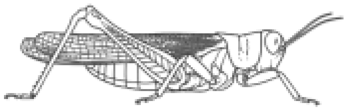
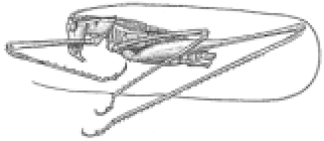
Monitoring and Mining Animal Sounds in Visual Space

Yuan Hao

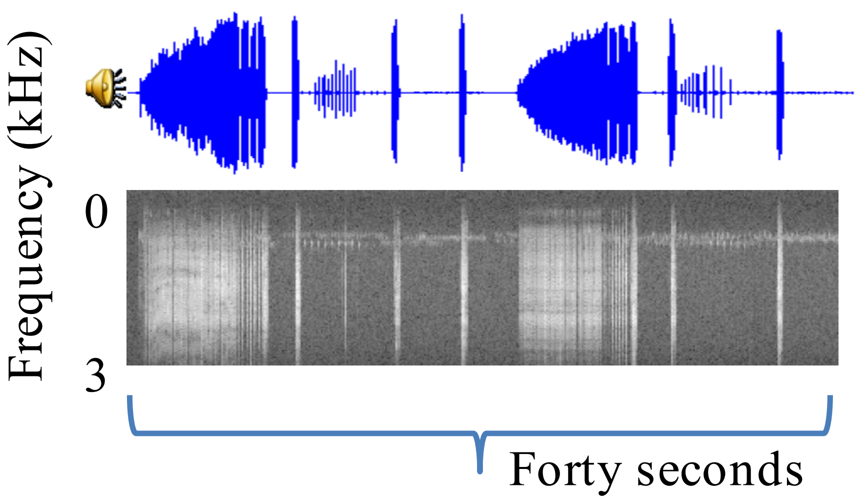
Dept. of Computer Science & Engineering
University of California, Riverside

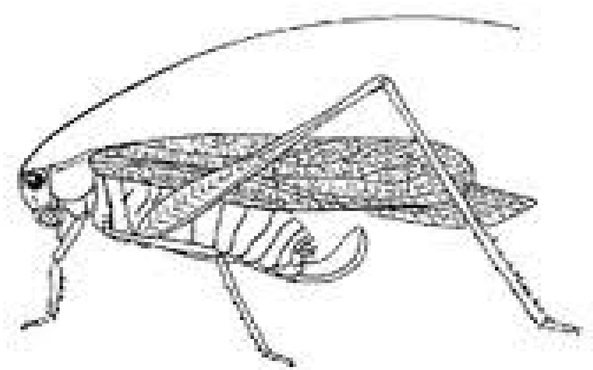




Task

- Monitoring animals by examining the sounds they produce
- Build animal sound recognition/classification framework





Common Virtuoso Katydid
(*Amblycorypha longinicta*)

Outline

- Motivation
- Our approach
- Experimental evaluation
- Conclusion & future work

Motivation-application

<u>Monitoring animals</u>:

Outdoors

- The density and variety of animal sounds can act as a measure of biodiversity

Laboratory setting

- Researchers create control groups of animals, expose them to different settings, and test for different outcomes

<u>Commercial application:</u>

Acoustic animal detection can save money

Motivation-difficulties

Most current bioacoustic classification tools have significant limitations

They…

- require careful tuning of many parameters
- are too computationally expensive for sensors
- are not accurate enough
- too specialized

Related Work

- **Dietrich et al (MCS 01), several classifications methods for insect sounds**
  - Preprocessing and complicated feature extraction
  - Up to *eighteen* parameters
  - Learned on a data set containing just *108* exemplars

- **Brown et al (J. Acoust. Soc 09), analyze Australian anurans (frogs and toads)**
  - Identify the species of the frogs with an average accuracy of 98%
  - Requires extracting features from syllables
  - "*Once the syllables have been properly segmented, a set of features can be calculated to represent each syllable*"

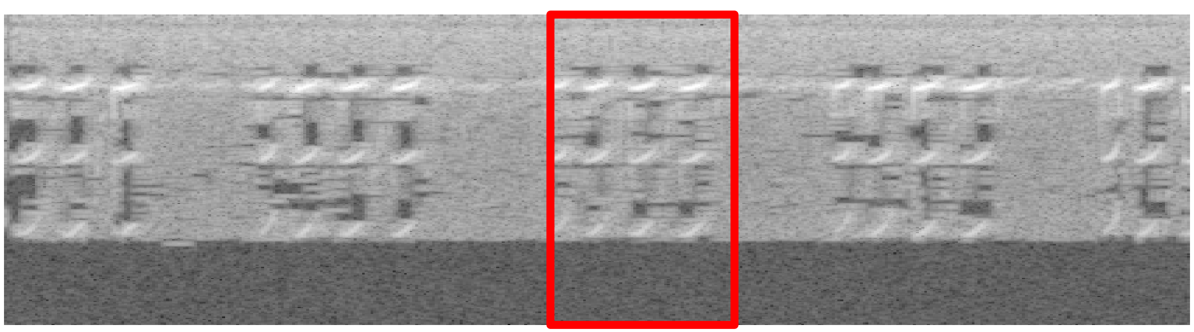
Outline

- Motivation
- Our approach
  - Visual space-spectrogram
  - CK distance measure
  - Sound fingerprint searching
- Experimental evaluation
- Conclusion & future work

Intuition of our Approach

- Classify the animal sounds in the *visual space*, by treating the *texture* of their spectrograms as an "acoustic fingerprint", using a recently introduced parameter-free texture measure as a distance measure

Can be considered the
"*fingerprint*" for this sound

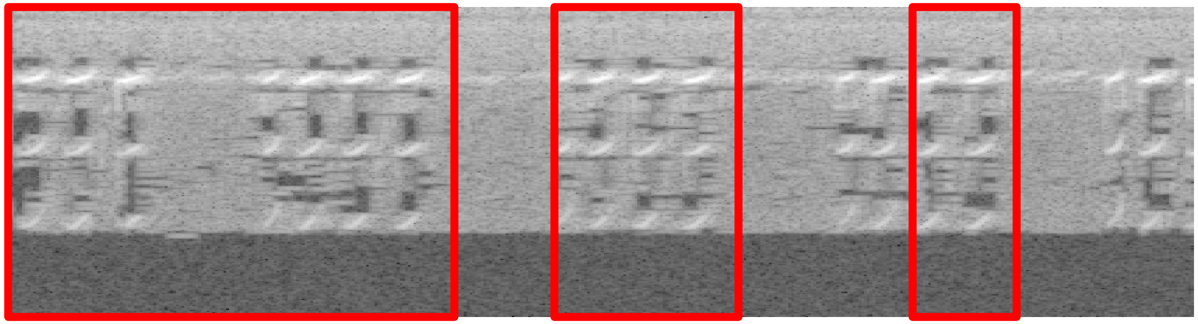


One second subset of a common cricket' sound spectrogram

Intuition of our Approach

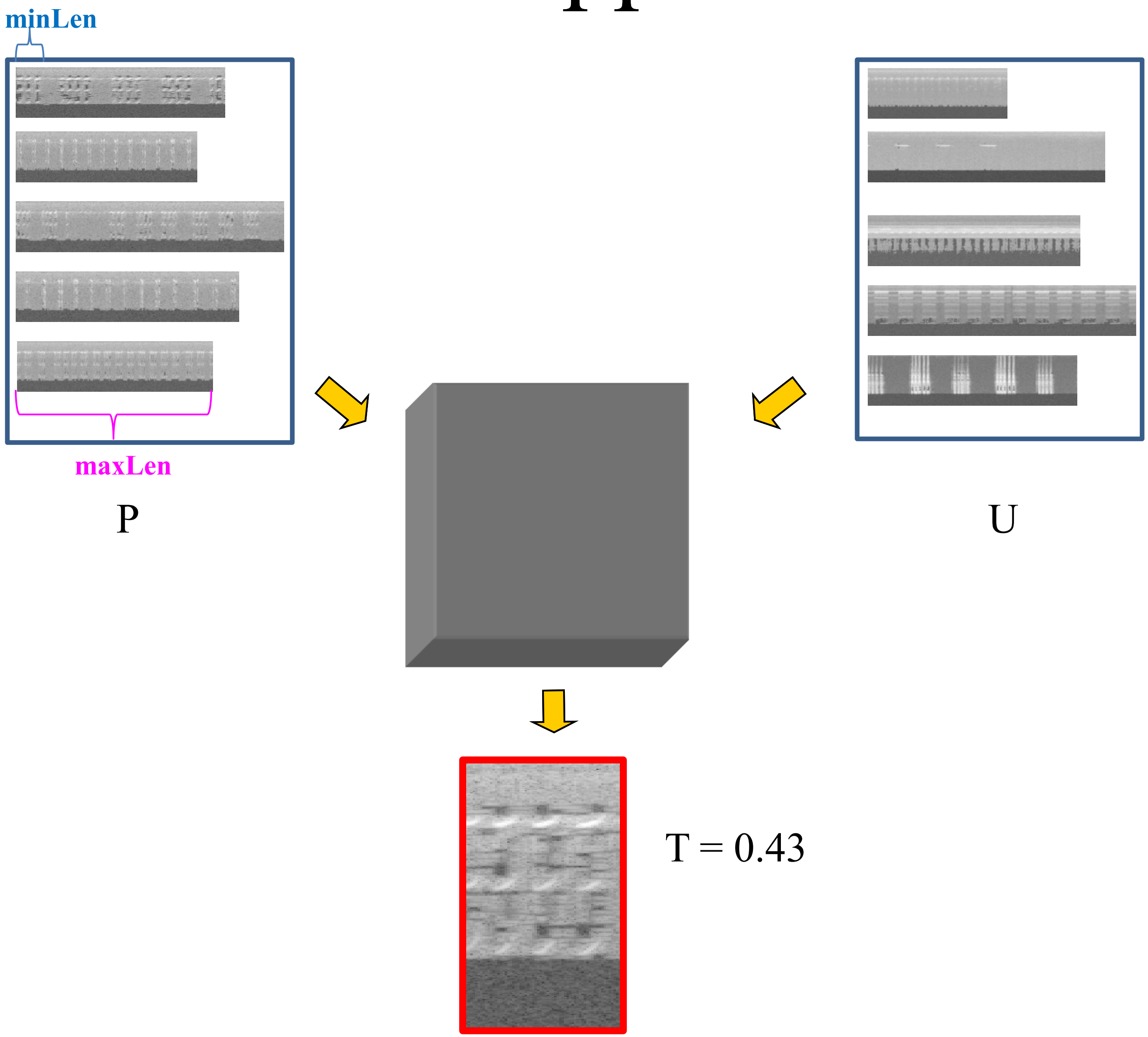
- Classify the animal sounds in the *visual space*, by treating the *texture* of their spectrograms as an "acoustic fingerprint", using a recently introduced parameter-free texture measure as a distance measure

Can be considered the
"*fingerprint*" for this sound



One second subset of a common cricket' sound spectrogram

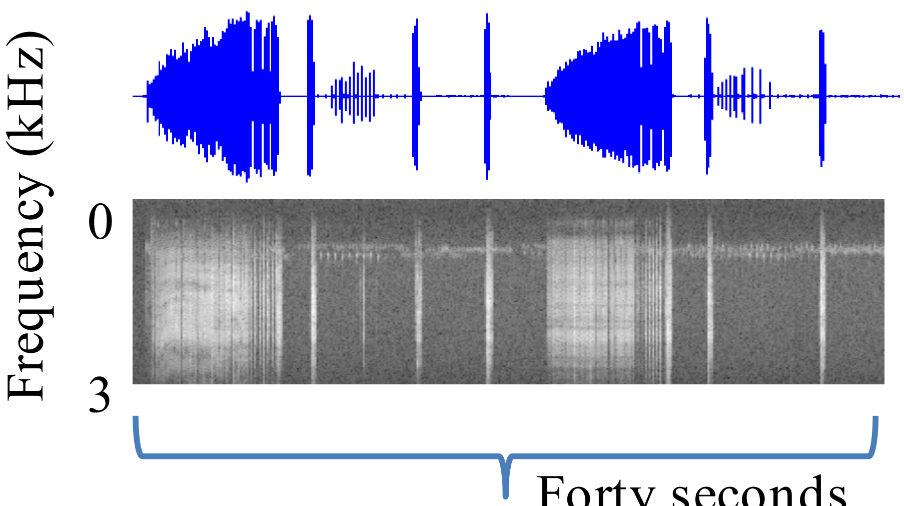
Our Approach

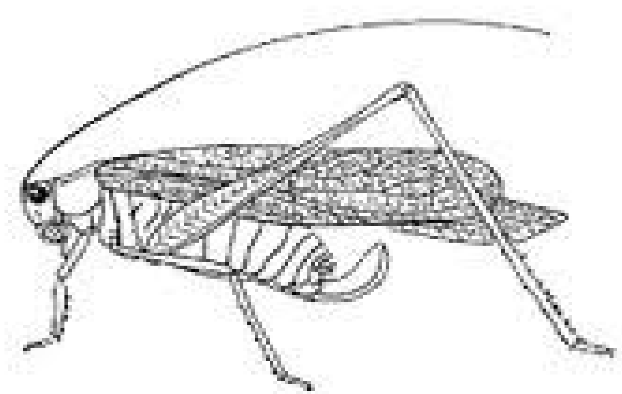


Visual Space

Spectrogram

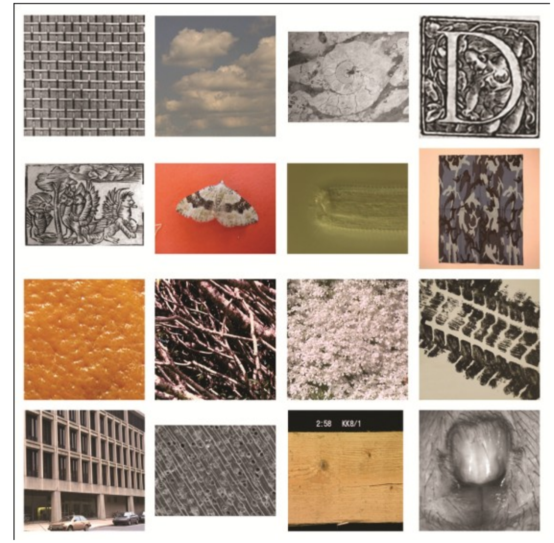
- Algorithmic analysis needed instead of manual inspection
- Significant noise artifacts
- Avoid any type of data cleaning or explicit feature extraction, and use the raw spectrogram





Common Virtuoso Katydid
(*Amblycorypha longinicta*)

CK Distance Measure

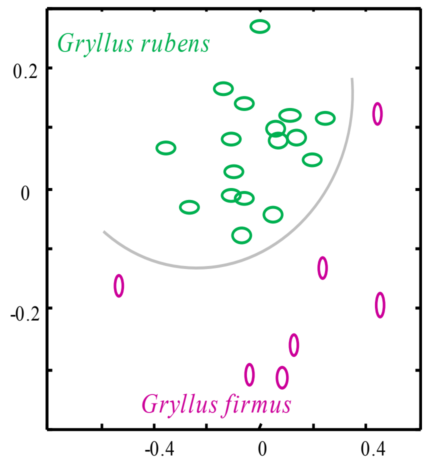


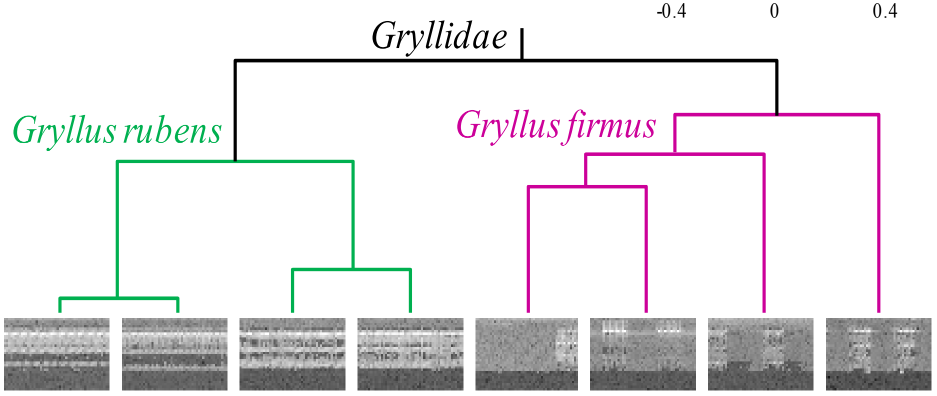
$$d_{CK}(x,y) = \frac{C(x|y)+C(y|x)}{C(x|x)+C(y|y)} - 1$$

- Distance measure of texture similarity
- Robustly extracting features from noisy field recordings is non-trivial
- Expands the scope of the compression-based similarity measurements to real-valued images by exploiting the compression technique used by MPEG video encoding.
- Effective on images as diverse as moths, nematodes, wood grains, tire tracks etc (SDM 10)

Sanity Check

CK as a tool for taxonomy





National Geographic article

"the sand field cricket (Gryllus firmus) and the southeastern field cricket (Gryllus rubens) look nearly identical and inhabit the same geographical areas"

Outline

- Motivation
- Our approach
  - Visual space-spectrogram
  - CK distance measure
  - Sound fingerprint searching
- Experimental evaluation
- Conclusion & future work

Difficulties

- Do not have carefully extracted prototypes for each class
  - Only have a collection of sound files
- Do not know the call duration
- Do not know how many occurrences of it appear in each file
- May have mislabeled data
- Noisy: most of the recordings are made in the wild

Example: Discrete Text Strings

Assume three observations that correspond to a particular species

P = {rrbbcxcfbb, rrbbfcxc, rrbbrrbbcxcbcxcf}

Given access to the universe of sounds that are known *not* to contain any example in P

U = {rfcbc, crrbbrcb, rcbbxc, rbcxrf,..,rcc }

Our task is equivalent to asking: *Is there substring that appears only in P and not in U*?

Example: Discrete Text Strings

Assume three observations that correspond to a particular species

P = {rrbbcxcfbb, rrbbfcxc, rrbbrrbbcxcbcxcf}

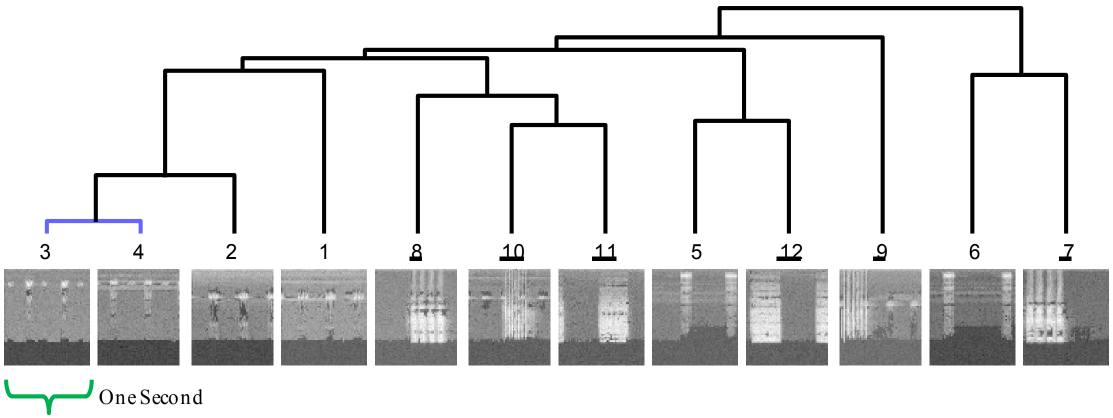
Given access to the universe of sounds that are known *not* to contain any example in P

U = {rfcbc, crrbbrcb, rcbbxc, rbcxrf,..,rcc }

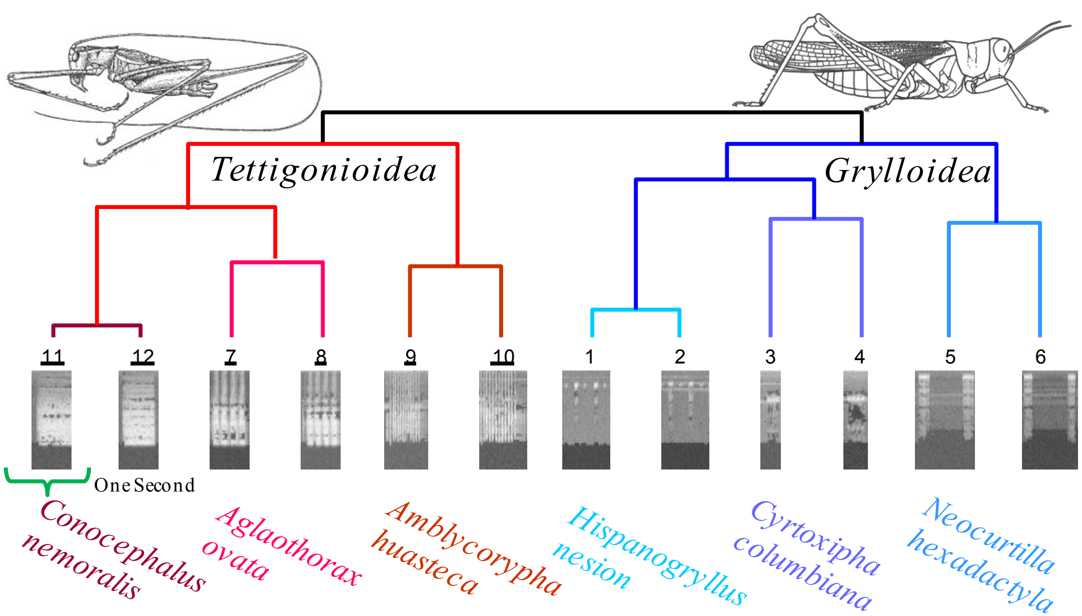
Our task is equivalent to asking: *Is there substring that appears only in P and not in U*?

T_1 = rrbb, T_2 = rrbbc, T_3 = cxc

Case Studies



Six pairs of recordings of various *Orthoptera*. Visually determined and extracted one-second similar regions



One size does not fit all, when it comes to the length of the sound sequence.

Sound Fingerprint

Given *U* and *P*

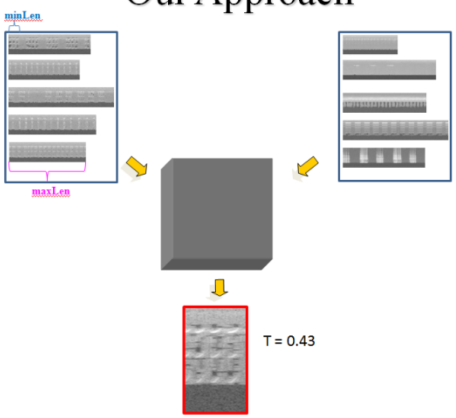
P: Contains examples only from the "positive" species class

U: Non-target species sounds

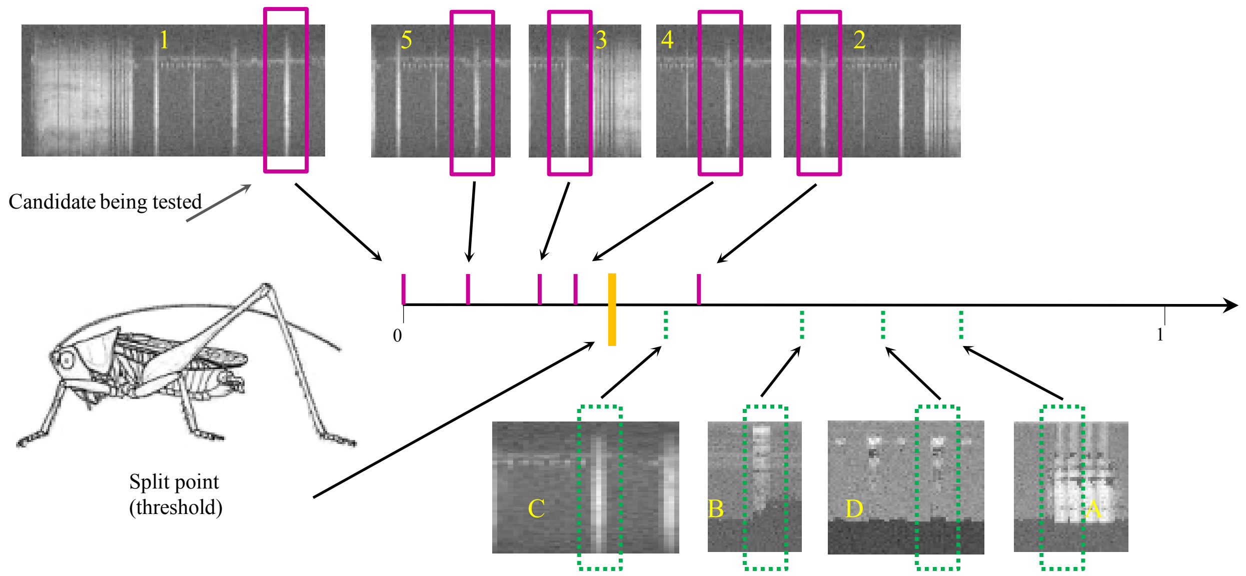
To find a subsequence of one of the objects in *P*, which is close to at *least one* subsequence in each element of *P*, but far from *all* subsequences in every element of *U*

Potential sound fingerprint

Our Approach

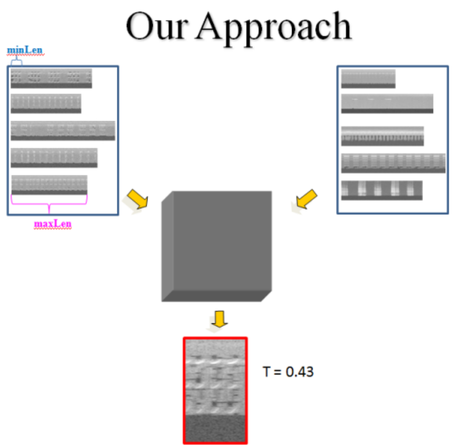


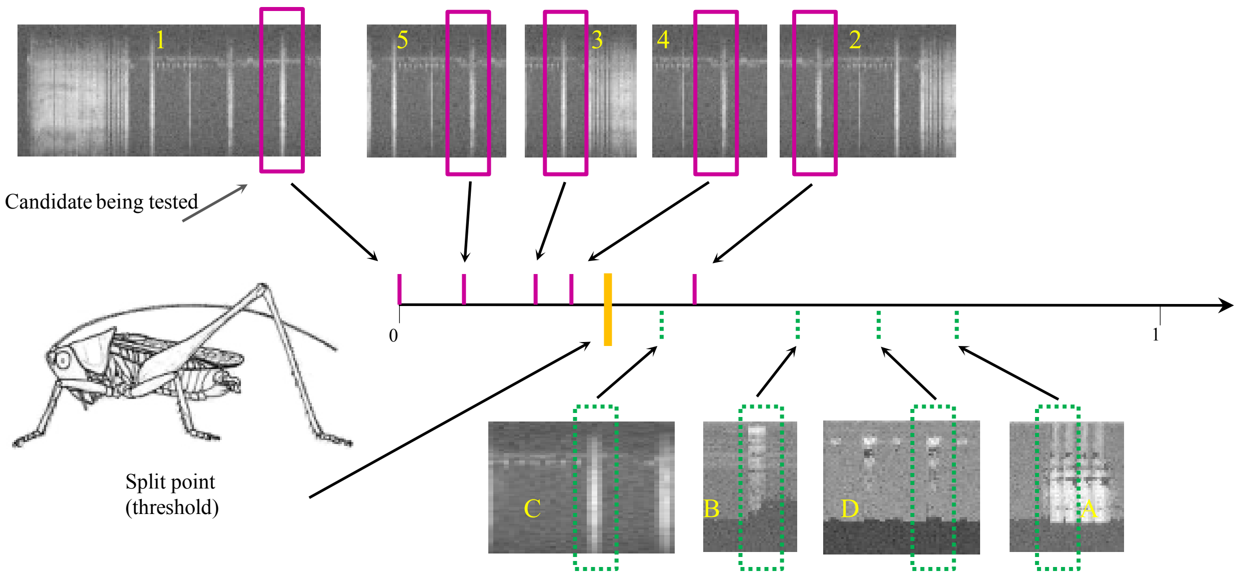
Example



To find a subsequence of one of the objects in *P*, which is close to at *least one* subsequence in each element of *P*, but far from *all* subsequences in every element of *U*

How Hard is This ?





$$\sum_{l=L_{\min}}^{L_{\max}} \sum_{S_i \in \{P\}} (M_i - l + 1)$$

where l is a certain length of candidate

M_i is the length of any sound sequence S_i in P
$L_{\min}$ and $L_{\max}$ is possible user defined length of sound fingerprint

Brute Force Search

Generate and Evaluate

Step 1:

Given **P** and **U**, generate all possible subsequences from the objects in **P** of length *m* as the sound fingerprint candidates.

Step 2:

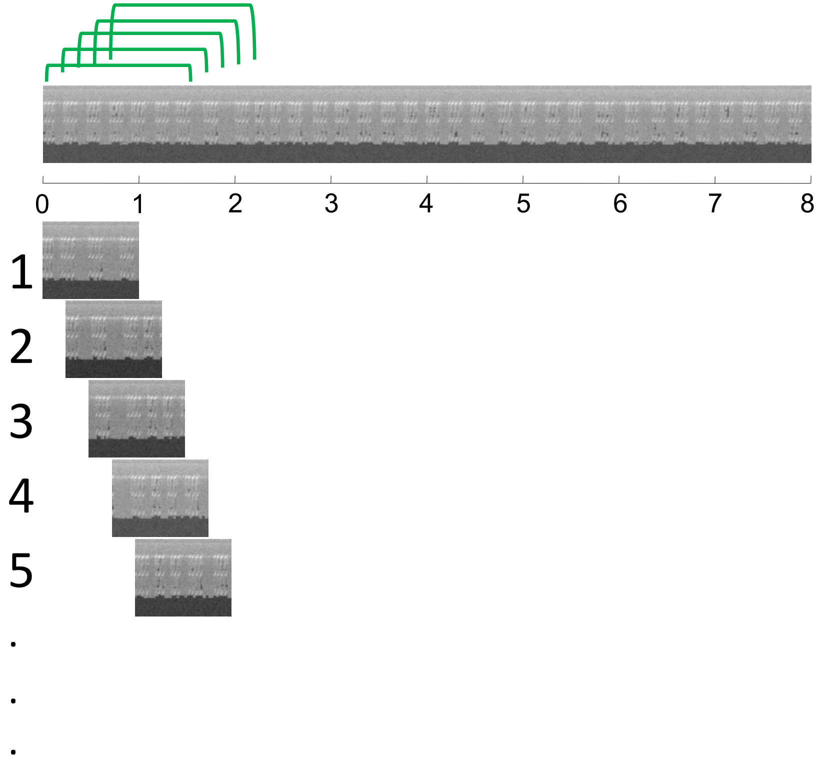
Using a sliding window with the same size of candidate’ s, locate the minimum distance for each object in **P** and **U**

Step 3:

Evaluation mechanism for splitting datasets into two groups

Step 4:

Sound fingerprint with the best splitting point, which is the one can produce the largest information gain to separate two classes



Evaluation Mechanism

Step3: Information gain to evaluate candidate splitting rules

$$E(\mathbf{D}) = -p(X)\log(p(X))-p(Y)\log(p(Y))$$

where X and Y are two classes in $\mathbf{D}$

$$Gain = E(\mathbf{D}) - E'(\mathbf{D})$$

where $E(\mathbf{D})$ and $E'(\mathbf{D})$ are the entropy before and after partitioning $\mathbf{D}$ into $\mathbf{D_1}$ and $\mathbf{D_2}$ respectively.

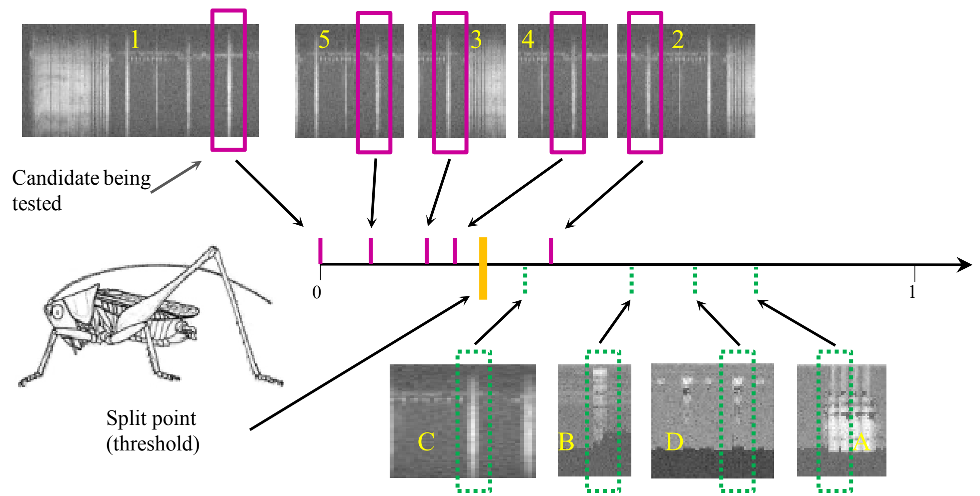
$$E'(\mathbf{D}) = f(\mathbf{D_1})E(\mathbf{D_1}) + f(\mathbf{D_2})E(\mathbf{D_2})$$

where $f(\mathbf{D_1})$ is the fraction of objects in $\mathbf{D_1}$, and $f(\mathbf{D_2})$ is the fraction of objects in $\mathbf{D_2}$.

Example

A total of nine objects, five from **P**, and four from **U**.
This gives us the entropy for the unsorted data
$[-(5/9)\log(5/9)-(4/9)\log(4/9)] = 0.991$

Information Gain = 0.991- 0.401 = **0.590**

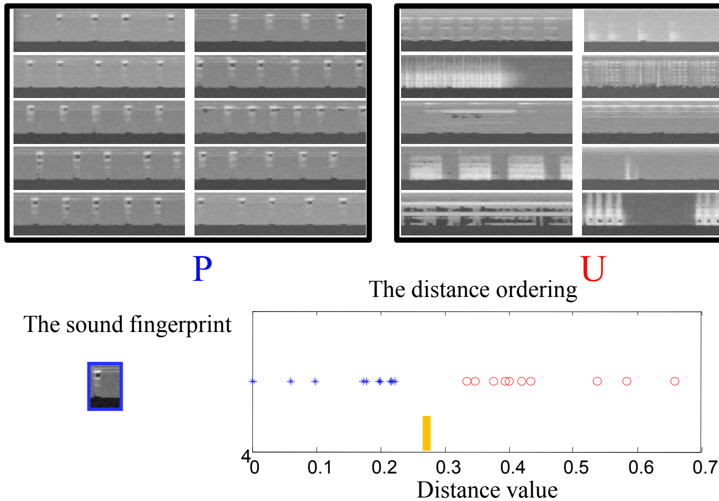


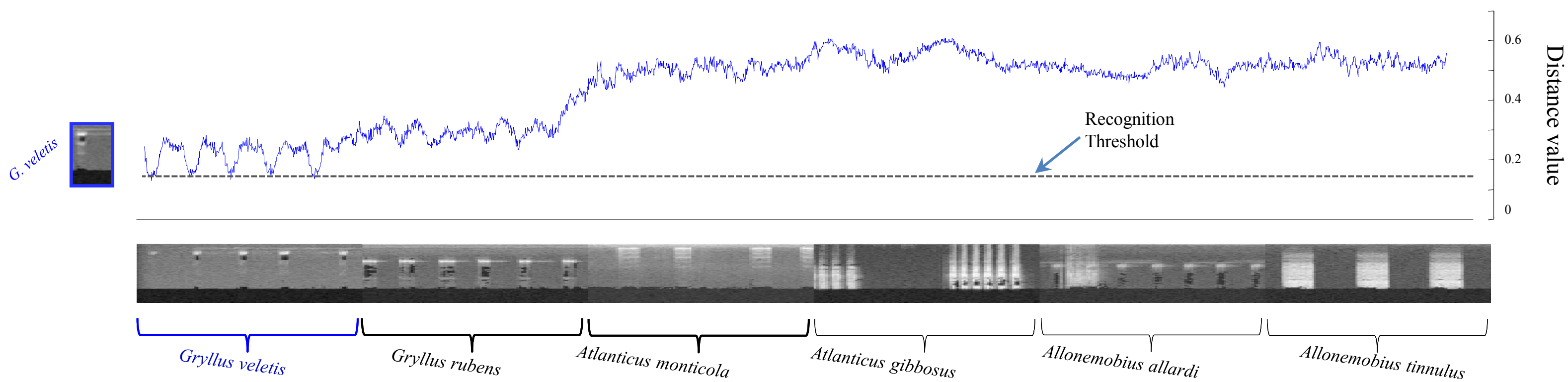
Four objects from **P** are the only four objects on the left side of the split point. Of the five objects to the right of the split point we have four objects from *U* and just one from **P**
$(4/9)[-(4/4)\log(4/4)]+(5/9)[-(4/5)\log(4/5)-(1/5)\log(1/5)] = 0.401$

Outline

- Motivation
- Our approach
  - Visual space-spectrogram
  - CK distance measure
  - Sound fingerprint searching
- Experimental evaluation
  - Brute force search evaluation
  - Speed up and efficiency
- Conclusion & future work

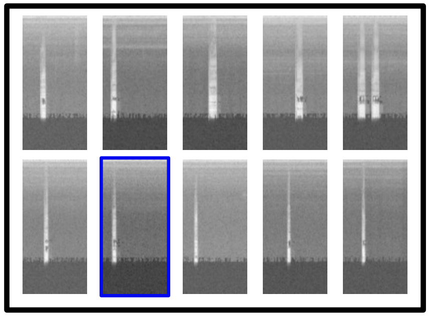
Example



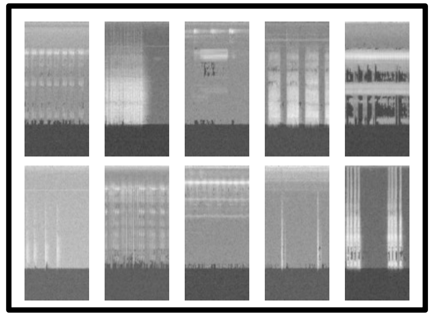


A demonstration of brute force search algorithm and the discrimination ability of the CK measure.

One short template of insect sounds is scanned along a long sequence of sound, which contains one example of the target sound, plus three examples commonly confused insect sounds



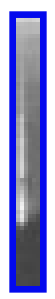
P

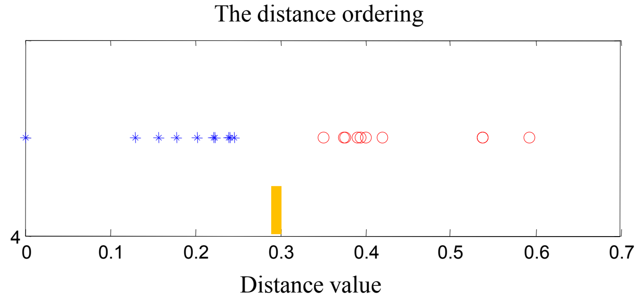


U

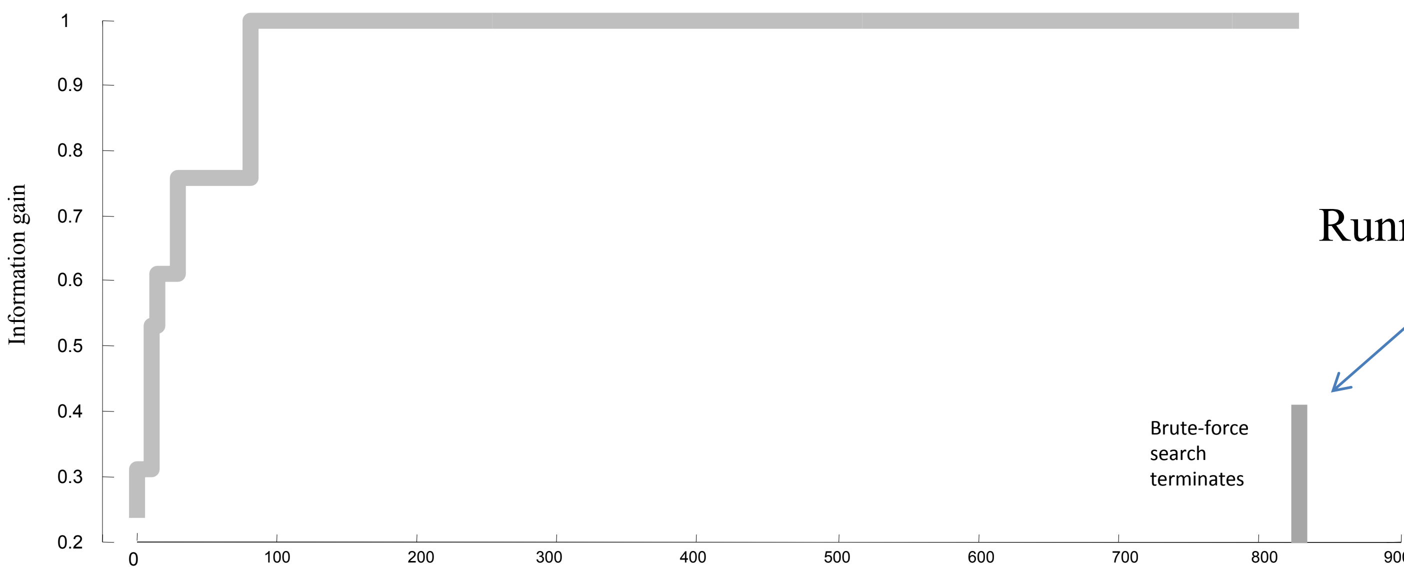
P = *Atlanticus dorsalis*

The sound fingerprint

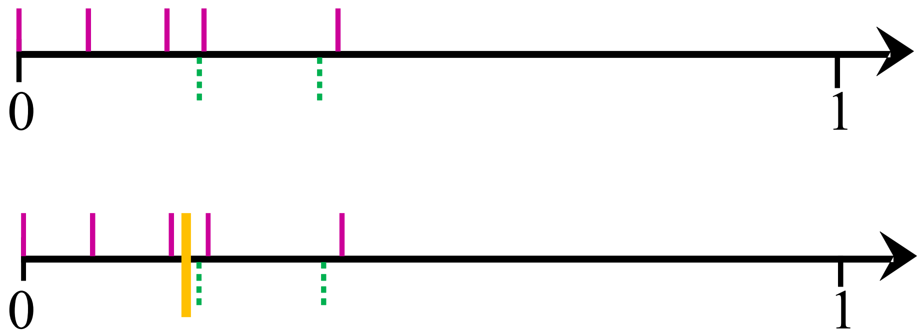


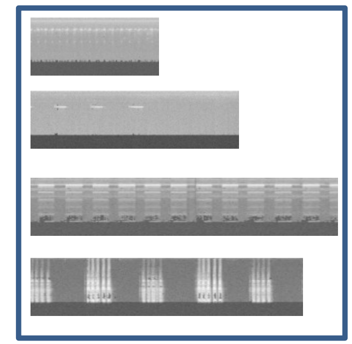


Running time: 7.5 hours



Speedup by Entropy-based Pruning





U

Before split: [-(5/9)log(5/9)-(4/9)log(4/9)] = 0.991

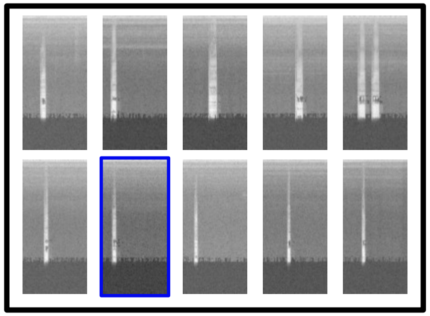
After split: (3/9)[-(3/3)log(3/3)]+(6/9)[-(4/6)log(4/6)-(2/6)log(2/6)] = 0.612

Best-so-far Information Gain

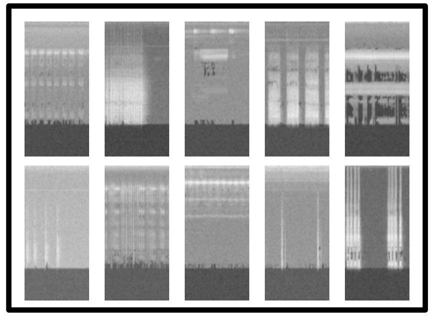
0.991- 0.401 = **0.590**

∨

Upper bound Information Gain = 0.991- 0.612= **0.379**



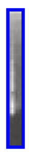
P

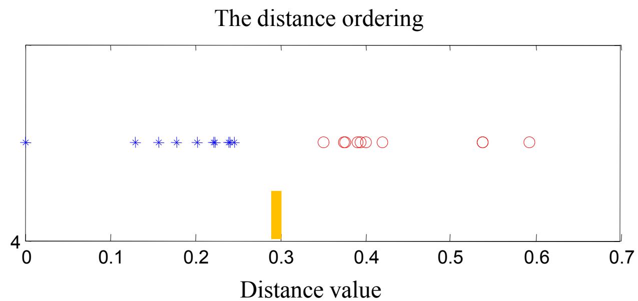


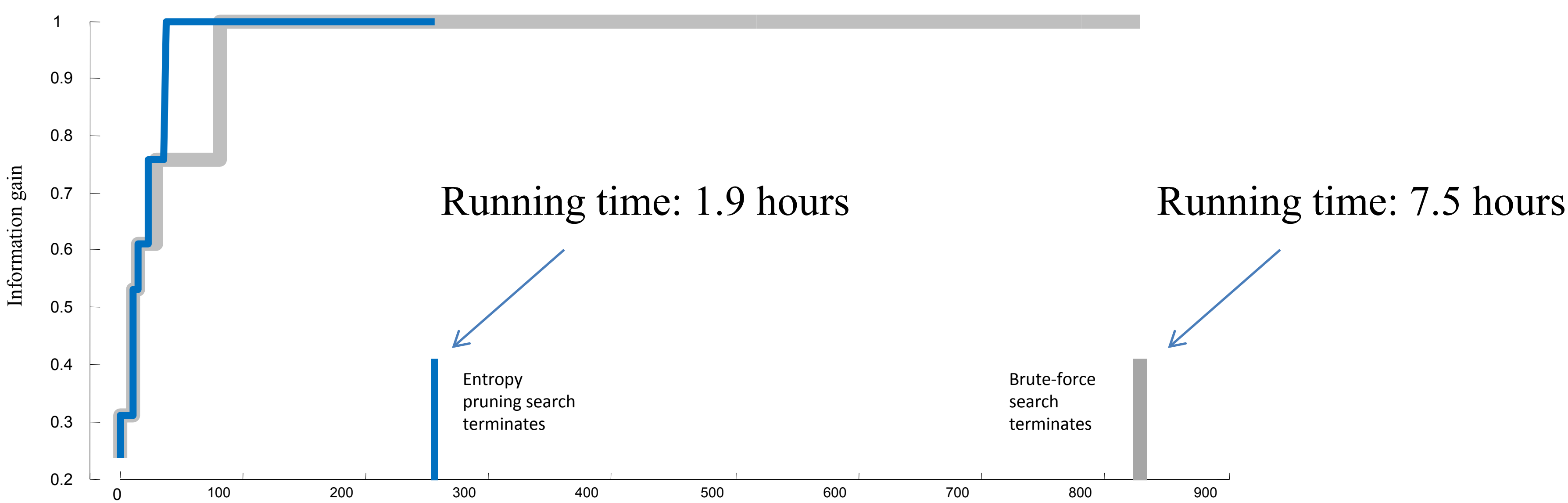
U

P = *Atlanticus dorsalis*

The sound fingerprint







Brute force search

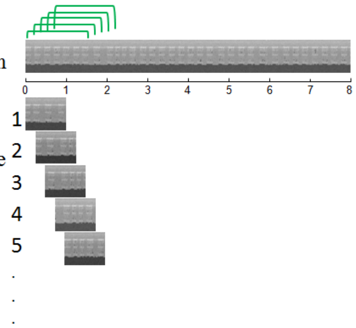
Generate and Evaluate

Step 1:
Given **P** and **U**, Generate all possible subsequences from the objects in **P** of length *m* as the sound fingerprint candidates.

Step 2:
Using a sliding window with the same size of candidate's, locate the minimum distance for each objects in **P** and **U**

Step 3:
Evaluation mechanism for splitting datasets into two groups

Step 4:
Sound fingerprint with best splitting point, which is the one can produce the largest information gain to separate two classes



In brute-force search, we search left to right, top to bottom

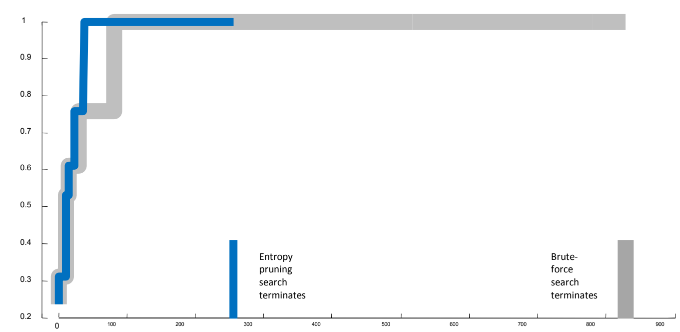
Is there a better order?
How can we find a good candidate earlier?

The earlier we find a good candidate, the information gain is higher, the more instances we can prune.

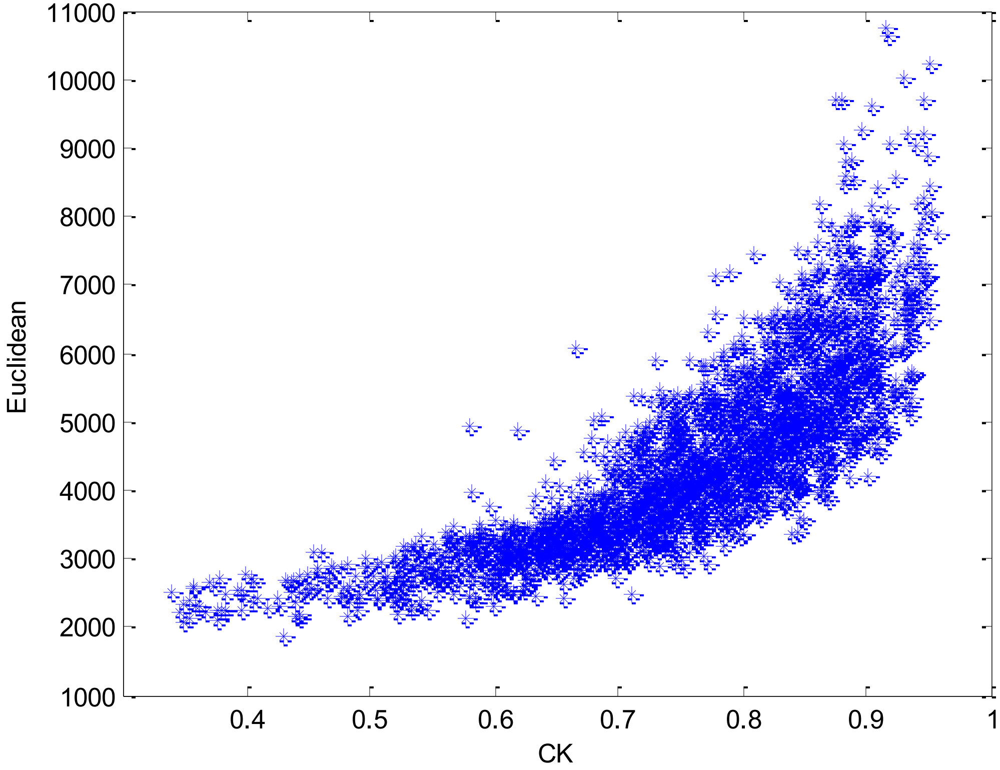
But how do we resolve this "*chicken and egg*" paradox?

Speedup intuition

- Euclidean distance is *much* faster than CK
- So let us use Euclidean distance to approximate the best search order for CK
- This will only work if Euclidean distance is a good proxy for CK…. (next slide)

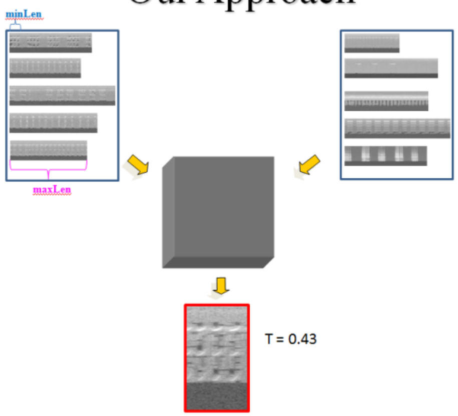


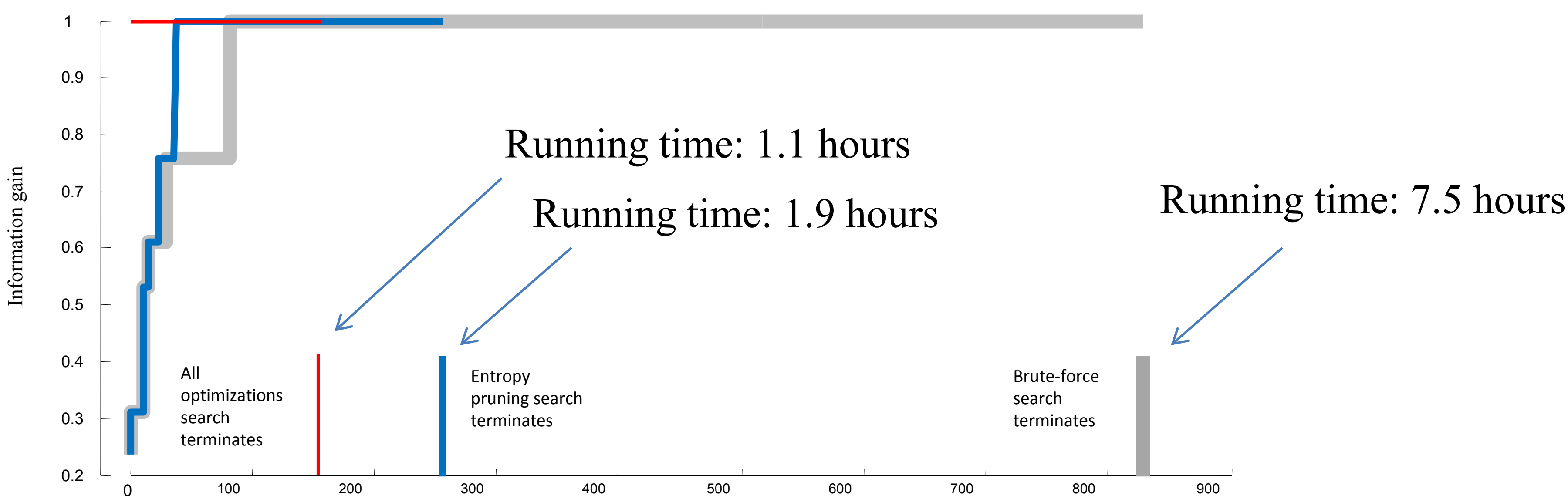
Euclidean Distance Measure Pruning



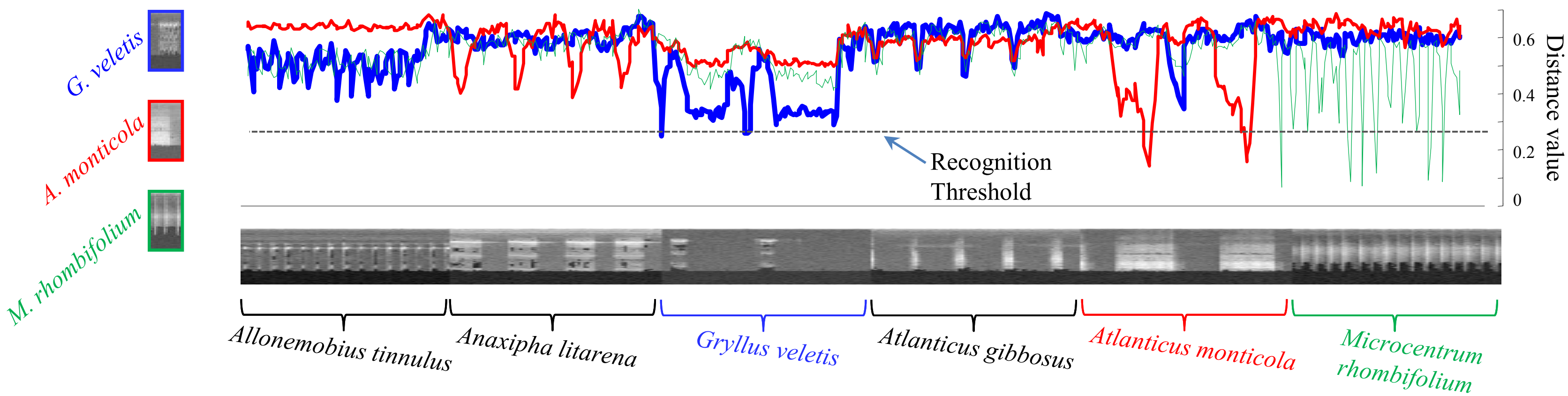
Performance of Optimization

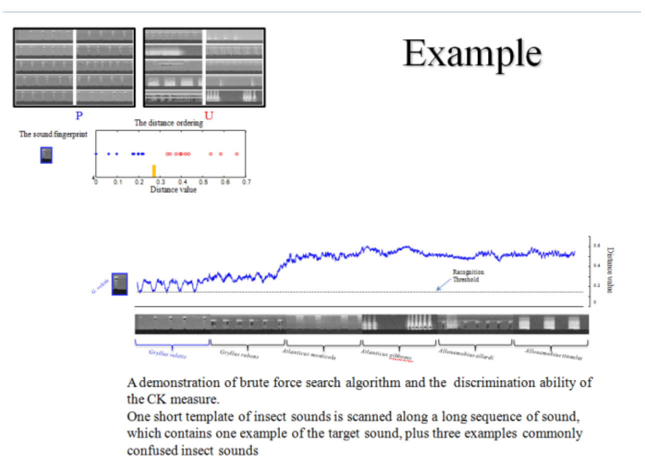
Our Approach





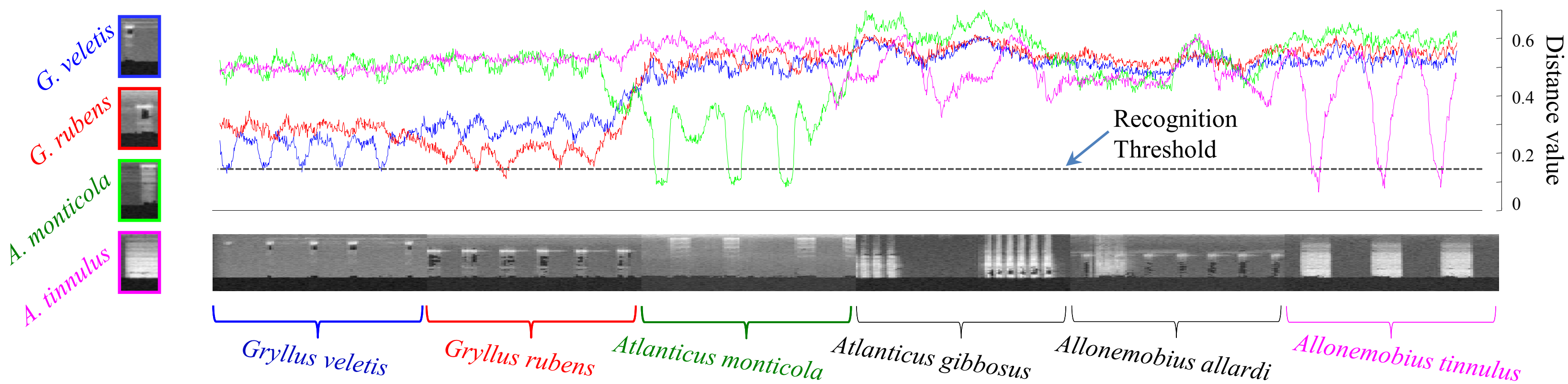
Case Study (1)

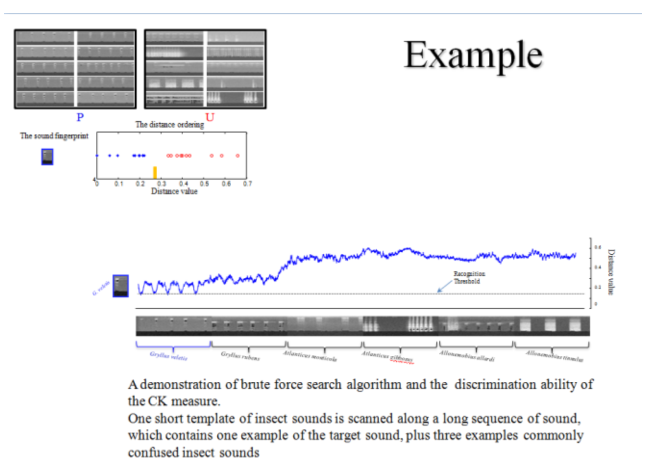




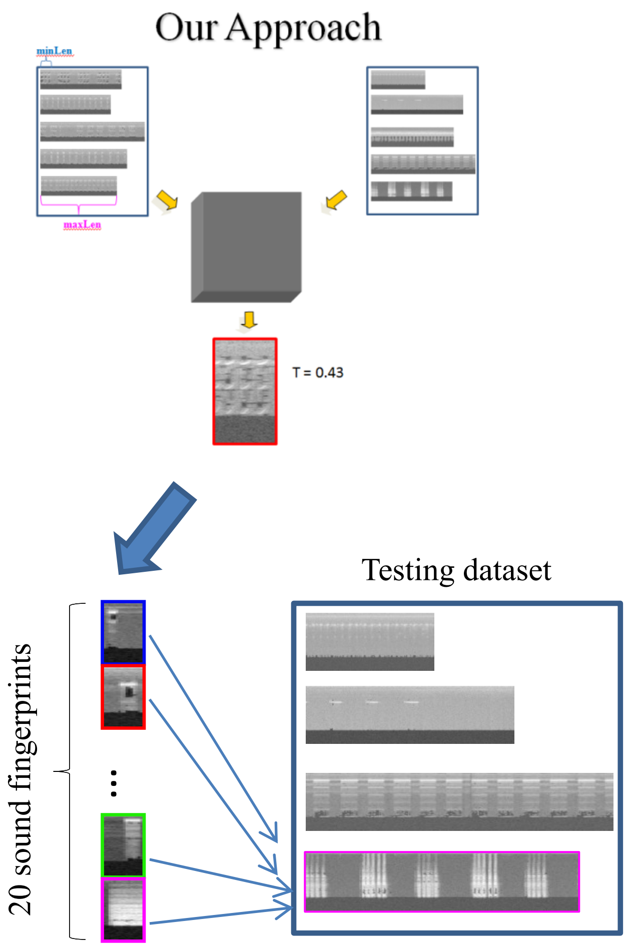
For more visual understanding, please take a look at the video on YouTube

Case Study (2)


G. veletis
G. rubens
A. monticola
A. tinnulus
Recognition Threshold
0.6
0.4
0.2
0
Distance value
Gryllus veletis
Gryllus rubens
Atlanticus monticola
Atlanticus gibbosus
Allonemobius allardi
Allonemobius tinnulus


Example
P
U
The sound fingerprint
The distance ordering
Distance value
A demonstration of brute force search algorithm and the discrimination ability of the CK measure.
One short template of insect sounds is scanned along a long sequence of sound, which contains one example of the target sound, plus three examples commonly confused insect sounds
26

Classification



Benchmark of insect classification:
The data consists of twenty species of insects, eight of which are Gryllidae (crickets) and twelve of which are Tettigoniidae (katydids)

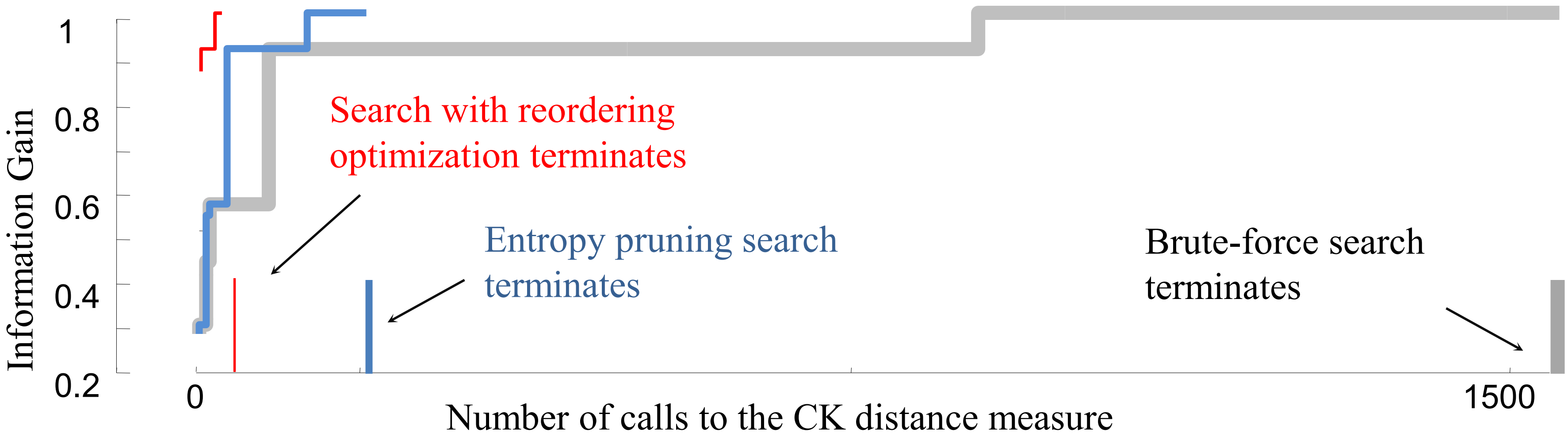
Problems: either a twenty-species level problem, or two-class genus level problem.

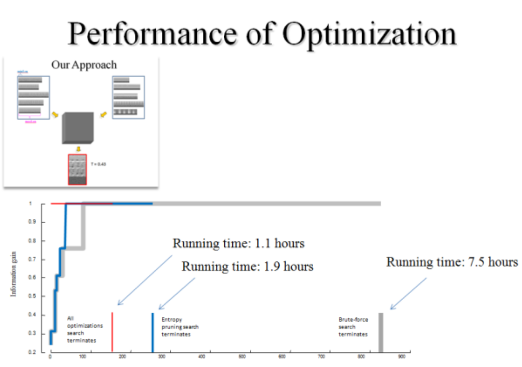
Method: predicted the testing exemplars class label (as the pink one shown on the left) by sliding each fingerprint across it and recording the fingerprint that produced the minimum value as the exemplar's nearest neighbor (the pink fingerprint).

Insect classification accuracy

	species-level problem		genus-level problem	
	default rate	fingerprint	default rate	fingerprint
10 species	0.10	0.70	0.70	0.93
20 species	0.05	0.44	0.60	0.77

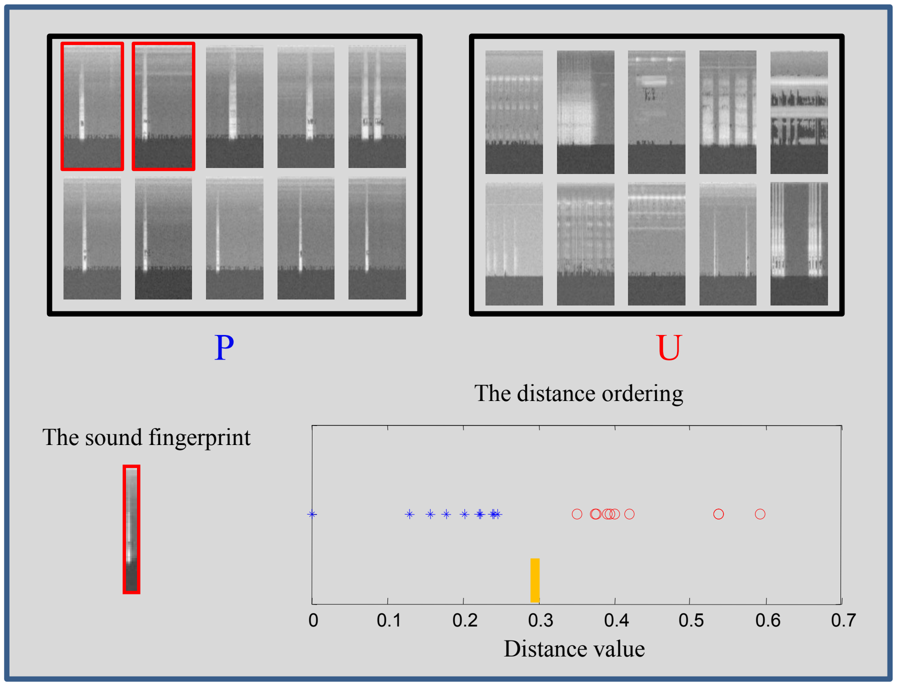
Scalability of Fingerprint Discovery

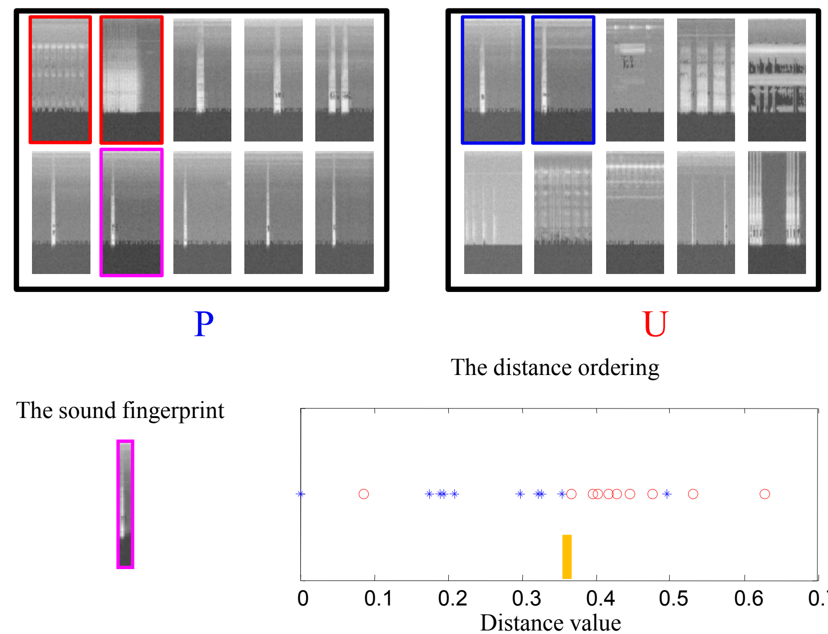




To test the speedup of our toy problem shown on the left,
we reran these experiments with a more realistically-sized universe U,
containing 200-objects from other insects, birds, trains, helicopters, etc.
The result is shown on above.

Mislabeled Data Sanity Check

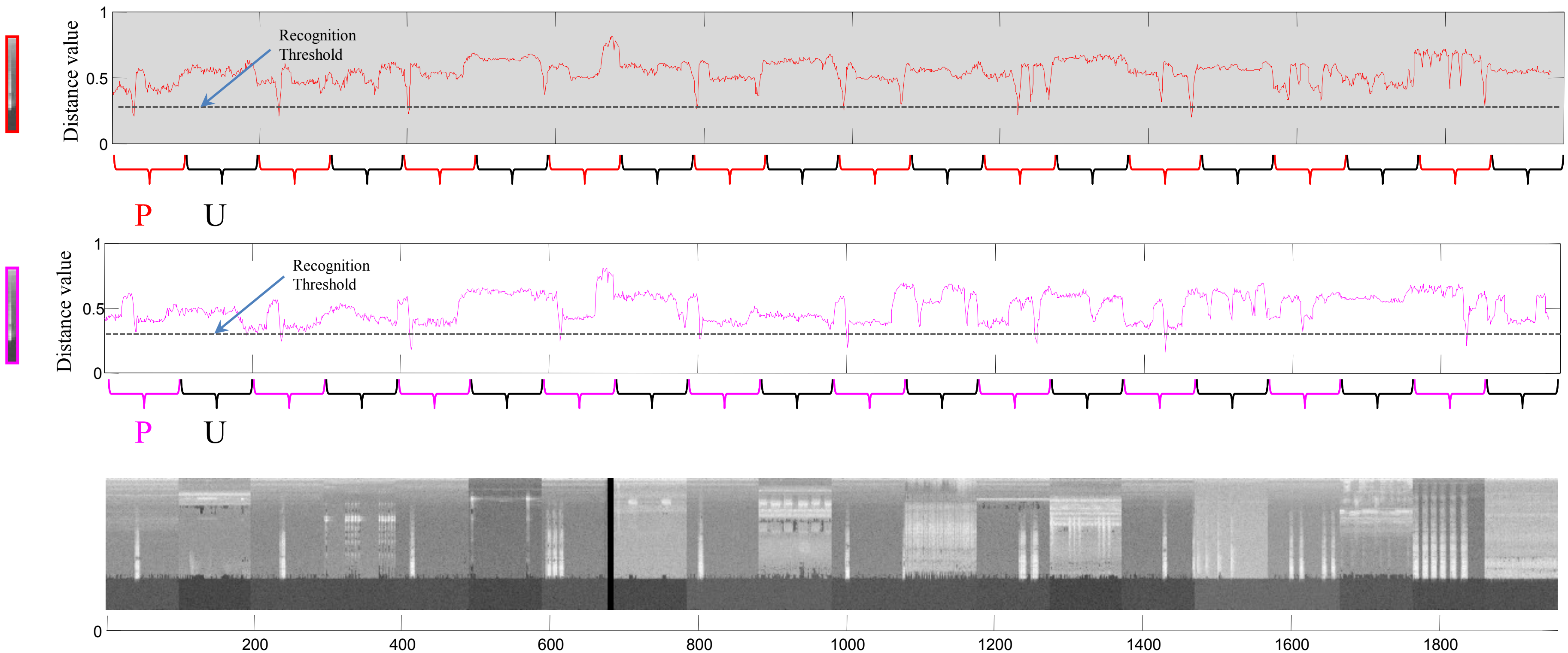




P = *Atlanticus dorsalis*

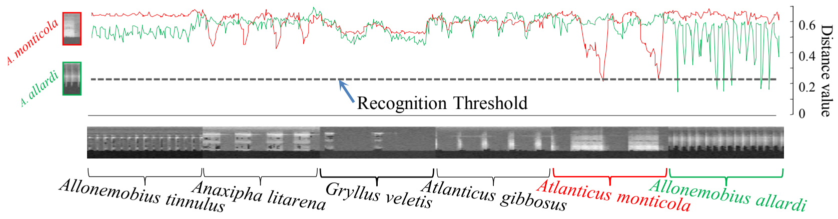
Same dataset for mislabel check
Left: assume all labeled correctly
Right: two instances in positive class mislabeled

Mislabeled Data Sanity Check

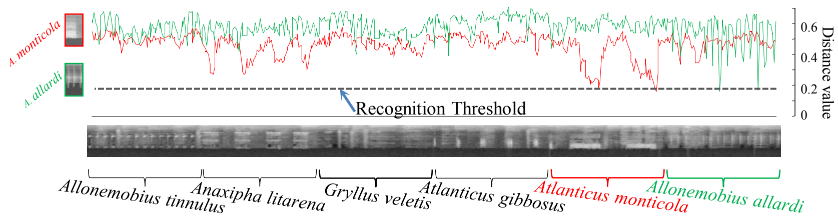


Same dataset for mislabel check
Top: assume all labeled correctly
Bottom: two instances in positive class mislabeled

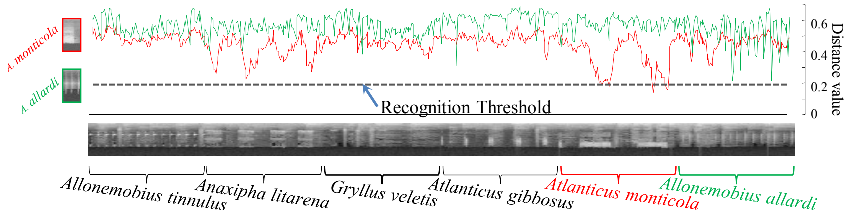
Noise background experiment



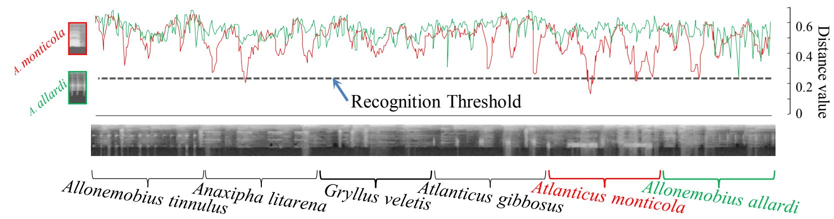
No noise



Noise: -5dB



Noise: -4dB



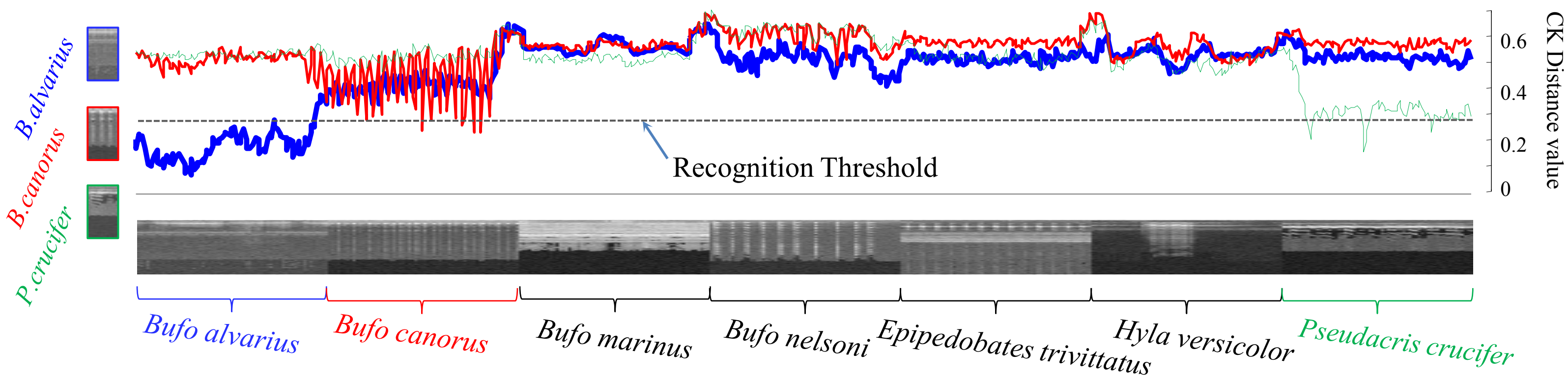
Noise: +5dB

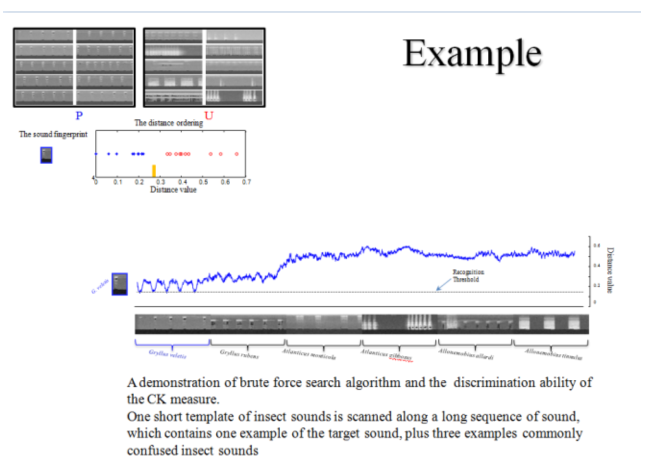
Classification

	species-level problem		genus-level problem	
	default rate	fingerprint	default rate	fingerprint
10 species	0.10	0.70	0.70	0.93
20 species	0.05	0.44	0.60	0.77

Twenty insect species datasets:
Eight of them are *Grylliadae* (crickets)
Twelve of them are *Tettigoniidae* (katydids)

Other animals-Frogs





Outline

- Motivation
- Our approach
  - Visual space-spectrogram
  - CK distance measure
  - Sound fingerprint searching
- Experimental evaluation
  - CK as a tool for taxonomy
  - Speed up and efficiency
- Conclusion & future work

Conclusion & Future Work

- Our approach to analyze insect sound in visual space is parameter free
- Our optimizations can speedup the brute-force search
- We will test more species and dataset
- We will further speedup the algorithm

Thank you

Code and Data:

http://www.cs.ucr.edu/~yhao/animalsoundfingerprint.html