Music Recognition
(using computer vision!)

Rahul Sukthankar
Intel Labs Pittsburgh &
Carnegie Mellon

Collaborators: Y. Ke, D. Hoeim, L. Yang

Recognition

- Let's agree on some terminology
  - object detection
  - recognition – instance vs. category
  - localization
  - classification vs. retrieval

- Examples of such tasks in vision and audio
- Key research challenges for each task

Popular Vision Techniques

- Recent successes in computer vision
  - Windowed object detectors
  - Local features for object recognition (e.g., SIFT)
  - Boosted classifiers (e.g., Viola-Jones face detector)
  - Sub-image retrieval
  - RANSAC geometric verification
  - Structure from motion

Computer Vision for *Audio?!*

- Recent successes in computer vision in audio domain
  - Windowed object detectors — sound obj det, music vs sound
  - Local feature object recognition — MusicID, sound object detect
  - Boosted classifiers — MusicID, sound object detect
  - Sub-image retrieval — MusicID
  - RANSAC geometric verification — MusicID
  - Structure from motion — affine structure from sound [Thrun, NIPS 2005]

- Claim: many vision ideas map naturally to audio domain

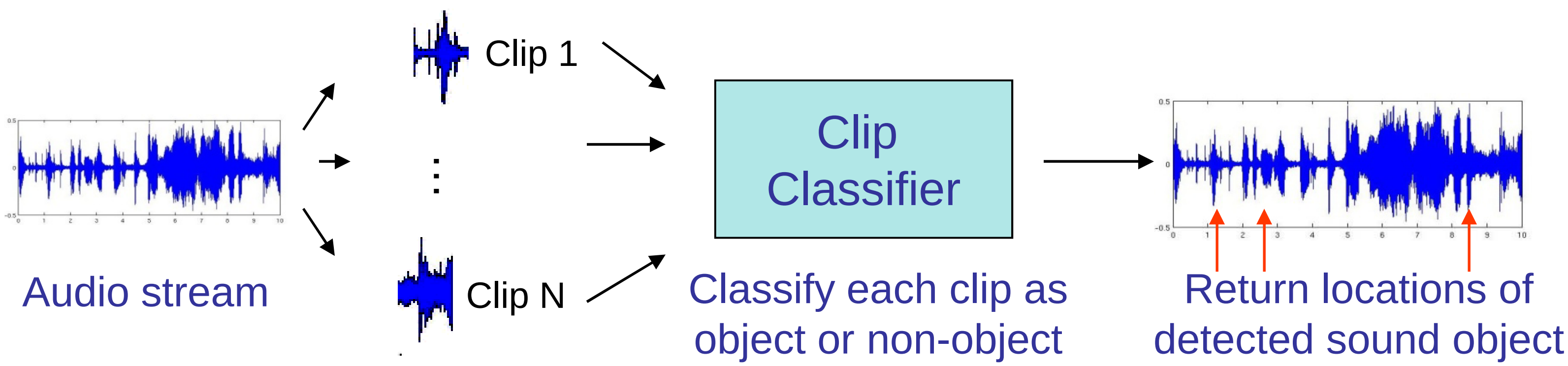
Outline

- Sound object detection
  (localizing a known sound in audio stream)

- Music identification
  (match audio snippet against large DB of songs)

Sound Object Detection in Movies

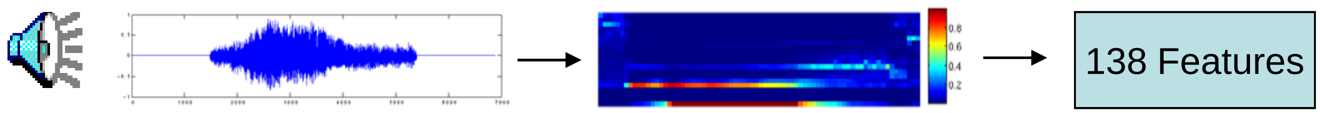
- Applications of sound object detection
  - “Tell me if you hear a gunshot.” (monitoring)
  - “Fast forward to the swordfight” (search and retrieval)
- Computer vision analogy: object detection/localization in images
  - Learn classifier from instances of the object
  - Scan windowed classifier over all possible locations



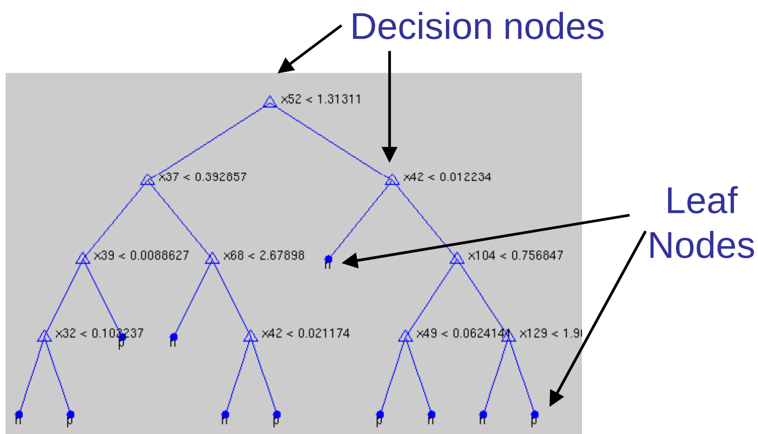
[Hoiem, Ke, Sukthankar, 2005]

Sound Object Detection: Clip Classifier

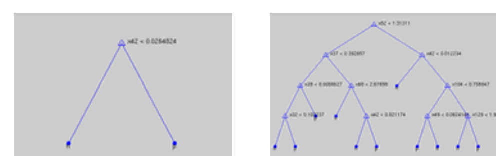
- Feature extraction



- Weak classifier – small decision trees on features



- Learn classifier cascade using Adaboost



…

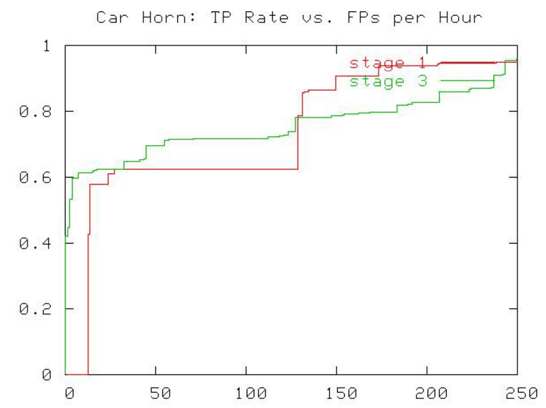
[Hoiem, Ke, Sukthankar, 2005]

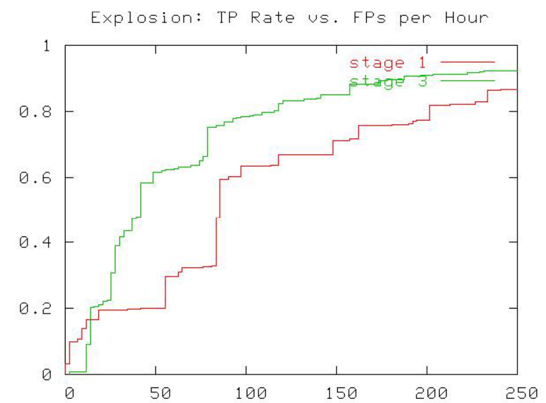
Sound Object Detection: Results

Best Performance

Worst Performance

	stage 1		stage 2		stage 3	
	pos	neg	pos	neg	pos	neg
meow	0.0%	1.4%	0.0%	1.2%	2.2%	0.8%
phone	0.0%	0.4%	4.3%	0.1%	5.9%	0.0%
car horn	0.0%	3.9%	0.6%	2.2%	3.6%	1.3%
door bell	1.4%	2.1%	2.1%	0.4%	6.3%	0.1%
swords	6.1%	1.3%	6.7%	0.1%	6.7%	0.0%
scream	0.3%	5.5%	2.7%	1.4%	5.3%	1.1%
dog bark	0.7%	1.0%	6.0%	0.3%	7.7%	0.2%
laser gun	0.0%	6.8%	4.4%	5.1%	6.7%	0.9%
explosion	4.1%	5.2%	7.5%	1.5%	12.0%	0.5%
light saber	4.8%	6.8%	9.7%	1.0%	13.9%	0.2%
gunshot	8.1%	6.1%	12.5%	2.3%	14.5%	1.1%
close door	7.9%	7.8%	14.5%	4.8%	17.6%	2.3%
male laugh	4.3%	14.7%	9.5%	9.7%	13.3%	7.0%
average	**2.9%**	**4.4%**	**6.0%**	**2.2%**	**8.5%**	**1.1%**





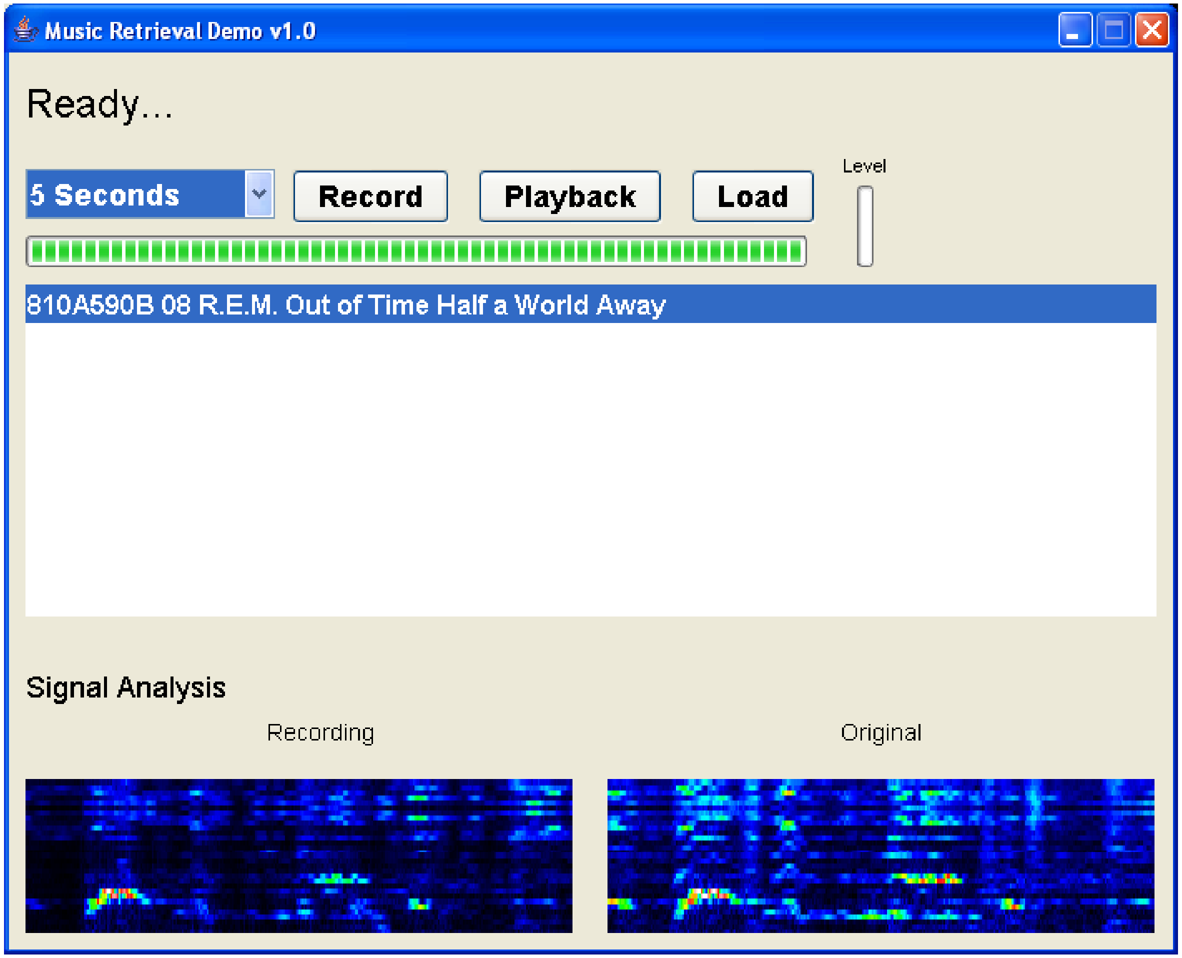
[Hoiem, Ke, Sukthankar, 2005]

Music Identification

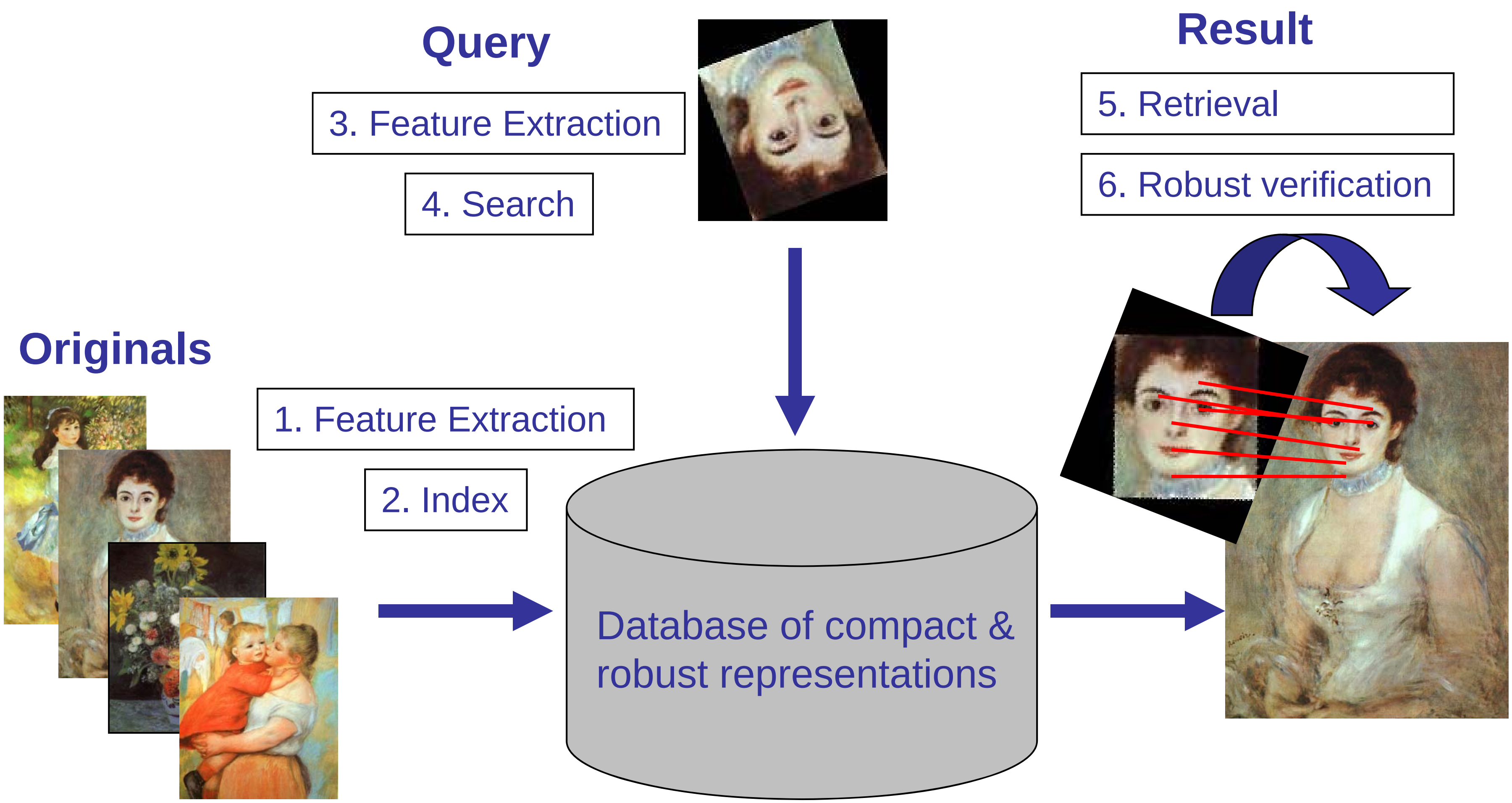
Music Identification: Challenges

- Query sample
  - is small (can't match complete song signatures)
  - can be taken from anywhere in the song
  - is typically noisy, distorted and occluded
- Database
  - contains large numbers of songs of varying genres
  - can be incrementally updated with new songs
- Performance:
  - demand high accuracy (in both precision and recall)
  - interactive query times
  - compact storage requirements

Live demo

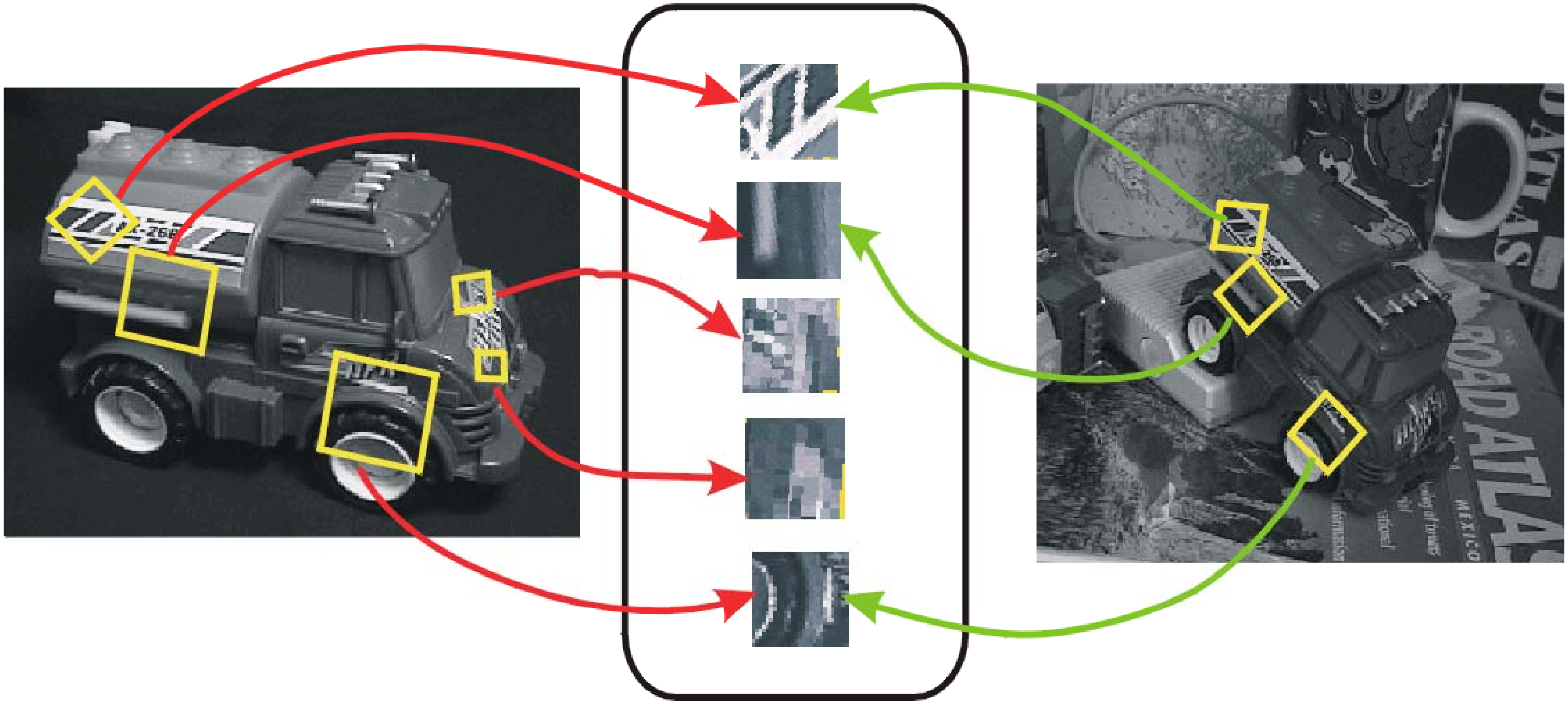

Music Retrieval Demo v1.0
Ready...
5 Seconds
Record
Playback
Load
Level
810A590B 08 R.E.M. Out of Time Half a World Away
Signal Analysis
Recording
Original

Similar Vision Task – Sub-Image Retrieval



[Ke & Sukthankar, ACM MM 2004]

Keypoints for Image Matching



SIFT images from [Lowe 1999]

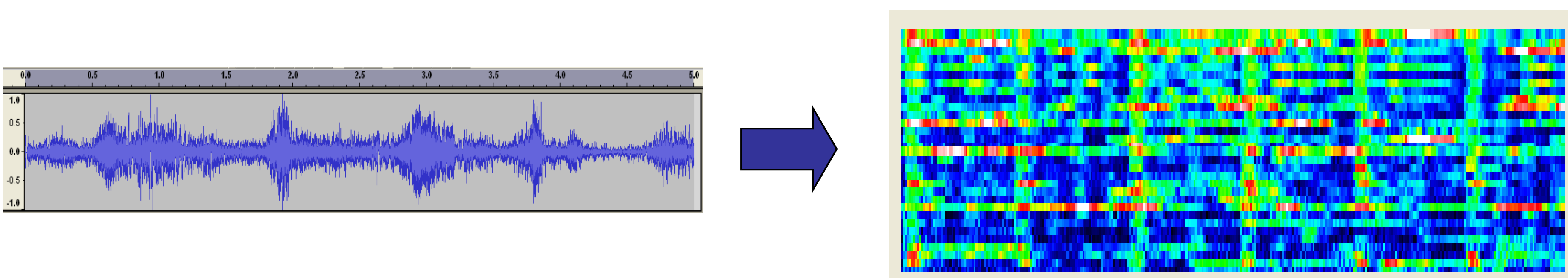
MusicID Algorithm

- Transform audio into spectrogram (2D image)
- Compute distinctive local descriptors (learned by pairwise boosting)
- Retrieve candidates using efficient index (near-neighbor in high-dim)
- Identify song using robust alignment (RANSAC + noise model)

[Ke, Hoiem, Sukthankar, CVPR 2005]

MusicID Algorithm

- Transform audio into spectrogram (2D image)
- Compute distinctive local descriptors (learned by pairwise boosting)
- Retrieve candidates using efficient index (near-neighbor in high-dim)
- Identify song using robust alignment (RANSAC + noise model)



[Ke, Hoiem, Sukthankar, CVPR 2005]

Name That Tune

Noisy recording

John Mellencamp – Suzanne and the Jewels

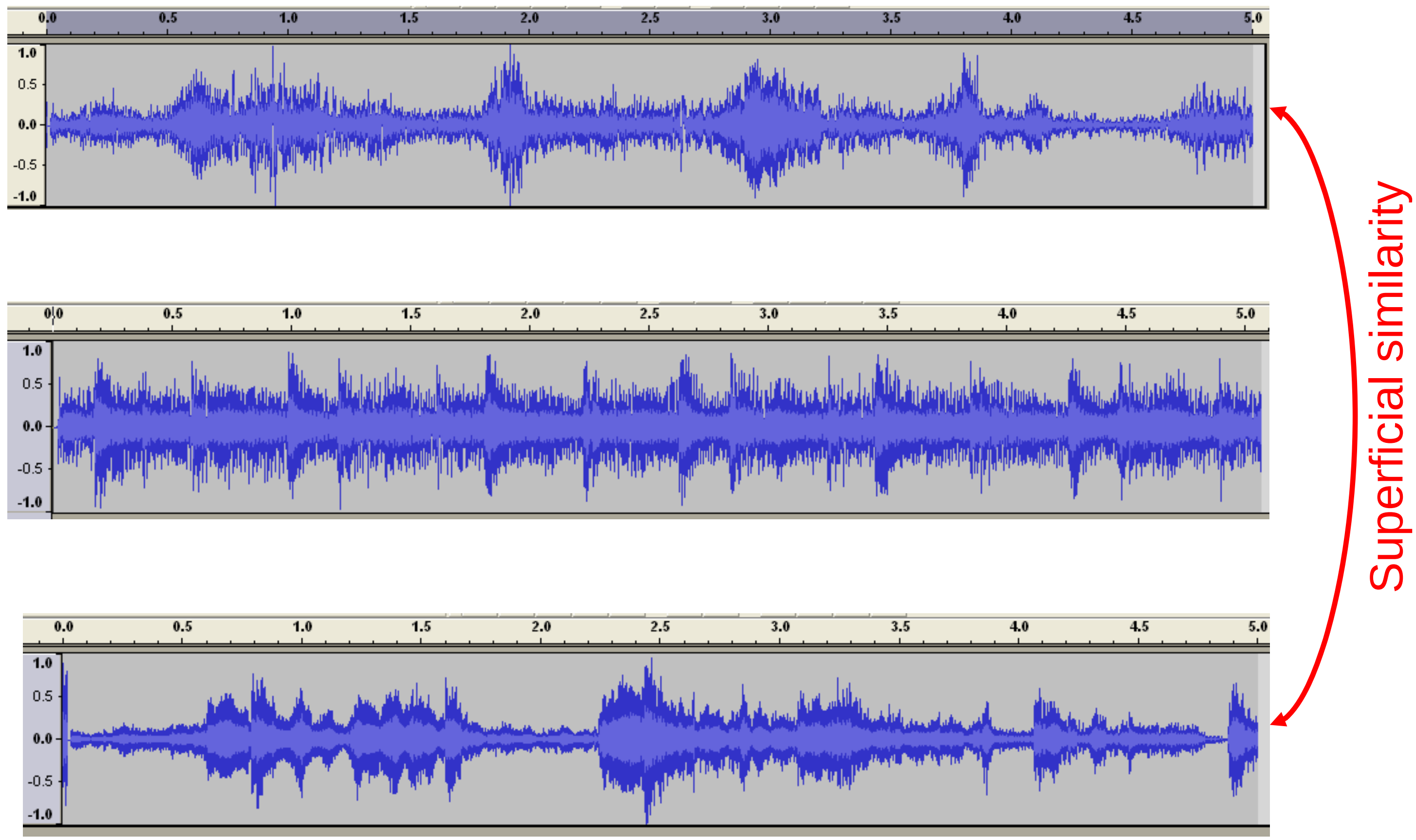
Waterworld soundtrack

Name That Tune: Raw Audio

Query (Mellencamp)

Mellencamp

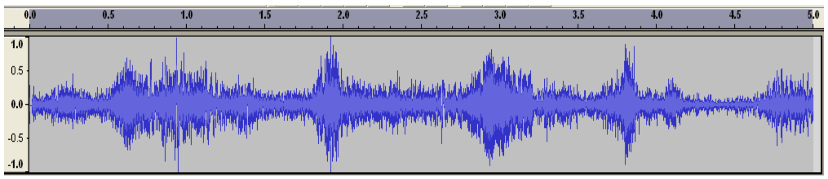
Waterworld



amplitude vs. time

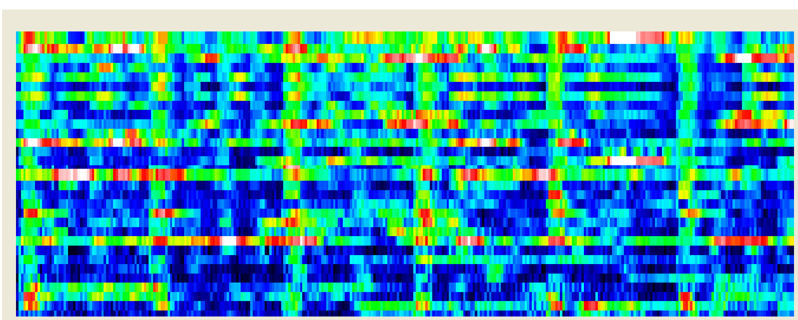
Spectrogram Representation

Raw audio



Spectrogram

Frequency (Hz)



Time (frames)

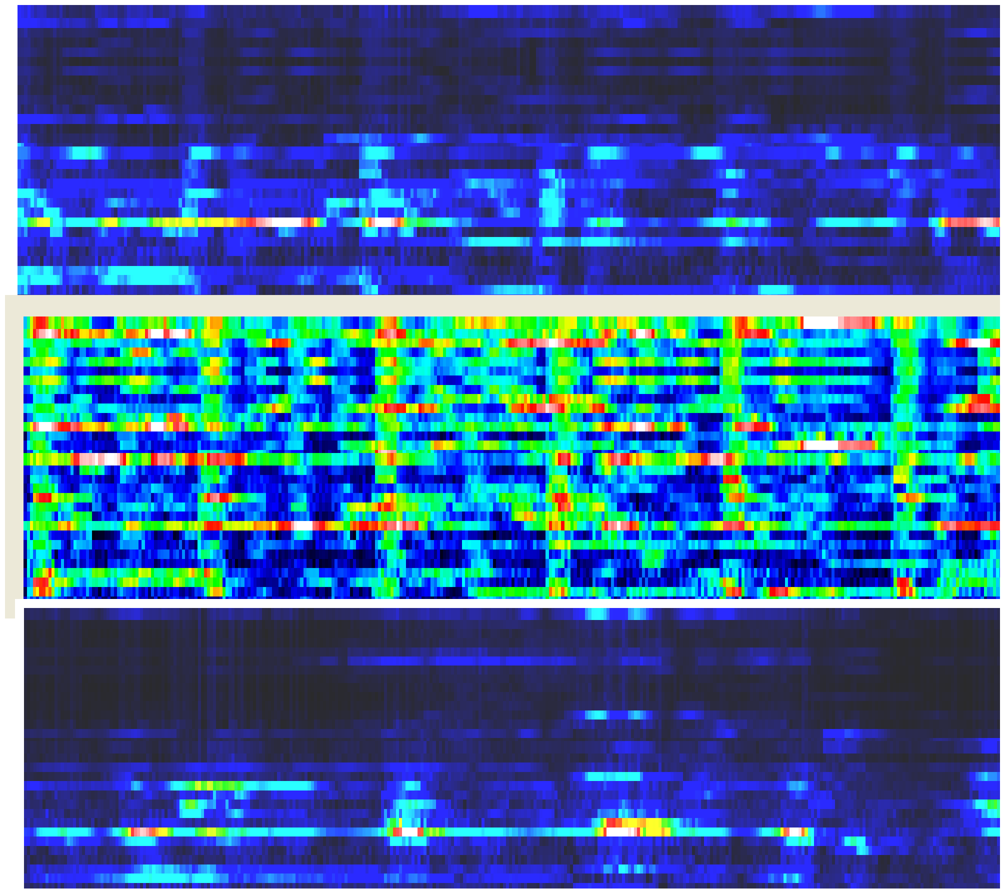
- 2D time-frequency image
- Short-term Fourier Transform on overlapping windows of 372ms at 11.6ms intervals
- Intensity shows power content in 33 logarithmically-spaced frequency bands
- Spectrograms are popular and have demonstrated good performance in several audio processing applications

Name That Tune: Spectrogram

Query
(Mellencamp)

Mellencamp

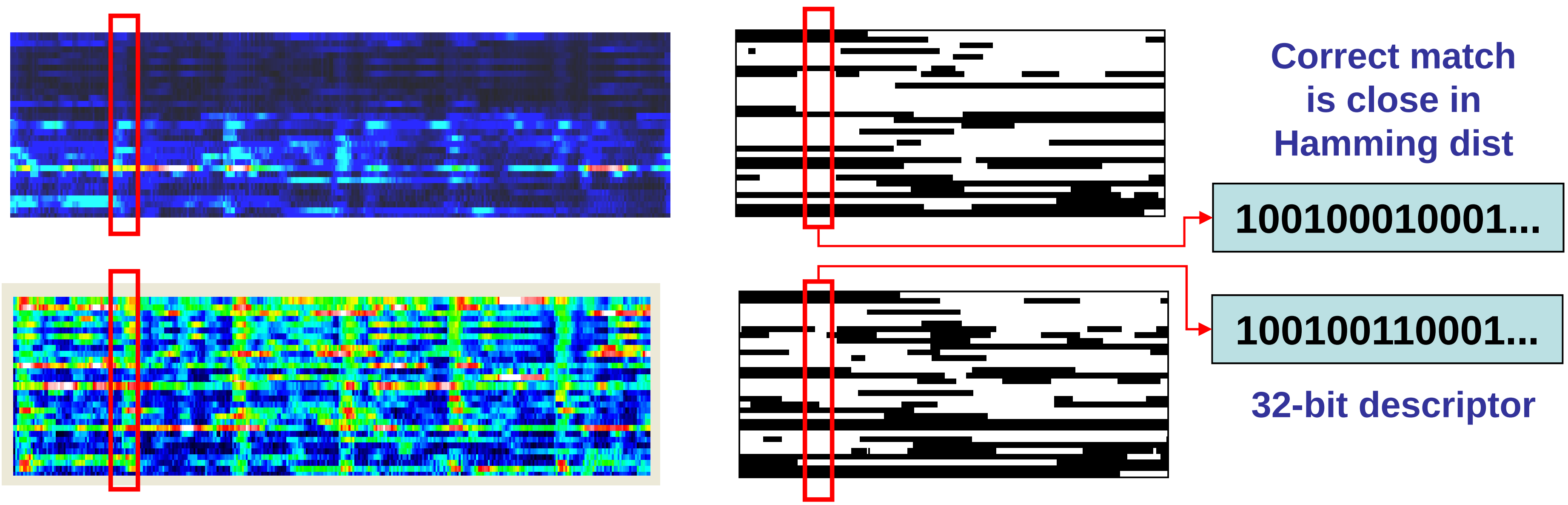
Waterworld



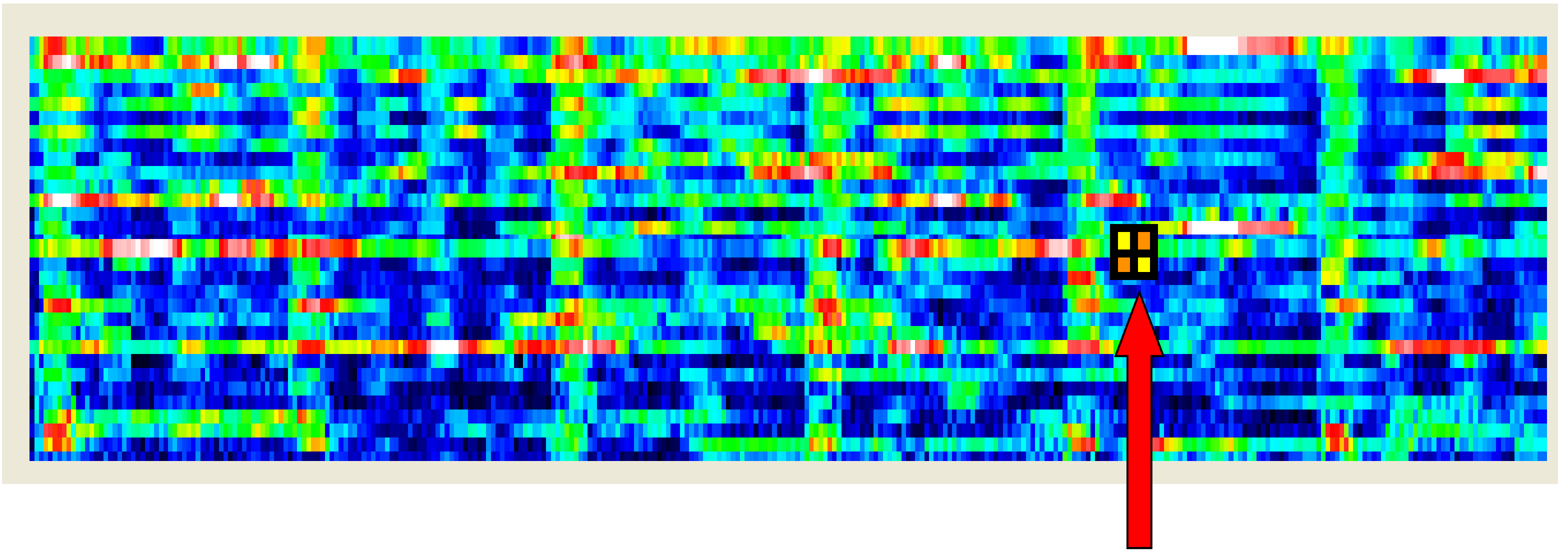
Spectrograms (frequency vs. time)

MusicID Algorithm

- Transform audio into spectrogram (2D image)
- Compute distinctive local descriptors (learned by pairwise boosting)
- Retrieve candidates using efficient index (near-neighbor in high-dim)
- Identify song using robust alignment (RANSAC + noise model)

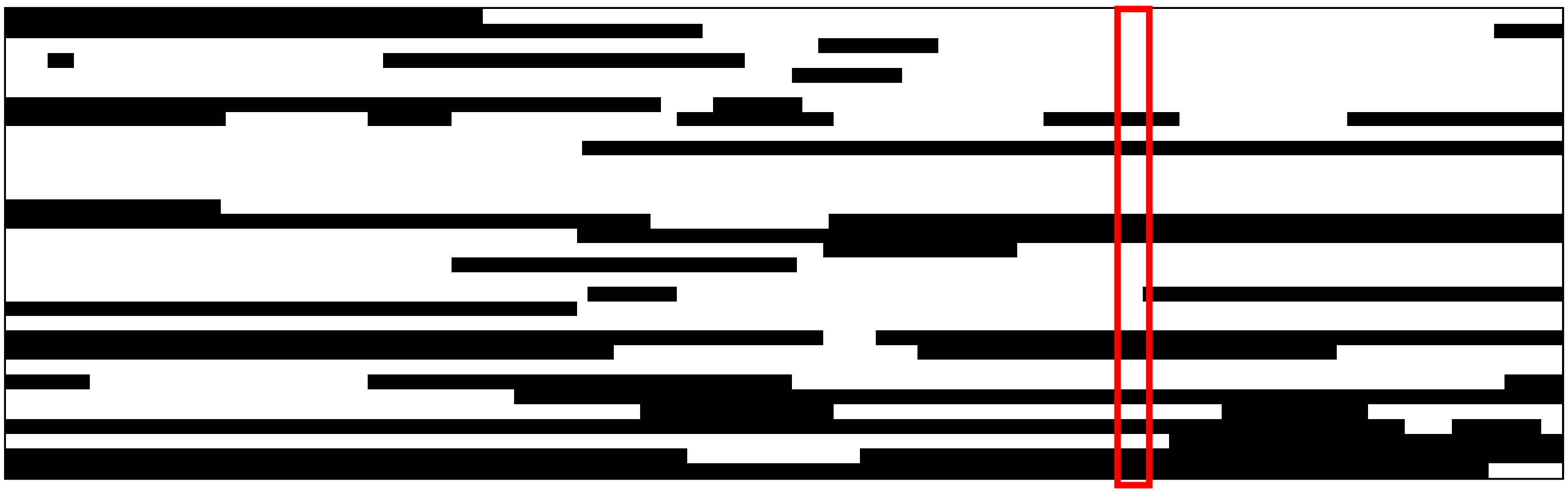


Motivation: [Haitsma & Kalker]



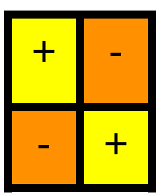
- At every frame & frequency band, compute: $\dfrac{d^2E}{dTdF}$
- Threshold at 0 to get a 32-bit descriptor at every time frame

[Haitsma & Kalker] Descriptor



000000101000011000…

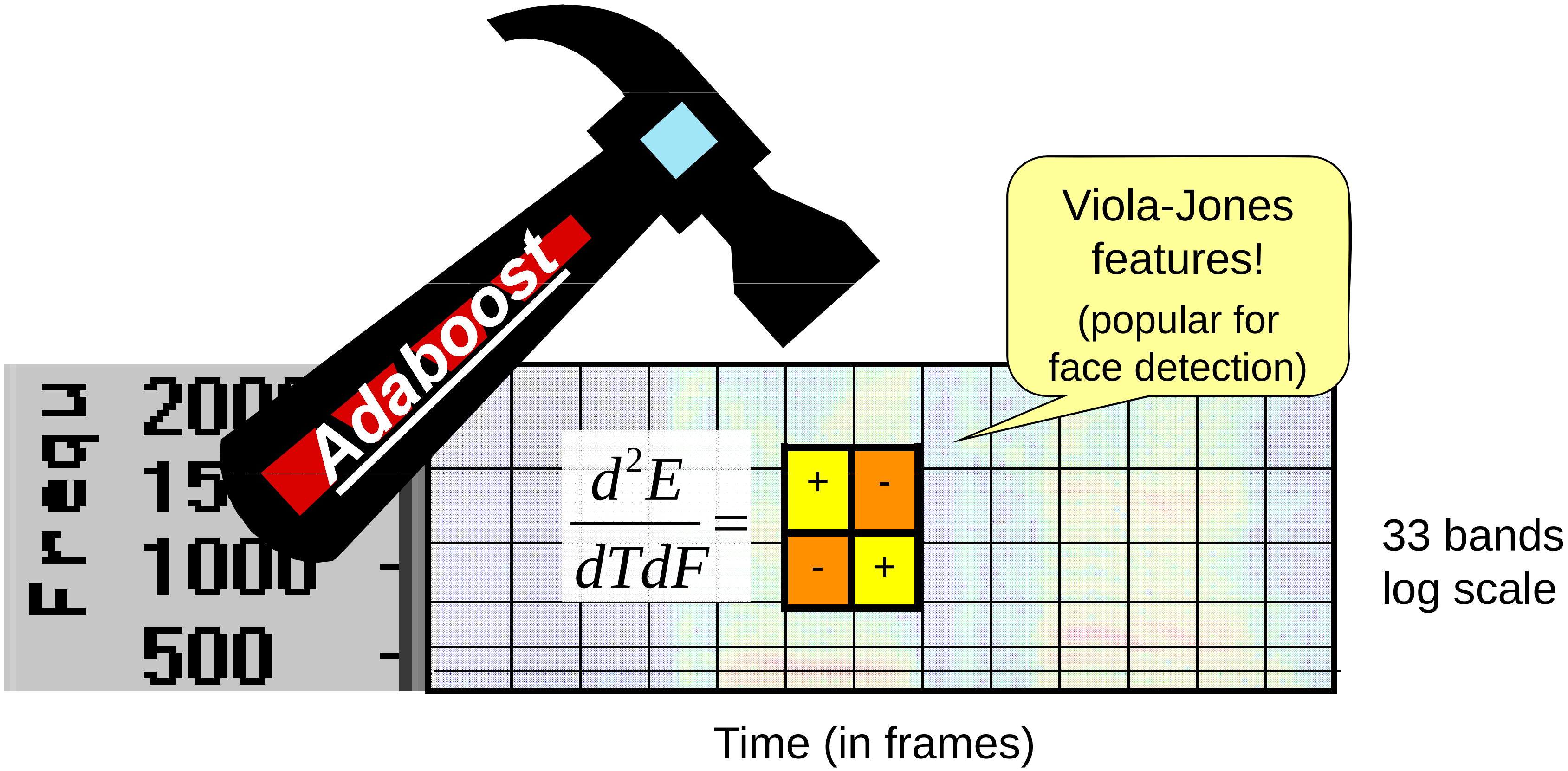
- At every frame & frequency band, compute: $\dfrac{d^2E}{dTdF}$
- Threshold at 0 to get a 32-bit descriptor at every time frame



?

[Haitsma & Kalker]’s choice of corner filter was arbitrary
Could we build much better descriptors using machine learning?

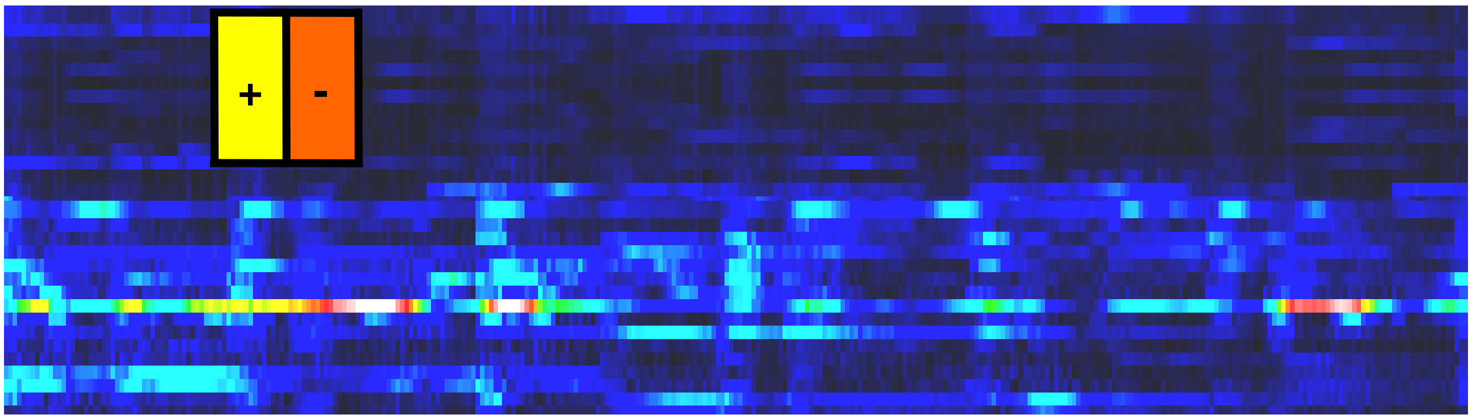
Boosting a Better Descriptor



A descriptor is composed from the outputs of the chosen set of binary filters.
Our goal is to pick a good set of filters

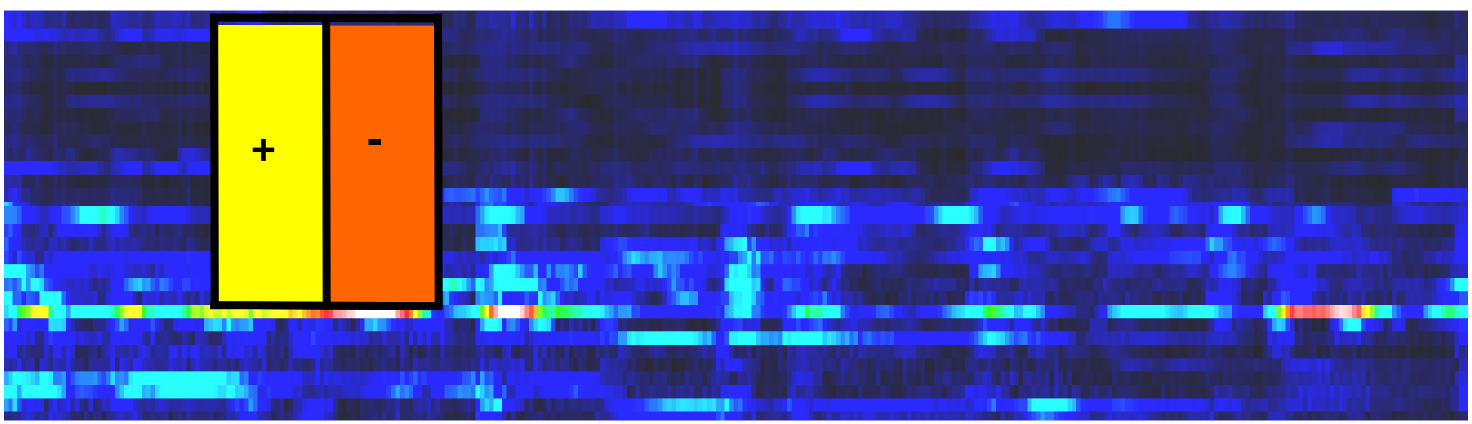
What is a Filter?

- Generates one bit from box sums/differences
- Intuition: filters should generate the same output for similar snippets
- Parameters: filter type, corner locations (in time & freq.), threshold
- If (sum >= threshold) then filter output = 1, else filter output = 0
- One filter is weak indicator, so we use several independent ones
- How to select good filters from a pool of 30,000? Boosting



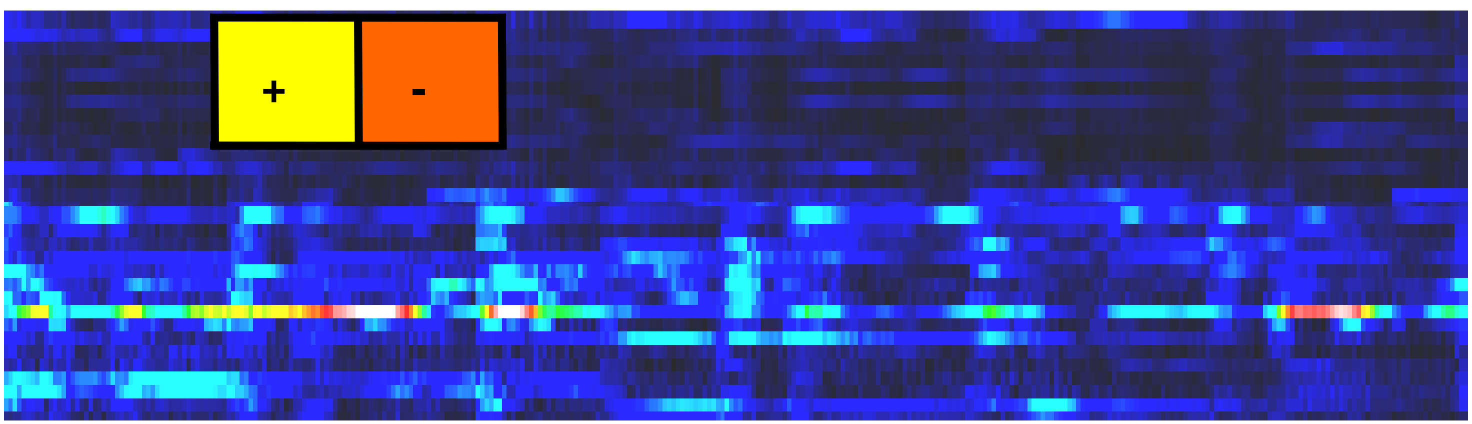
What is a Filter?

- Generates one bit from box sums/differences
- Intuition: filters should generate the same output for similar snippets
- Parameters: filter type, corner locations (in time & freq.), threshold
- If (sum >= threshold) then filter output = 1, else filter output = 0
- One filter is weak indicator, so we use several independent ones
- How to select good filters from a pool of 30,000? Boosting

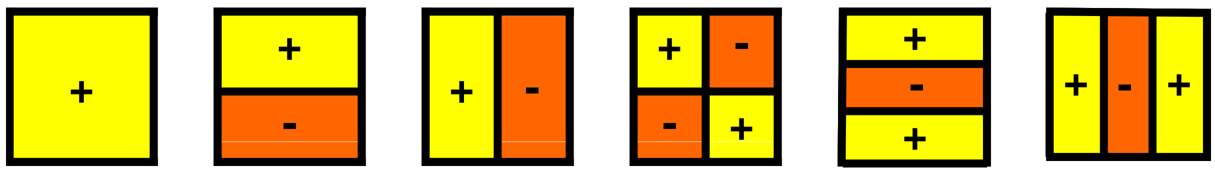


What is a Filter?

- Generates one bit from box sums/differences
- Intuition: filters should generate the same output for similar snippets
- Parameters: filter type, corner locations (in time & freq.), threshold
- If (sum >= threshold) then filter output = 1, else filter output = 0
- One filter is weak indicator, so we use several independent ones
- How to select good filters from a pool of 30,000? Boosting



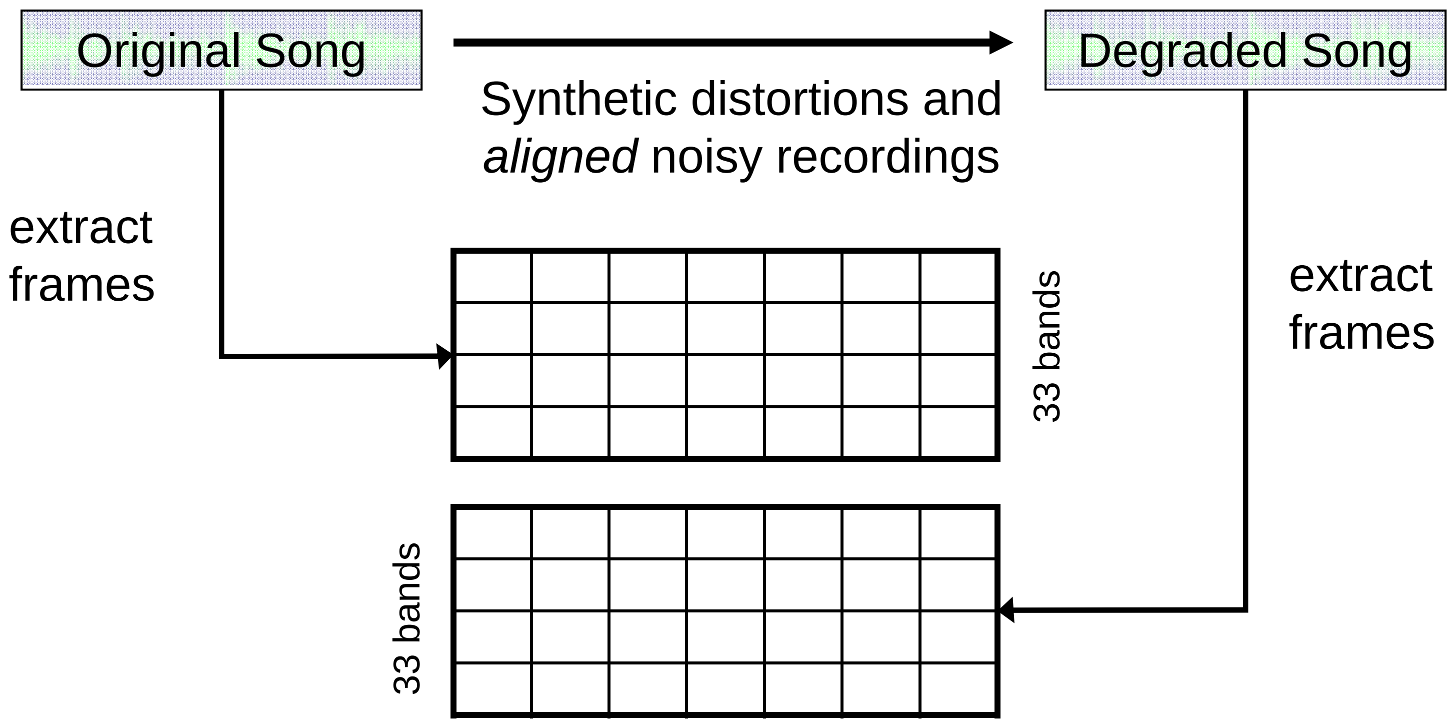
Candidate Filters

- Learning parameters: time width, band width, start band, filter type, threshold.
- Times: 1, 2, 4, 8,… frames, up to 1 second
- Filter types: 

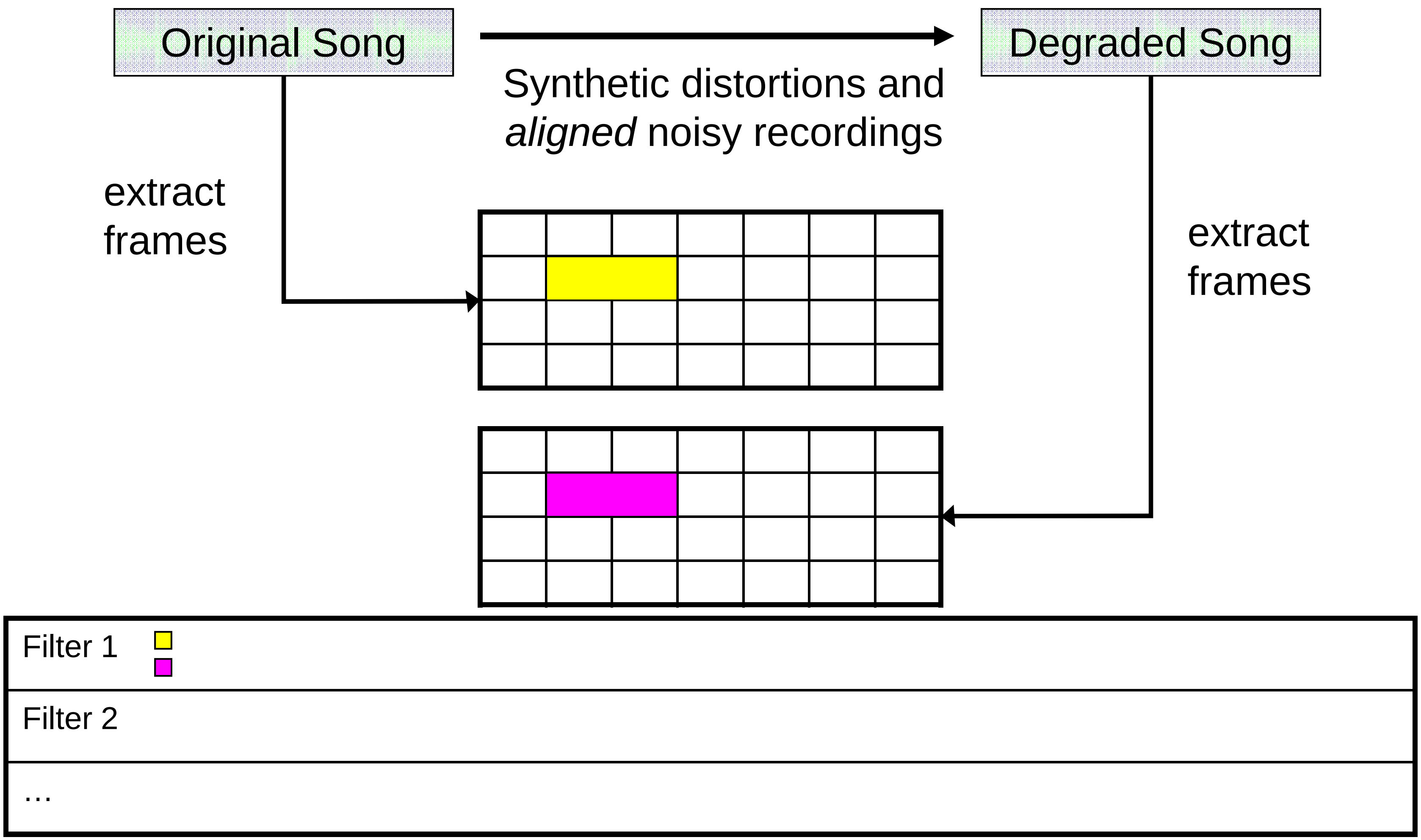
- ~ 30,000 filters total to choose from

Goal: select best 32-element subset of filters

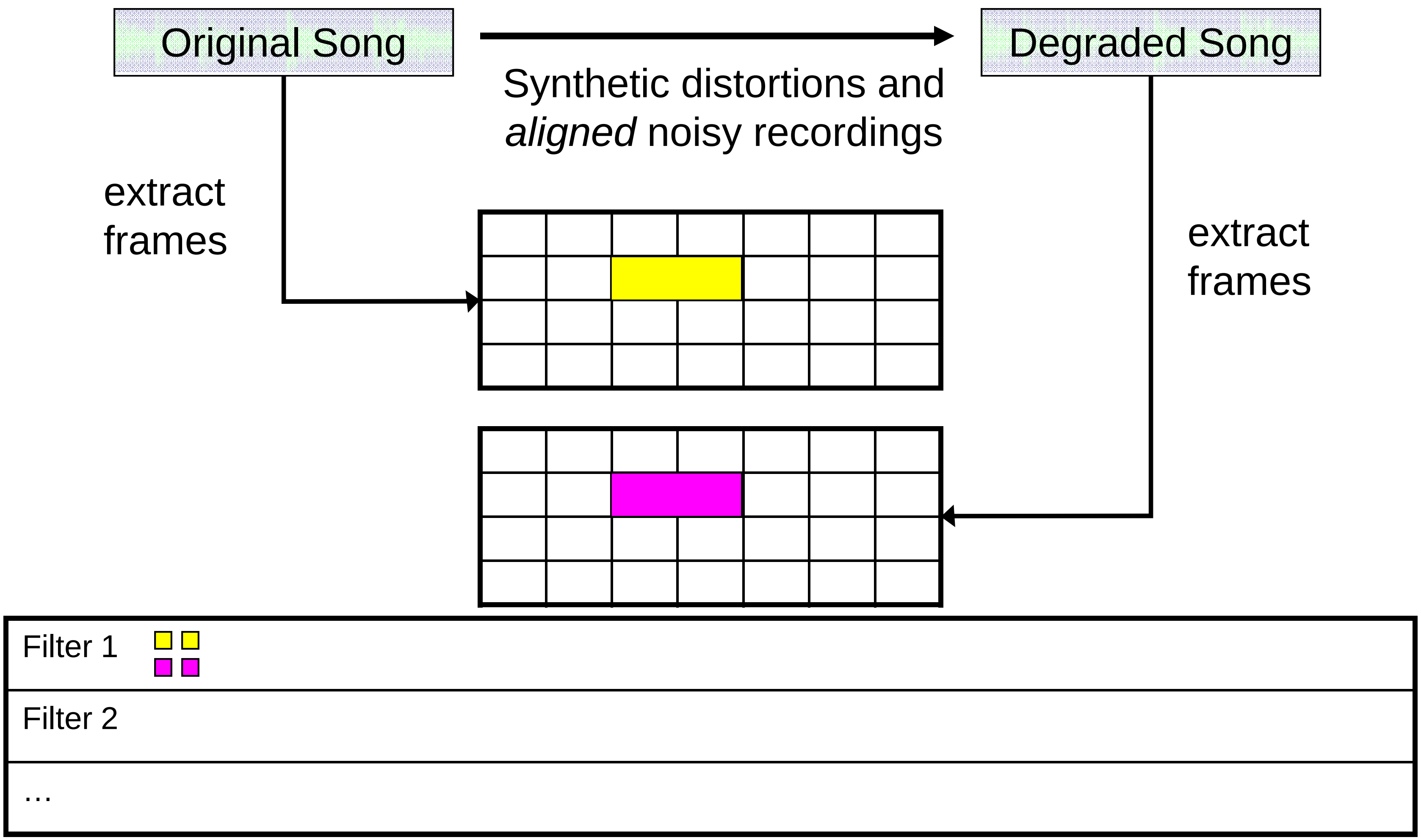
Generating Training Data



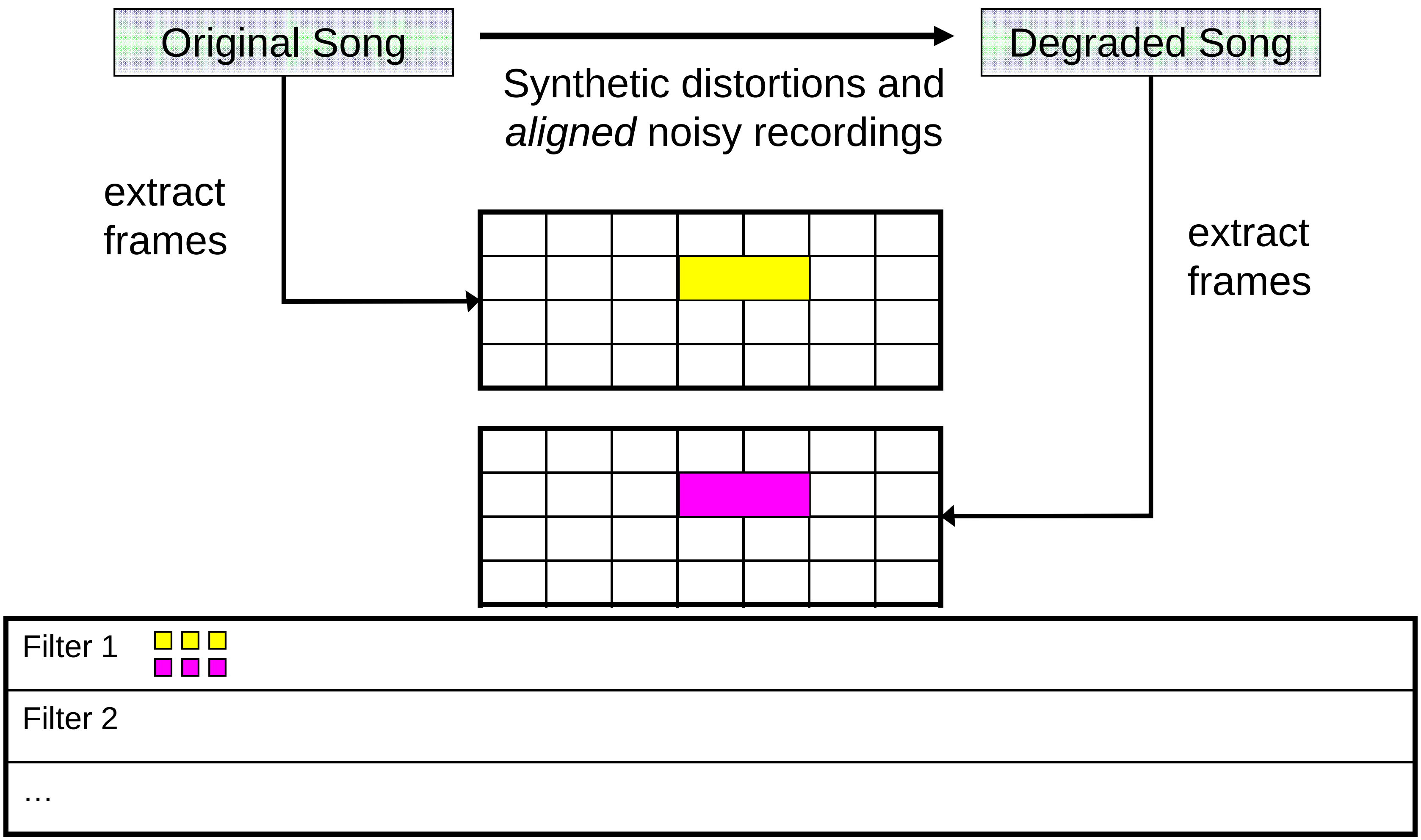
Generating Training Data



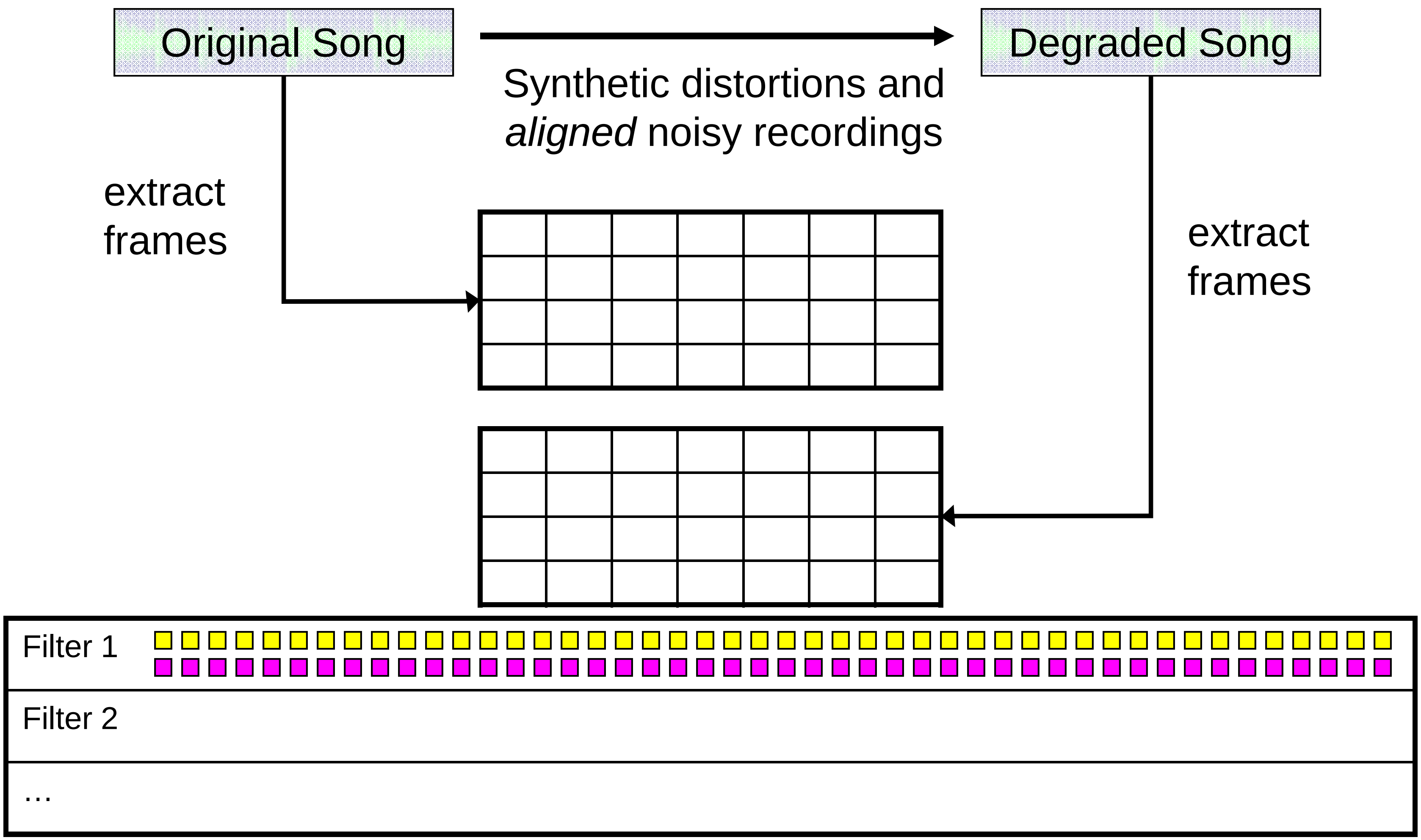
Generating Training Data



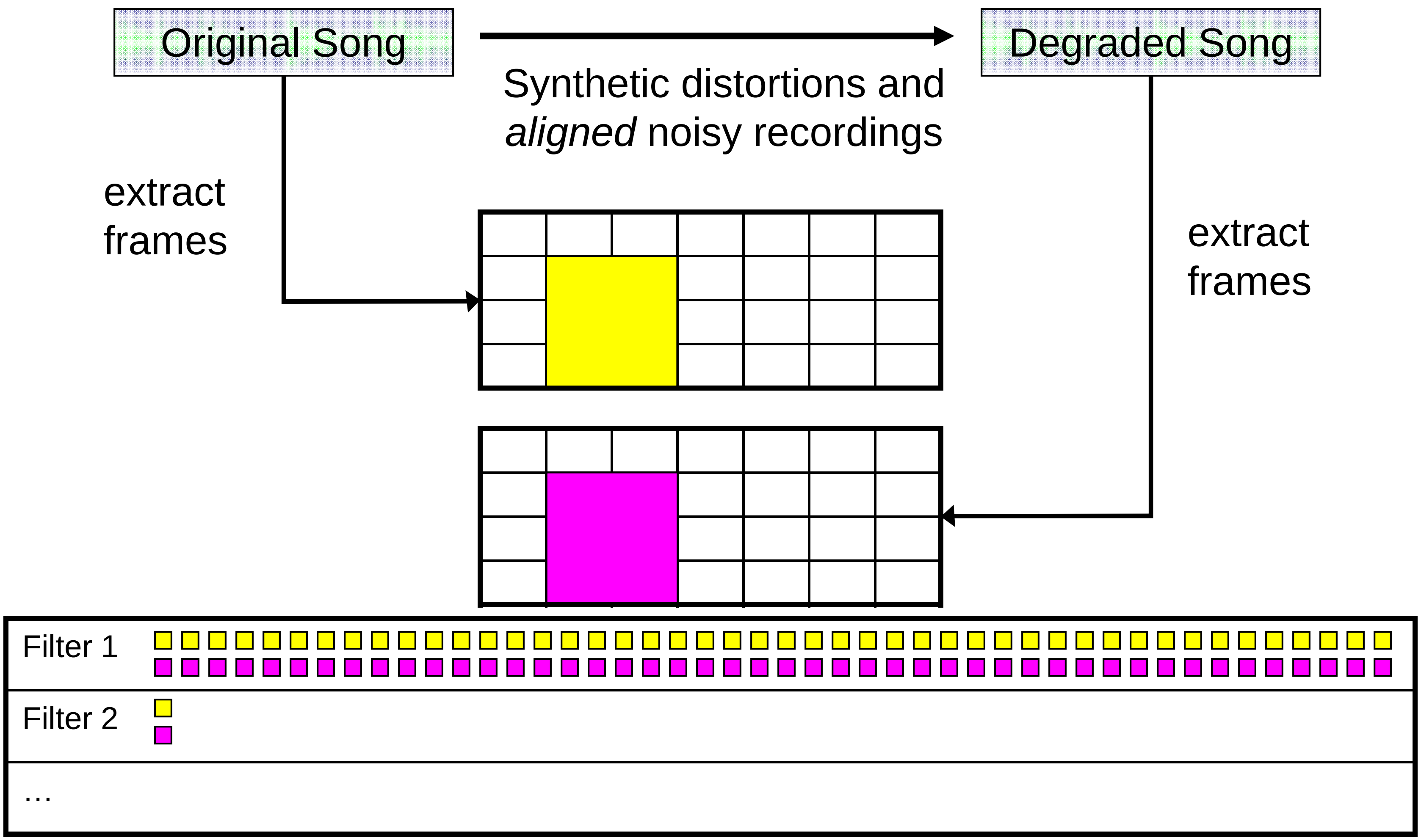
Generating Training Data



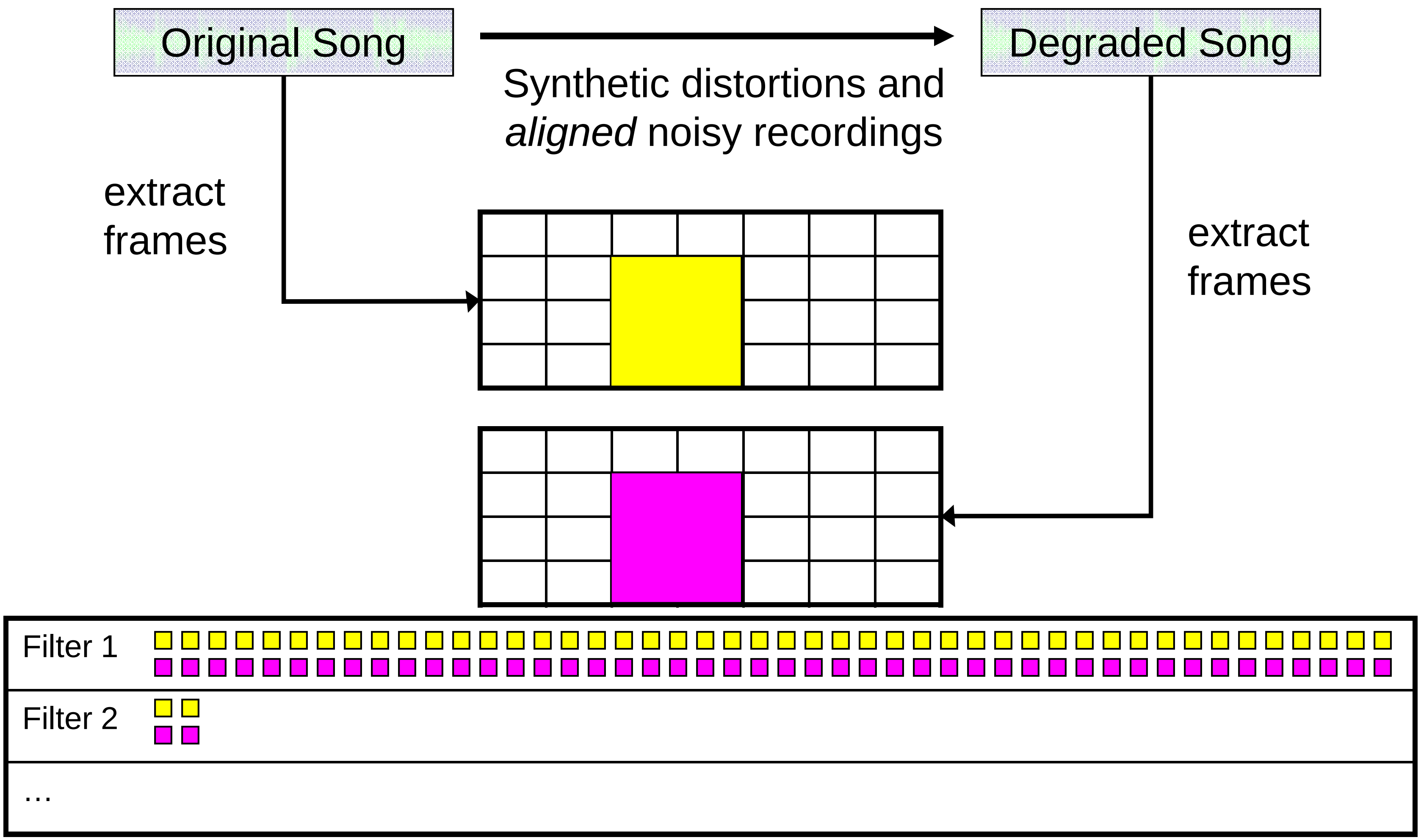
Generating Training Data



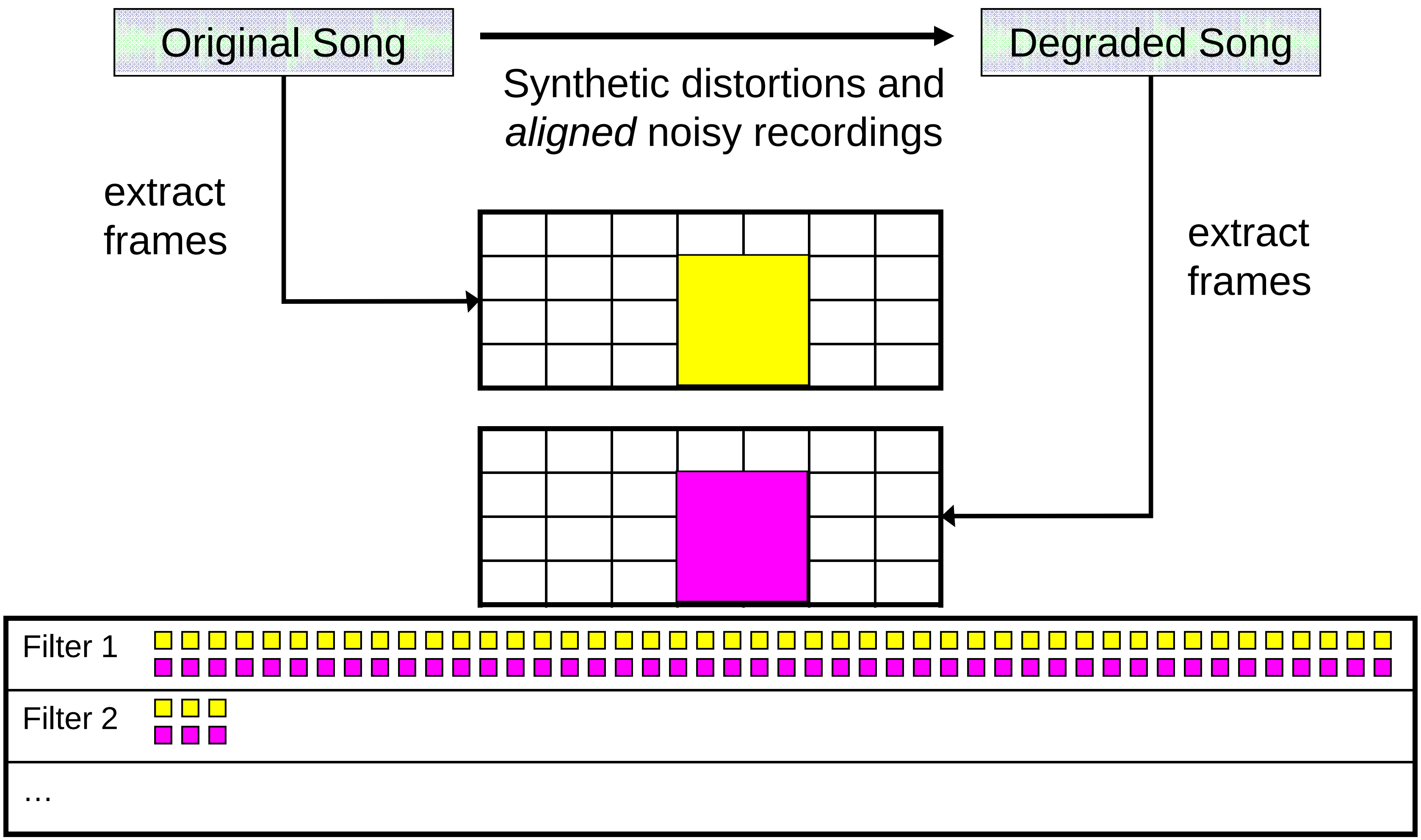
Generating Training Data



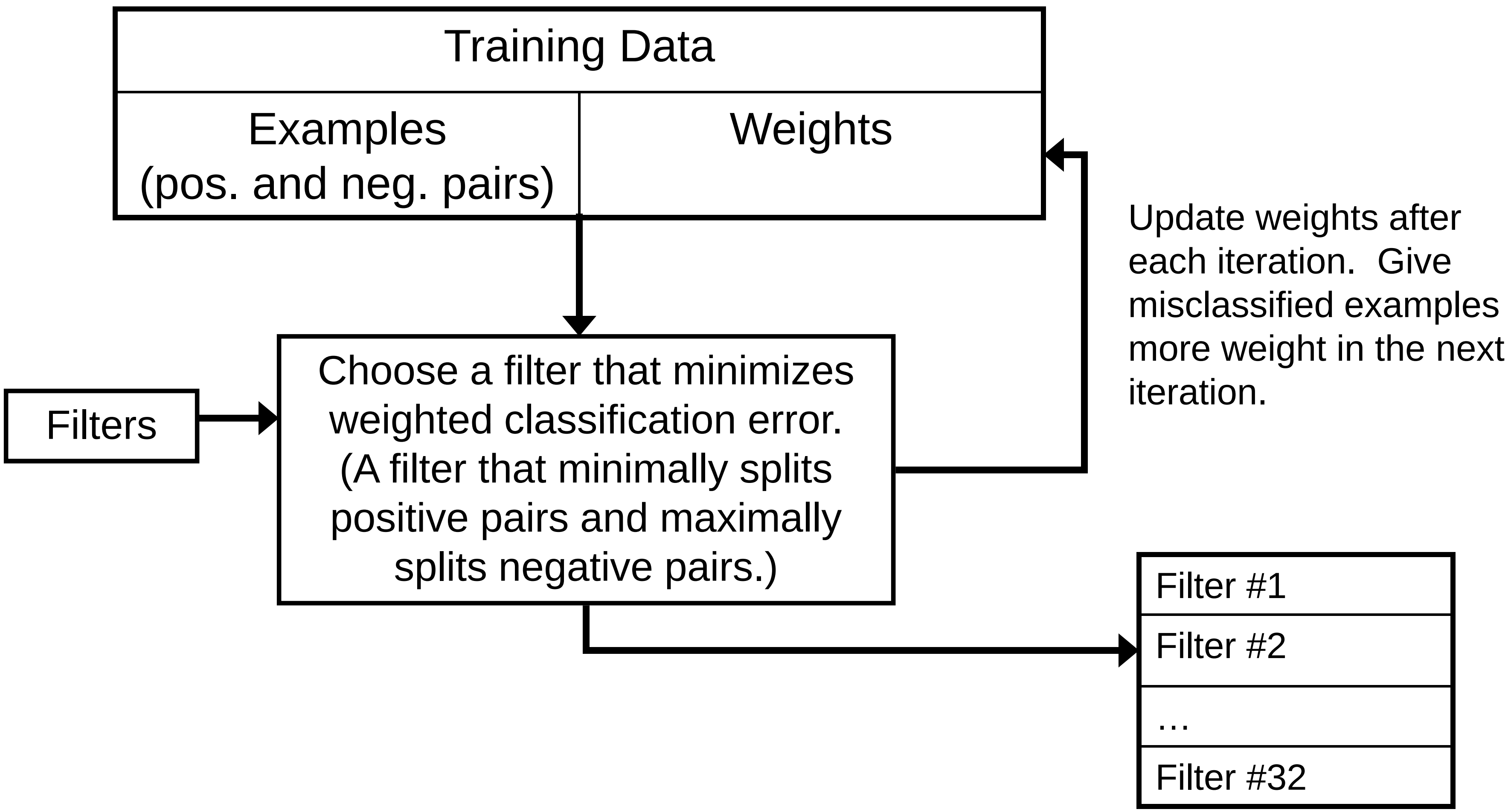
Generating Training Data



Generating Training Data



Choosing Filters with Adaboost



Why Boosting?

- Benefits:
  - Chooses a set of filters that works well together
  - Successive filters minimize bound on error
  - Selected filters tend to be independent
- What's new (our contribution):
  - Trained on *pairs* of positive & negative exemplars.
  - Filter output used as *descriptor*, not as a classifier

Pairwise Boosting

Pairwise Boosting

input: sequence of n examples $\langle(x_{11}, x_{21})\rangle..\langle(x_{1n}, x_{2n})\rangle$, each with label $y_i \in \{-1, 1\}$

initialize: $w_i = \frac{1}{n}, i = 1..n$

for m = 1..M

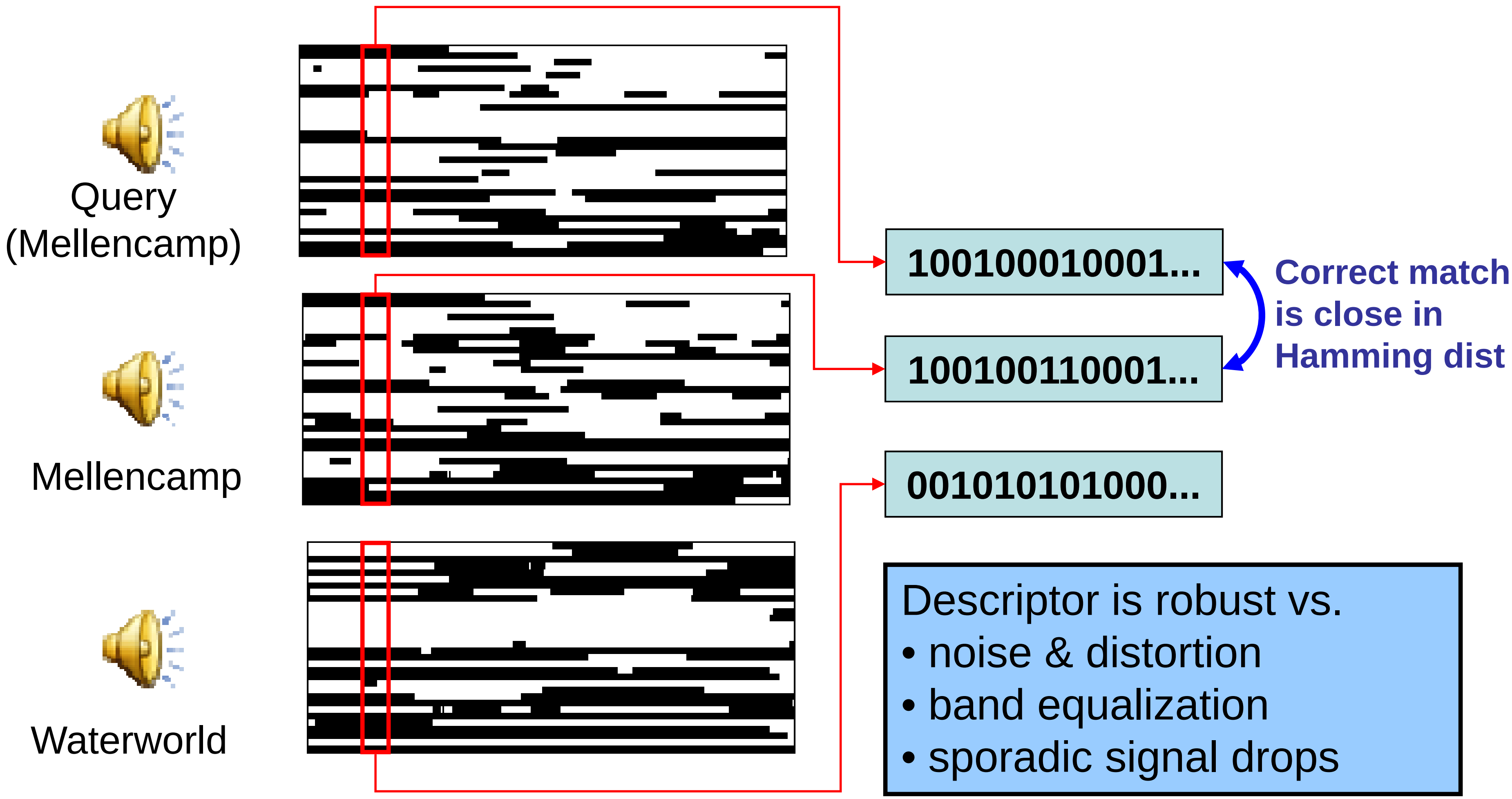
1. find the hypothesis $h_m(x_1, x_2)$ that minimizes weighted error over distribution w, where $h_m(x_1, x_2) = sgn[(f_m(x_1) - t_m)(f_m(x_2) - t_m)]$ for filter f_m and threshold t_m
2. calculate weighted error: $err_m = \sum_{i=1}^{n} w_i \cdot \delta(h_m(x_{1i}, x_{2i}) \neq y_i)$
3. assign confidence to h_m: $c_m = \log(\frac{1-err_m}{err_m})$
4. update weights for matching pairs: if $y_i = 1$ and $h_m(x_{1i}, x_{2i}) \neq y_i$, then $w_i \leftarrow w_i \cdot exp[c_m]$
5. normalize weights such that $\sum_{i:y_i=-1}^{n} w_i = \sum_{i:y_i=1}^{n} w_i = \frac{1}{2}$.

final hypothesis:
$H(x_1, x_2) = sgn(\sum_{m=1}^{M} c_m h_m(x_1, x_2))$

Observations

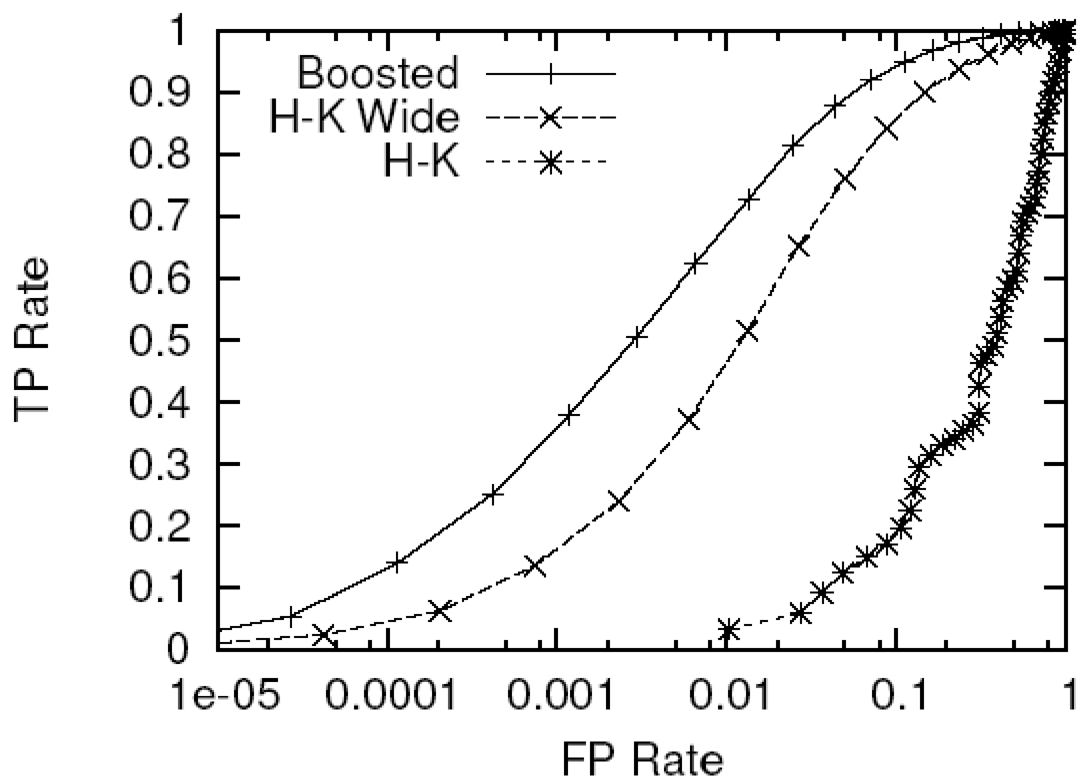
- Standard Adaboost doesn't work on this multi-class problem
- Two snippets match if they fall on same side of the threshold
- Asymmetry: No weak classifier can do better than chance on *non-matching* pairs – can only learn from the *matching* pairs
- Median response is optimal threshold for non-matching pairs – greatly reduces training time

Name That Tune: Our Descriptors



Descriptor is robust vs.
- noise & distortion
- band equalization
- sporadic signal drops

Descriptor-level Matching Results



H-K = [Haitsma & Kalker 2002]
H-K Wide = our improvements on H-K
Boosted = our pairwise boosted features (32-bits)

Descriptors vs. Distance Metrics

- Alternate view: pose the descriptor learning problem as supervised distance metric learning
- Given pairs of similar/dissimilar snippets, can we directly *learn* a good Hamming space where similar songs are near while dissimilar songs are far?

MusicID Algorithm

- Transform audio into spectrogram (2D image)
- Compute distinctive local descriptors (learned by pairwise boosting)
- Retrieve candidates using efficient index (near-neighbor in high-dim)
- Identify song using robust alignment (RANSAC + noise model)

- Near-neighbor for similar descriptors in high-dimensions is painful
- Sub-image retrieval [MM2004] used locality-sensitive hashing
- MusicID employs direct hashing with extra probes
  - Threshold = 0 needs 1 hash probe
  - Threshold = 1 needs 1 + 32 hash probes
  - Threshold = 2 needs 1 + 32 + 32*31/2 = 529 probes
  - Threshold = 3 needs 1+32+32*31/2+32*31*30/6 = 5489 probes

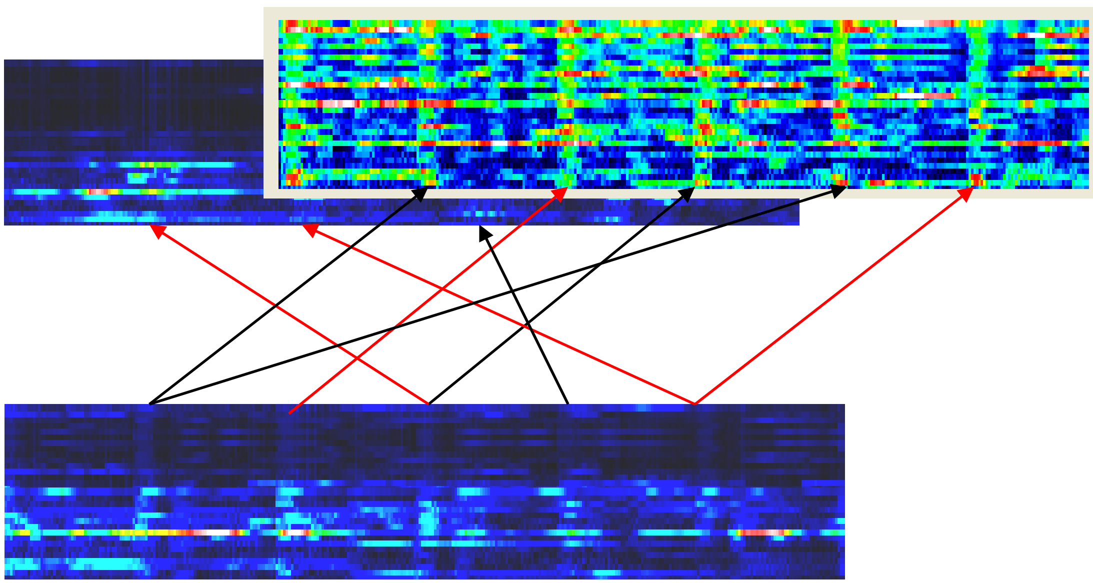
Direct Hashing: Recall vs. Computation Tradeoff

	Distance Threshold			
	0	1	2	3
Boosted	1.1%	5.4%	14.0%	25.2%
H-K Wide	$< 0.01\%$	0.09%	0.64%	2.5%
H-K	$< 0.01\%$			

- Recall for a snippet with given Hamming threshold
- Threshold = 0 needs 1 hash probe
- Threshold = 1 needs 1 + 32 hash probes
- Threshold = 2 needs 1 + 32 + 32*31/2 = 529 probes
- Threshold = 3 needs 1+32+32*31/2+32*31*30/6 = 5489

MusicID Algorithm

- Transform audio into spectrogram (2D image)
- Compute distinctive local descriptors (learned by pairwise boosting)
- Retrieve candidates using efficient index (near-neighbor in high-dim)
- Identify song using robust alignment (RANSAC + noise model)



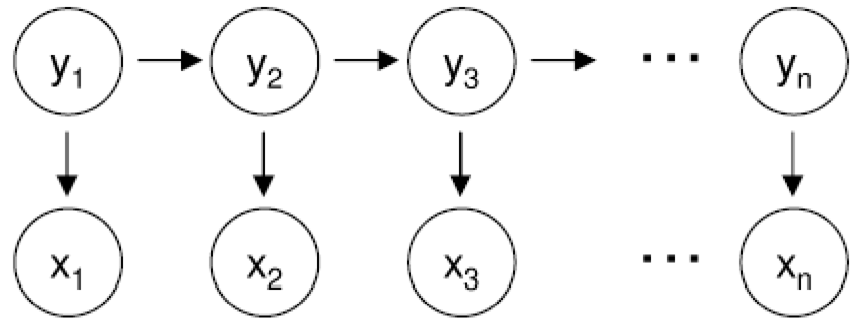
RANSAC
-Sample minimal set
-Generate transform
-Snippet matches "vote"
-Best song wins

Incorporate HMM "occlusion" model

Simple "Occlusion" Model

Music vs. Noise Label

Descriptor Distance



Bit difference on descriptors for one snippet

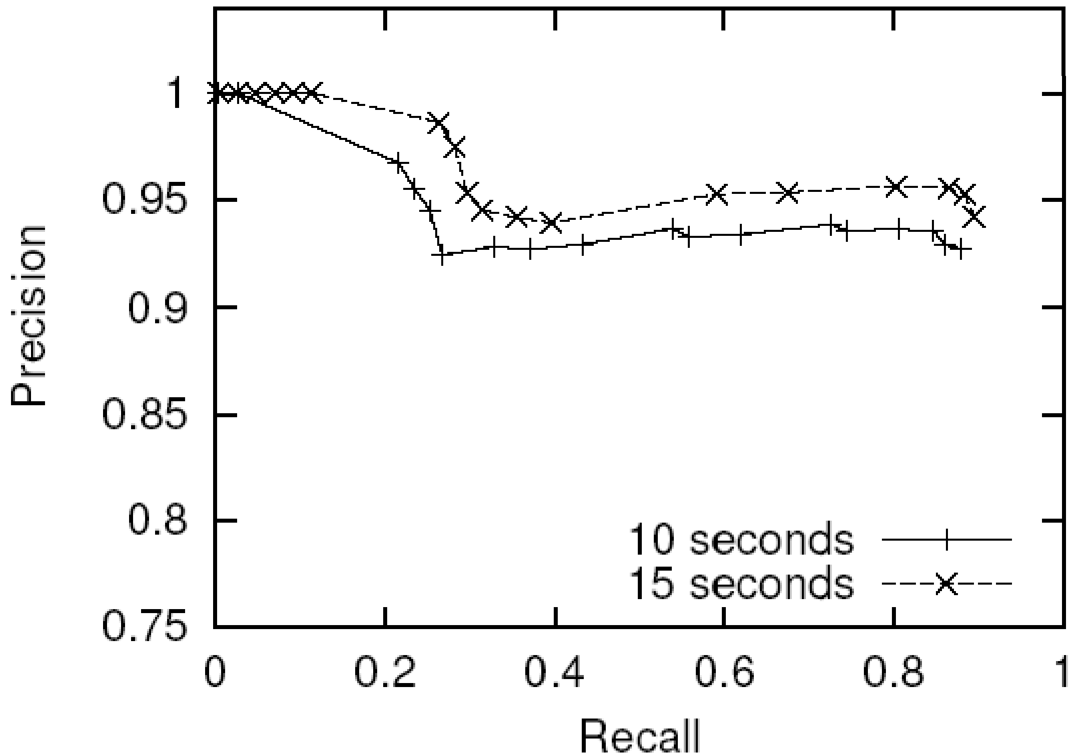
Transition probability

$$P(x^r|x^o) = P(x^{r-o}) = \prod_{i=1}^{n} P(x_i^{r-o}|y_i)P(y_i|y_{i-1})$$

66 parameters, trained easily using EM

Independent, non-identically distributed Bernoulli random variables

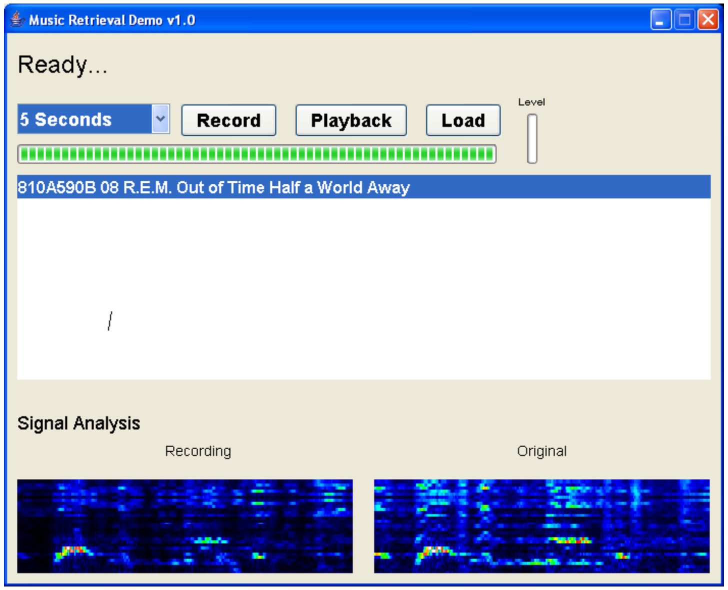
Music Identification Results



Test set: ~300 clips played at low volume with significant background noise
Drawn from database with 1862 songs (classical, vocal, rock, pop).
Random guess accuracy is 1/1862 = 0.05%

MusicID Summary

- This system accurately and efficiently identifies music from a 5-10 second sample taken in noisy conditions
- Our pairwise boosted descriptors outperform traditional ones
- Geometric verification adds robustness to “occlusions”



Download demo, video, CVPR paper, source code from http://www.cs.cmu.edu/~rahuls/

Application of Music Identification: Google's Ambient Audio Identification

- Applies and extends audio fingerprinting from MusicID to detect current TV channel based on ambient audio in living room

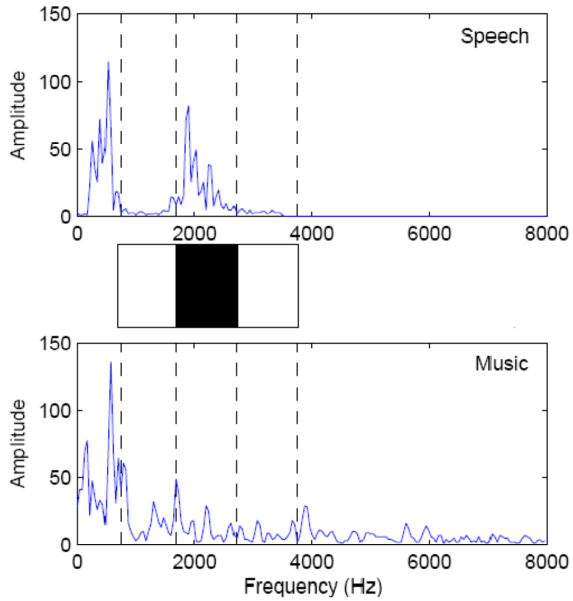
- M. Fink, M. Covell, S. Baluja, "Social and Interactive TV using Ambient-Audio Identification", EuroITV 2006.

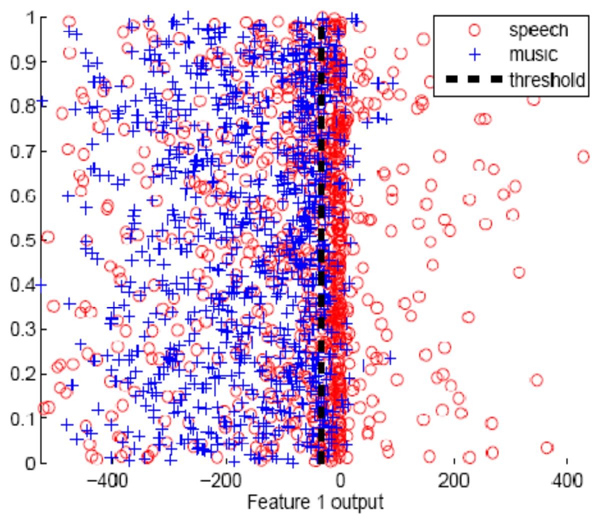
Conclusion

- Machine learning approaches developed for vision often translate nicely to audio tasks (and vice versa).
- Interesting relationships between learning feature descriptors and distance metrics
- Download papers, code and video from:
  http://www.cs.cmu.edu/~rahuls/

Related work:
Music vs. Speech Classification

- Problem: classify clip as either “music” or “speech”
- Analogy: VJ binary classifier using Haar-like features

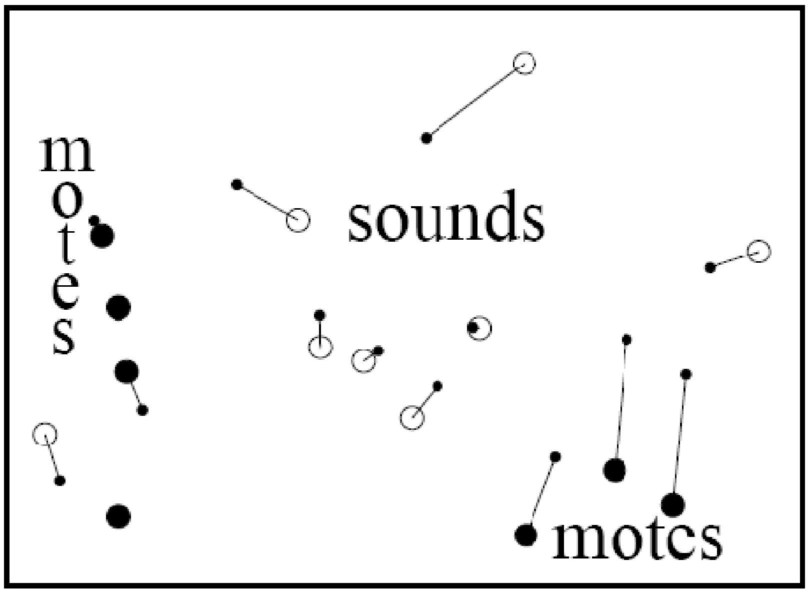




- N. Casagrande *et al.*, “Frame-level speech/music discrimination using AdaBoost”, ISMIR 2005

Related work: Structure from Sound

- Problem: localize microphones from sound events
- Analogy: structure from motion with affine camera model



- S. Thrun, “Affine structure from sound”, NIPS 2005