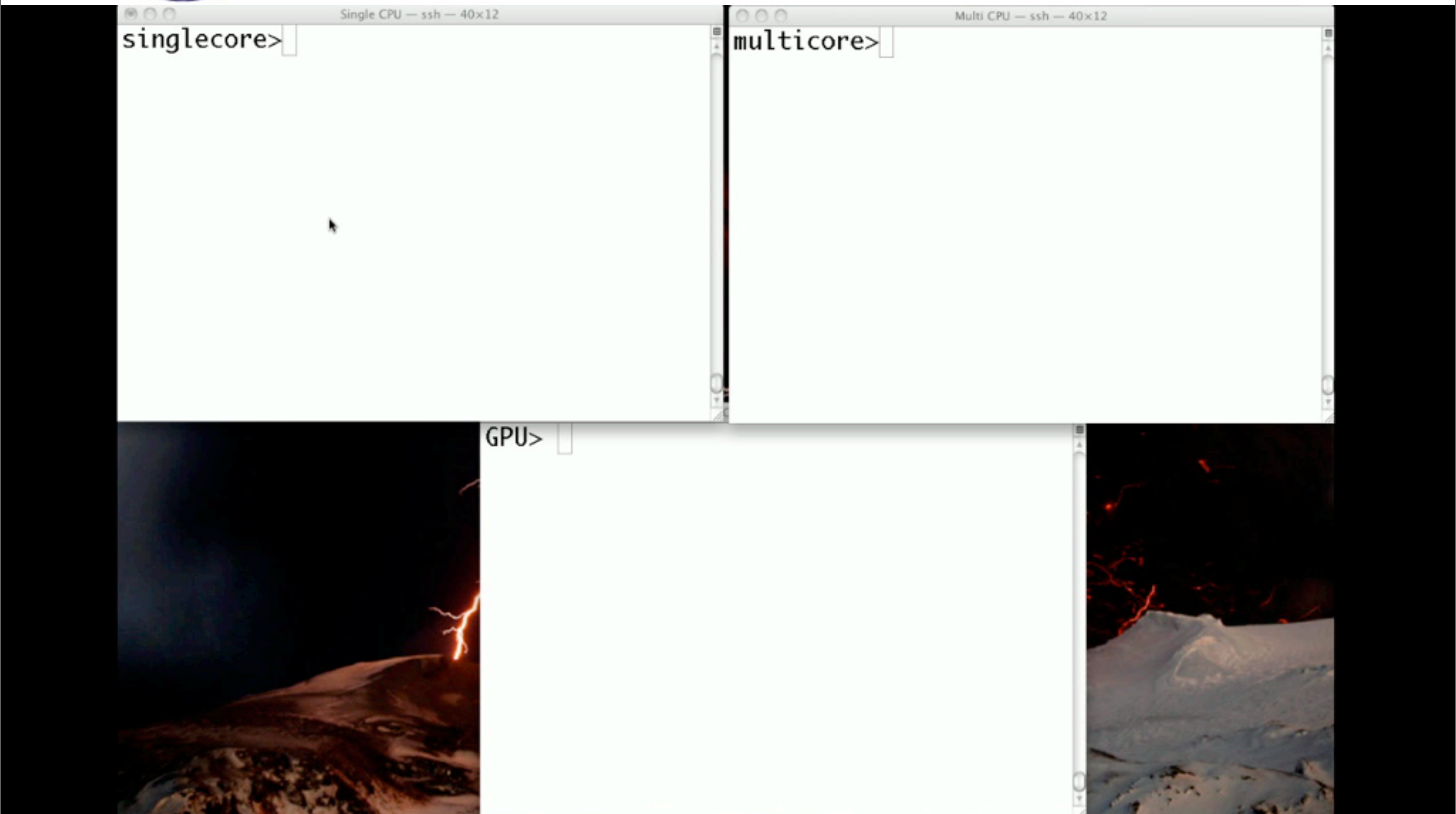
GPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: 250 x realtime, i.e. 1h of audio is processed in 14.4sec!
- Uses current CUDA NVidia Framework as backend.
- Frontend: Python!

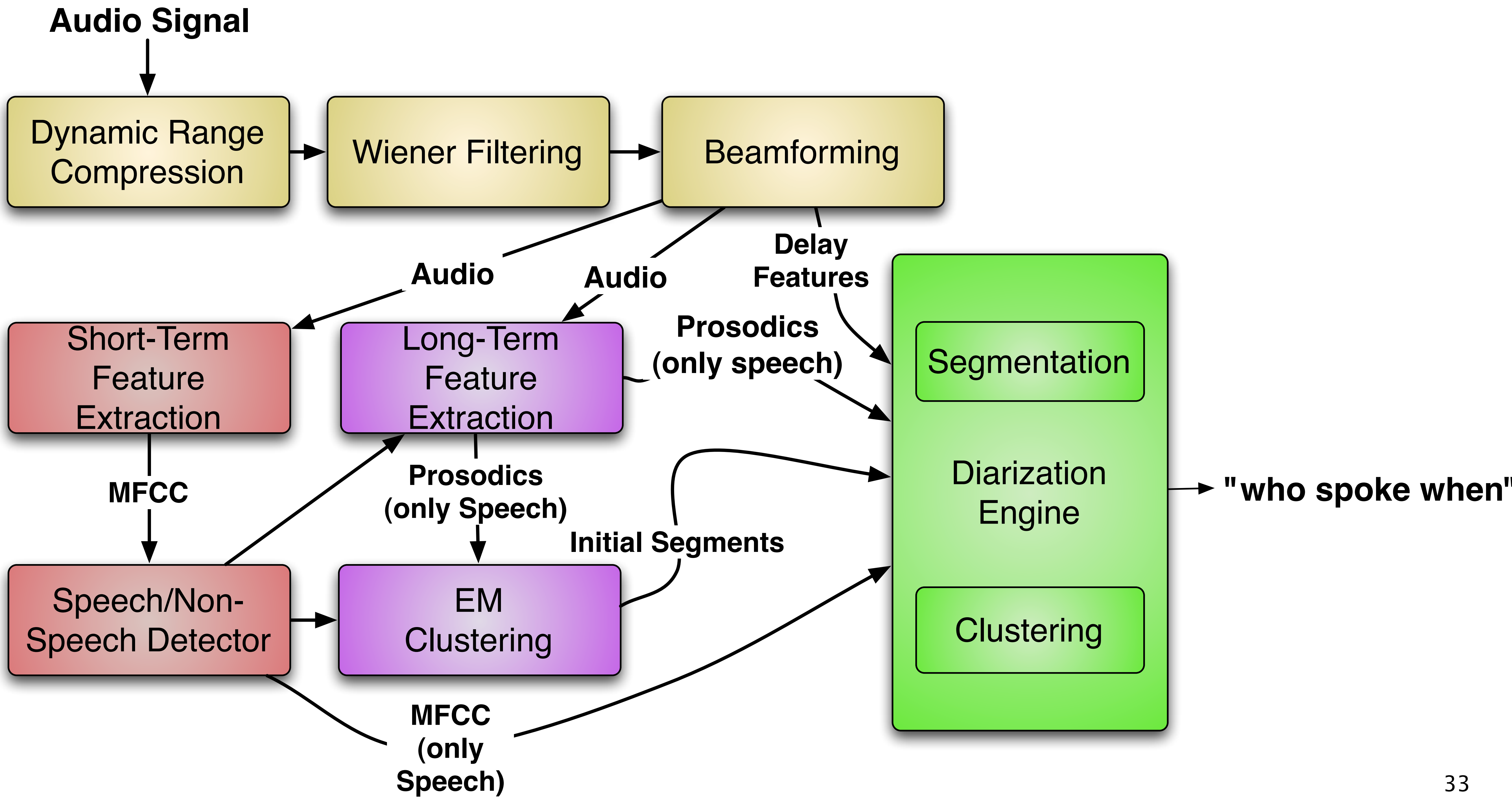
GPU Speaker Diarization: Facts

- Same as Basic Speaker Diarization
- Runtime: 250 x realtime, i.e. 1h of audio is processed in 14.4sec!
- Uses current CUDA NVidia Framework as backend.
- Frontend: Python!
- Runtime bottleneck usually: Speech/ Non-Speech Detector, Feature Extraction

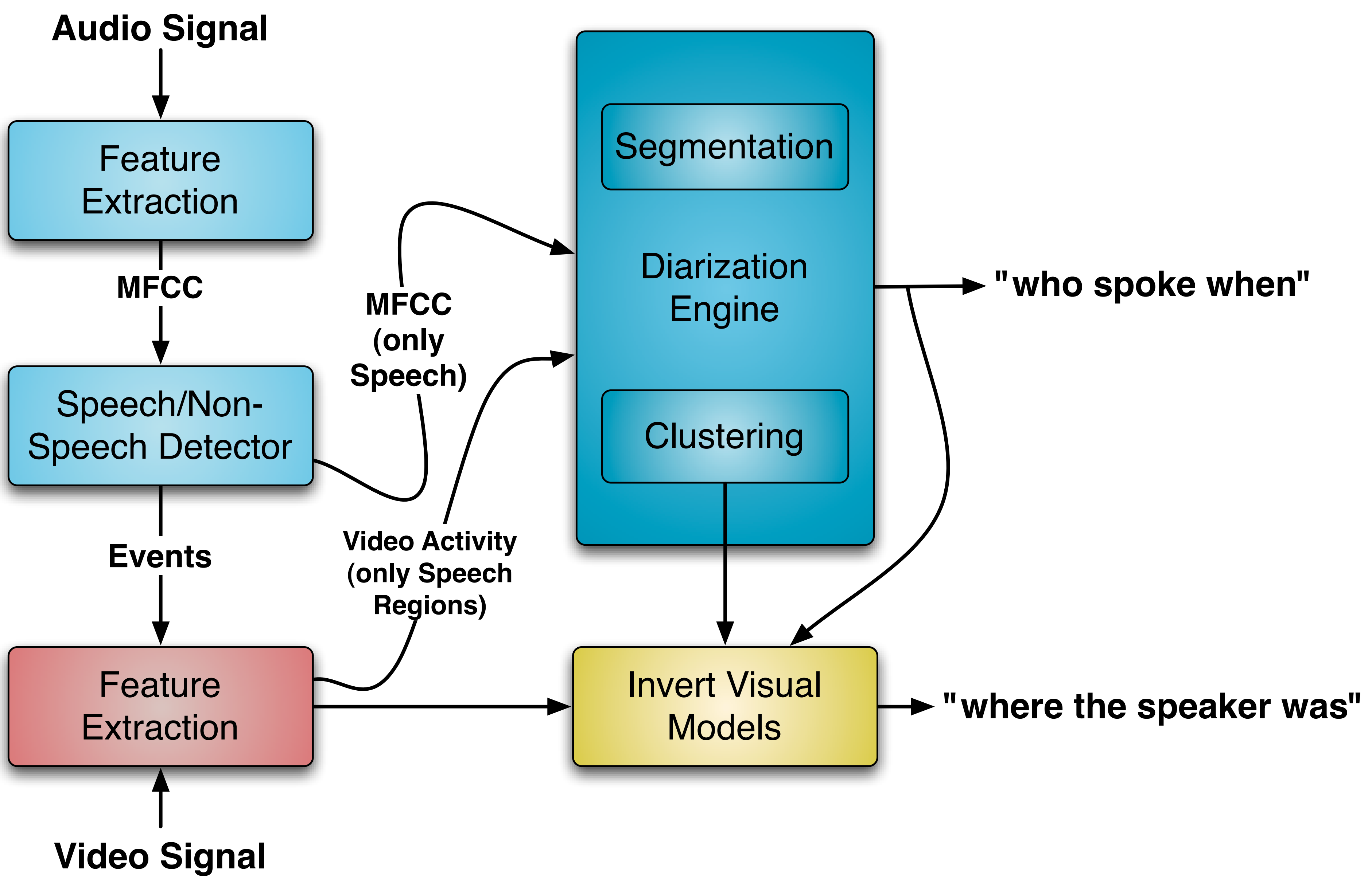
Demo: 1CPU vs 8CPU vs GPU



Most Accurate Speaker Diarization: Overview



Audio/Visual Speaker Diarization: Overview



Video Feature Extraction

MPEG-4 Video → Detect Skin Blocks → Avg. Motion Vectors → Divide Frames into n Regions → **n-dimensional activity vector**

Windowsize: 400ms

Audio/Visual Speaker Diarization: Facts

Audio/Visual Speaker Diarization: Facts

- **One** engine for audio and video

Audio/Visual Speaker Diarization: Facts

- **One** engine for audio and video
- Scales with n cameras

Audio/Visual Speaker Diarization: Facts

- **One** engine for audio and video
- Scales with n cameras
- Robust against visual changes such as different cloth, occlusions, etc...

  "A voiceprint does not care about somebody dimming the light"

ICSI
INTERNATIONAL
COMPUTER SCIENCE
INSTITUTE

Audio/Visual Diarization: Example Video

In a perfect world...

In a perfect world...

- There is no overlapped speech

In a perfect world...

- There is no overlapped speech
- The signal is clean

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male – female, young – old)
- Speakers are non-emotional

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)
- Speakers are non-emotional
- Recording is at 16kHz.

In a perfect world...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)
- Speakers are non-emotional
- Recording is at 16kHz.
- Recording is 15-60 minute length

Current Results using Different Inputs

Error/System	Basic System: 1 Audio Stream	8 Audio Streams	1 Audio Stream + 1 Camera	1 Audio Stream + 4 Cameras
Diarization Error Rate	32.09%	27.55%	27.52%	24.00%
Relative Improvement	baseline	14%	14%	25%
Core Speed (x realtime)	1.0	2.2	1.4	1.3

12 Meeting Recordings from AMI corpus

Most Accurate Results

Error/System	MFCC only (basic system)	Full System	Full System + One Camera
Diarization Error Rate	32.09%	20.33%	18.98%
Relative Improvement	baseline	36%	41%
Core Speed (x realtime)	1.0	2.5	2.9

12 Meetings from AMI corpus "VACE Meetings"

Top Error Sources

Top Error Sources

- Overlapped Speech

Top Error Sources

- Overlapped Speech
- Short Speech Segments (<2s)

Top Error Sources

- Overlapped Speech
- Short Speech Segments (<2s)
- Environmental Noise

Top Error Sources

- Overlapped Speech
- Short Speech Segments (<2s)
- Environmental Noise
- Low SNR

Top Error Sources

- Overlapped Speech
- Short Speech Segments (<2s)
- Environmental Noise
- Low SNR
- Bad Speech/Non-Speech Detector performance based on training data mismatch

Top Error Sources

- Overlapped Speech
- Short Speech Segments (<2s)
- Environmental Noise
- Low SNR
- Bad Speech/Non-Speech Detector performance based on training data mismatch
- Parameter mismatch, e.g. too few initial clusters

Optimal Performance is achieved when...

Optimal Performance is achieved when…

- There is no overlapped speech

Optimal Performance is achieved when...

- There is no overlapped speech
- The signal is clean

Optimal Performance is achieved when…

- There is no overlapped speech
- The signal is clean
- No environmental noise

Optimal Performance is achieved when...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)

Optimal Performance is achieved when...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)

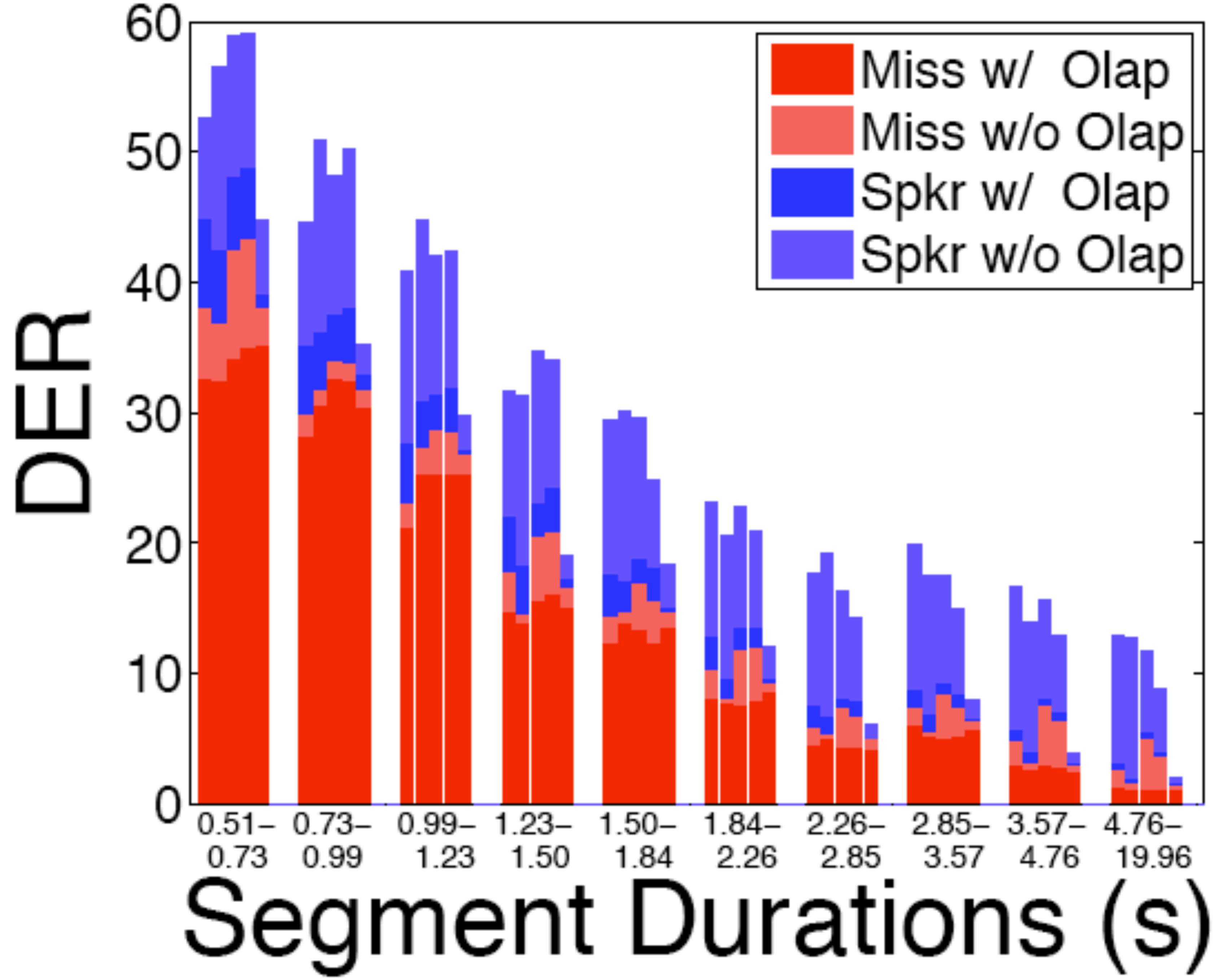
Optimal Performance is achieved when...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)
- Speakers are non-emotional

Optimal Performance is achieved when...

- There is no overlapped speech
- The signal is clean
- No environmental noise
- Limited amount of speakers (4 or so)
- Speaker are well-distinguishable in their voice (e.g. male - female, young - old)
- Speakers are non-emotional
- Recording is at 16kHz or higher.

Future Work!



DER
60
50
40
30
20
10
0
Miss w/ Olap
Miss w/o Olap
Spkr w/ Olap
Spkr w/o Olap
0.51– 0.73
0.73– 0.99
0.99– 1.23
1.23– 1.50
1.50– 1.84
1.84– 2.26
2.26– 2.85
2.85– 3.57
3.57– 4.76
4.76– 19.96
Segment Durations (s)

Thank You!
Questions?

Some of the Presented Work
performed together with:
Mary Knox, Katya Gonina, Adam Janin
and others.