MATLAB Audio Database Toolbox

User Manual

Version 1.00, July 2008

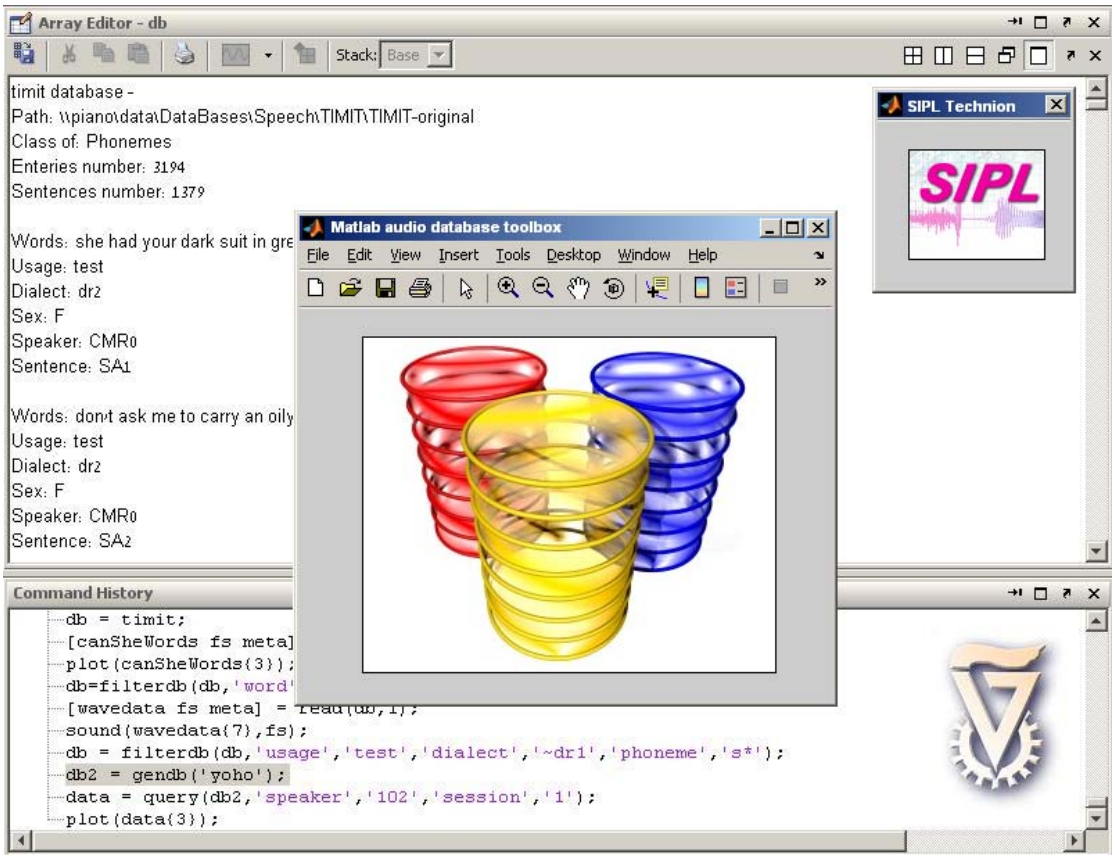

Table of Contents

Introduction

MatlabADT (Audio Database Toolbox) enables easy access and filtering of audio databases such as TIMIT and YOHO by their metadata. The database toolbox comes to replace the manual filtering and custom coding usually required for accessing such databases. This toolbox will save you the learning time of the database structure and will enable you to focus on algorithmic aspects of your code.

The following databases are supported:

1. **TIMIT** - Acoustic Phonetic Continuous Speech Corpus (American-English). Supported search criteria: word, phoneme, usage, sex, dialect, speaker and sentence. For more information on TIMIT see the appendix at the end of this document.
2. **NTIMIT** - Telephone Network Acoustic Phonetic Continuous Speech Corpus. Supported search criteria: word, phoneme, usage, sex, dialect, speaker and sentence. For more information on NTIMIT see the appendix at the end of this document.
3. **CTIMIT** - Cellular Telephone Acoustic Phonetic Continuous Speech Corpus.
4. **YOHO** - Speaker Verification Corpus. Supported search criteria: usage, speaker, session, numbers.
5. **TI-Digits** - Speaker-Independent recognition of connected digit sequences. Supported search criteria: usage, group, type, speaker and digit.
6. **Children Voices** - Hebrew Speech.
7. **Hebrew BGU** - Hebrew word samples were.
8. Gutenberg Books - MP3 format books.

For more information on database structures see the database documentation available on the SIPL site.

For any problem in using MatlabADT please contact: matlab_adt@sipl.technion.ac.il

Quick Start

Installation outside of SIPL:

a. Extract the files and add "MatlabADT" directory to your MATLAB path.
b. In MATLAB, execute the command:
   db = ADT('timit','C:\timit_path_on_your_computer','setup');
   This will load TIMIT and set the default path to the directory entered.
   For NTIMIT or CTIMIT run corresponding commands.
c. In case you want to use other databases: in MatlabADT\@gendb\instance directory enter the sub directories and change defalt_path.txt to the directory of the specific database on your computer.

Installation at SIPL workstations:

a. Add the "sipl_matlab_utils" directory to the MATLAB path by executing the command: addpath('\\piano\Data\sipl_matlab_utils'); or by using MATLAB menus.
b. Execute the command timitdemo(); to check installation.

Usage:

1. Load the desired database

Loading the database object:

Database name	Loading command
TIMIT	db = ADT('timit');
NTIMIT	db = ADT('ntimit');
CTIMIT	db = ADT('ctimit');
YOHO	db = gendb('yoho');
Ti-Digits	db = gendb('tidigits');
Children Voices	db = gendb(' children voices');
Hebrew BGU	db = gendb(' hebrew bgu');
Gutenberg Books	db = gendb(' gutenberg books');

ll operations on the database will be performed using the database object which is passed to them as the first parameter.

2. Make a query for your wanted data
   [wavdata] = query(db,'dialect','dr1','word',{'she','it'},30);
   This call to the query function will return the wave data of the first 30 words 'she' or 'it' form dialect 'dr1' in the form of a cell array

3. Accessing the wave data

```
oneWord = wavdata{1};
```

Function Overview

Function name	*Description*
`ADT/gendb`	Creates database object
`query`	Returns file data form database by filtering criteria
`filterdb`	Creates a subset of the database by filtering criteria
`read`	Returns file data form database
`play`	Plays a database entire
`getpath`	Returns the path of a specific entire
`length`	Returns the number of entries in the object

Function Reference Guide

Note the following usage examples are using the TIMIT database.

query

Usage	`[wave fs metadata] = query(`*`db`*`, [`*`criterion1`*`, `*`value1`*`, [`*`criterion2`*`,`*`value2`*`,…]],[max_returns]);`	
Arguments	Criteria list describing the query	
Return value	`wave`	cell array of waveforms
	`metadata`	structure array describing waveforms
	`fs`	sampling frequency

Remarks

This function queries the database for entries matching the "criterion – value" pairs. Criteria are the fields of the metadata defined by database structure. Following is a short description of these fields. It is optional to pass maximum amount of returned entries.

`value` parameter may be one of the following:

1. criterion content: the query will return only entries matching this value

   Example:

   ```
   query(db, 'sentence', 'SA1')
   ```

   returns all instances of sentence 'SA1'.
2. criterion content negation: if criterion value is preceded by ~ (tilde) then the query will return only those entries that does not match the specified value:

   Example:

   ```
   query (db, 'dialect', '~dr2')
   ```

   returns all sentences of dialect region other than dr2.
3. Cell array of criteria content or criteria content negation: the query will return entries matching either one of criteria.

Example:

```
wav = query(db, 'phoneme', {'d', 'p'});
```

returns all instances of phonemes 'd' or 'p'

4. criterion content *: either one of criteria.

   Example:

```
wav = query(db, 'word', ‘sh*’);
```

   returns all instances of words starting with sh

filterdb

Usage	`filtered_db = filterdb(`*`db`*`, [`*`criterion1`*`, `*`value1`*`, [`*`criterion2`*`, `*`value2`*`,…]], max_returns);`
Arguments	Criteria list describing the query
Return value	Subset of timit database passed as an argument

Remarks

Unlike `query` function, `filterdb` does not load all waveforms to memory but returns filtered database object. This is useful when the resulting set of the query is too big to fit in memory. Consequent calls to read function can be made to read content of this filtered database object.

See *Remarks* section of query function for the description of criteria arguments.

read

Usage	`[wave fs metadata] = read(`*`db`*`[, index]);`	
Arguments	timit database object and (optionally) index of the entry to be returned	
Return value	`wave`	cell array of waveforms
	`metadata`	structure array describing waveforms
	`fs`	sampling frequency

Remarks

Returns index entry in the database. If index argument is not specified, returns all entries from the database object passed as an argument.

play

Usage	`play(db,index);`
Arguments	database object, and index of the entire to play
Return value	null

getpath

Usage	`path = getpath(db,index);`
Arguments	database object, and index to an entire
Return value	The path to the file containing the entire

length

Usage	`n = length(db);`
Arguments	database object
Return value	Number of entries in the database

Usage Examples

Example 1

```
%TIMITDEMO - a timit database interface demonstration,
%Reads text by searching TIMIT database for the words.

db =ADT('timit');      %loads TIMIT database interface
db = filterdb(db,'sex','m');  %filters database for male speakers
text = 'have a good time using this program and enjoy yourself i hope
this program is useful';
volume = 1;  %assigning the volume
text =strread(text,'%s');%converting Text form string to cell array
for ii=1:length(text)
    [word smpr] = query(db,'word',text{ii},1);%querying for the ii
word  returnig the first match
    if ~isempty(word)
        max_v = max(word{1});   %normalizing the volume
        word{1}=word{1}.*(volume/max_v); %normalizing the volume
        sound(word{1},smpr);%play word (words return form query in
the form of cell array
        pause(0.2);   %pauses for 0.2s
    end
end
```

Example 2

```
%GENDBDEMO a demonstration of gendb
%loading YOHO database
  db = gendb('yoho');
  %filtering YOHO database returning 3 first eateries matching the
query
  db = filterdb(db,'usage','verify','speaker','10*',3);
  %loading Hebrew BGU database
  db2 = gendb('Hebrew BGU');
  %selecting a random index
  index = fix(rand(1)*length(db2))+1;
  %playing index entire
  play(db2,index);
  %reading wave data from filtered YOHO database
  wavedata = read(db);
  %plotting entire number 2
  plot(wavedata{2});
```

Appendix - TIMIT/NTIMIT Fields

Filed	*Content*	*Description*
Usage	*train, test*	Train/test set defined by TIMIT
dialect	*dr1, dr2, dr3, dr4, dr5, dr6, dr7, dr8*	Dialect regions as defined in TIMIT documentation
Sex	*m,f*	Gender of the speaker
speaker	Four alphanumeric characters	The id of the speaker as given by TIMIT
sentence	Up to 5 characters. Format defined in TIMIT documentation	Sentence id as defined by TIMIT
word	*Word*	Any word from a sentence present in TIMIT
phoneme	*Phoneme*	Any phoneme from a word present in TIMIT. Phoneme codes can be found in TIMIT documentation

The TIMIT/NTIMIT database eateries can take the form of sentences words or phonemes.

The query or read functions will return a cell array, The returned cell array of waveforms will contain waveforms of entire sentences, words or phonemes, depends whether the query result is sentence, word or phoneme. The query result is determined by the "finest" query term in a query.

Query example	*Finest term in a query*	*Content of each cell in a returned cell array*
`query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm')`	*'sentence', 'dialect', 'sex'*	*Sentence*
`query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm', 'word', 'she')`	*'word'*	*Word*

*query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm', **'phoneme', 's'**)*	*'phoneme'*	*Phoneme*
*query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm', **'word', '#all'**)*	*'word'*	*Word*
*query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm', **'phoneme', '#all'**)*	*'phoneme'*	*Phoneme*

Special value '#all' can be used to force finest term in a query without actually filtering by that criterion. For example `query(db, 'sentence', 'SA1', 'dialect', 'dr1', 'sex', 'm', 'phoneme', '#all')` will return all phonemes in the requested sentence.