

Laboratorio di Informatica Musicale

mercoledì 17 ottobre 2007

Corso di Laurea specialistica in Ingegneria informatica, a.a. 2007/2008

Esercitazione 1: **Sound analysis**

Luca Mion

`luca.mion@dei.unipd.it`

- Per accedere alle macchine in aula Te è sufficiente un account studenti
 - La procedura per l'attivazione di un account studenti è descritta qui:
<http://www.dei.unipd.it/wdyn/?IDsezione=1018>
 - Chi non ha attivato il proprio account è pregato di farlo quanto prima!
-
- Portare cuffie o auricolari!
 - Firmare il foglio presenze.

- Per accedere alle macchine in aula Te è sufficiente un account studenti
 - La procedura per l'attivazione di un account studenti è descritta qui:
<http://www.dei.unipd.it/wdyn/?IDsezione=1018>
 - Chi non ha attivato il proprio account è pregato di farlo quanto prima!
-
- Portare cuffie o auricolari!
 - Firmare il foglio presenze.

Informazioni sul laboratorio (2)

- Homepage del corso
<http://www.dei.unipd.it/~musica/>
- Homepage del laboratorio
[http://www.dei.unipd.it/~musica/IM07/Laboratorio07/
laboratorio07.html](http://www.dei.unipd.it/~musica/IM07/Laboratorio07/laboratorio07.html)
- Homepage del gruppo di ricerca
<http://smc.dei.unipd.it>

Calendario delle esercitazioni (tutte in aula Te)

- mercoledì 17 ottobre 2007, 12:30 - 14:15 (I laurea spec.)
- giovedì 18 ottobre 2007, 10:30 - 12:15 (II laurea spec.)
- mercoledì 31 ottobre 2007, 12:30 - 14:15 (I laurea spec.)
- mercoledì 31 ottobre 2007, 16:30 - 18:15 (II laurea spec.)
- giovedì 15 novembre 2007, 10:30 - 12:15
- giovedì 29 novembre 2007, 10:30 - 12:15

Calendario delle esercitazioni (tutte in aula Te)

- mercoledì 17 ottobre 2007, 12:30 - 14:15 (I laurea spec.)
- giovedì 18 ottobre 2007, 10:30 - 12:15 (II laurea spec.)
- mercoledì 31 ottobre 2007, 12:30 - 14:15 (I laurea spec.)
- mercoledì 31 ottobre 2007, 16:30 - 18:15 (II laurea spec.)
- giovedì 15 novembre 2007, 10:30 - 12:15
- giovedì 29 novembre 2007, 10:30 - 12:15

Ogni esercitazione consiste di:

- Prima parte introduttiva
(descrizione del problema, esempi, ecc.)
- Seconda parte realizzativa
(elaborazione di esempi, esercizi, ecc.)

Per tutti:

- Possibilità di tesine (vedere il sito del corso)

Esercitazione 1:

Sound analysis

Materiale per la prima esercitazione

- scaricare dalla homepage del laboratorio gli script MATLAB e i file audio:

<http://www.dei.unipd.it/~musica/IM07/Laboratorio07/Lab1matlab.zip>

- scompattare **tutto** in una sola directory (ad es. in C:\musica\Lab1matlab)

- Per abilitare l'audio:

pannello di controllo → periferiche audio → impostazioni → attiva

- aprire MATLAB
- posizionarsi nella directory contenente gli esempi

```
>> cd 'C:\musica\Lab1matlab'
```

Materiale per la prima esercitazione

- scaricare dalla homepage del laboratorio gli script MATLAB e i file audio:

<http://www.dei.unipd.it/~musica/IM07/Laboratorio07/Lab1matlab.zip>

- scompattare **tutto** in una sola directory (ad es. in C:\musica\Lab1matlab)
- Per abilitare l'audio:
pannello di controllo → periferiche audio → impostazioni → attiva
- aprire MATLAB
- posizionarsi nella directory contenente gli esempi

```
>> cd 'C:\musica\Lab1matlab'
```

Materiale per la prima esercitazione

- scaricare dalla homepage del laboratorio gli script MATLAB e i file audio:

<http://www.dei.unipd.it/~musica/IM07/Laboratorio07/Lab1matlab.zip>

- scompattare **tutto** in una sola directory (ad es. in C:\musica\Lab1matlab)
- Per abilitare l'audio:
pannello di controllo → periferiche audio → impostazioni → attiva
- aprire MATLAB
- posizionarsi nella directory contenente gli esempi
>> cd 'C:\musica\Lab1matlab'

```
>> esempio1
```

- read and play audio files
- time analysis, temporal envelope
- frequency analysis, DFT
- basic processing examples
- basic filter example (low-pass IIR)

```
>> esempio2
```

- Short-Time Average Energy, Short-Time Average Magnitude
- zero-Crossing rate
- resampling
- autocorrelation
- pitch extraction (via autocorrelation)
- pitch extraction (via FFT)
- pitch tracking via FFT

```
>> esempio3
```

- spectral centroid
- spectrograms (e.g. violin vs. snare drum)
- windowing

Beat tracking

Algorithm from Columbia University (GNU GPL)
winner of the MIREX context in 2006 (Audio Cover Song section)
http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

```
>> cd beat
```

Load a song (mp3 file) and force the data to be mono, downsampled by a factor of 4:

```
>> [d,sr]=mp3read('beatles.mp3',[500000,550000],1,4);
```

Calculate the beat times in the audio waveform d (at sampling rate sr):

```
>> b = beat(d,sr);
```

Resynthesize a blip-track of the same length:

```
>> db = mkblips(b,sr,length(d));
```

Listen to them mixed together:

```
>> soundsc(d+db,sr);
```

Beat tracking

Algorithm from Columbia University (GNU GPL)
winner of the MIREX context in 2006 (Audio Cover Song section)
http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

```
>> cd beat
```

Load a song (mp3 file) and force the data to be mono, downsampled by a factor of 4:

```
>> [d,sr]=mp3read('beatles.mp3',[500000,550000],1,4);
```

Calculate the beat times in the audio waveform d (at sampling rate sr):

```
>> b = beat(d,sr);
```

Resynthesize a blip-track of the same length:

```
>> db = mkblips(b,sr,length(d));
```

Listen to them mixed together:

```
>> soundsc(d+db,sr);
```

Beat tracking

Algorithm from Columbia University (GNU GPL)
winner of the MIREX context in 2006 (Audio Cover Song section)
http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

```
>> cd beat
```

Load a song (mp3 file) and force the data to be mono, downsampled by a factor of 4:

```
>> [d,sr]=mp3read('beatles.mp3',[500000,550000],1,4);
```

Calculate the beat times in the audio waveform d (at sampling rate sr):

```
>> b = beat(d,sr);
```

Resynthesize a blip-track of the same length:

```
>> db = mkblips(b,sr,length(d));
```

Listen to them mixed together:

```
>> soundsc(d+db,sr);
```

Beat tracking

Algorithm from Columbia University (GNU GPL)
winner of the MIREX context in 2006 (Audio Cover Song section)
http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

```
>> cd beat
```

Load a song (mp3 file) and force the data to be mono, downsampled by a factor of 4:

```
>> [d,sr]=mp3read('beatles.mp3',[500000,550000],1,4);
```

Calculate the beat times in the audio waveform d (at sampling rate sr):

```
>> b = beat(d,sr);
```

Resynthesize a blip-track of the same length:

```
>> db = mkblips(b,sr,length(d));
```

Listen to them mixed together:

```
>> soundsc(d+db,sr);
```

Beat tracking

Algorithm from Columbia University (GNU GPL)
winner of the MIREX context in 2006 (Audio Cover Song section)
http://www.music-ir.org/mirex2006/index.php/Audio_Cover_Song

```
>> cd beat
```

Load a song (mp3 file) and force the data to be mono, downsampled by a factor of 4:

```
>> [d,sr]=mp3read('beatles.mp3',[500000,550000],1,4);
```

Calculate the beat times in the audio waveform d (at sampling rate sr):

```
>> b = beat(d,sr);
```

Resynthesize a blip-track of the same length:

```
>> db = mkblips(b,sr,length(d));
```

Listen to them mixed together:

```
>> soundsc(d+db,sr);
```

Tempo (metronome) in BPM

Estimates the tempo of the audio waveform in d (at sampling rate sr) and returns two tempo in BPM:

$t(1)$ = lower BPM estimate

$t(2)$ = faster BPM estimate

$t(3)$ = is the relative weight for $t(1)$ compared to $t(2)$:

→ $t(1)$ is the best candidate when $t(3) > 0.5$

```
>> t=tempo(d,sr);
```

Provate con i vostri mp3!

Tempo (metronome) in BPM

Estimates the tempo of the audio waveform in d (at sampling rate sr) and returns two tempo in BPM:

$t(1)$ = lower BPM estimate

$t(2)$ = faster BPM estimate

$t(3)$ = is the relative weight for $t(1)$ compared to $t(2)$:

→ $t(1)$ is the best candidate when $t(3) > 0.5$

```
>> t=tempo(d,sr);
```

Provate con i vostri mp3!

Ordinate i vostri mp3 secondo il tempo in BPM

- attenzione ai percorsi relativi
- applicate `mp3read` ad una porzione di brano, non a tutto!
- sottocampionate i file per velocizzare la procedura

Aprirete un account sul sito collaborativo

<http://freesound.iua.upf.edu/>

Scaricate 2 file audio da freesound: un suono armonico (ad es. nota di piano) e un suono non armonico

Per ciascuno dei file dovete:

- analizzarne durata, frequenza di campionamento, bitrate
- plottare l'involuppo temporale e lo spettrogramma
- sottocampionare i file con fattore a vostra scelta e salvare il file audio risultante
- creare e salvare un file audio stereo, con un suono su ciascun canale
- salvare il file audio risultante dalla riproduzione inversa di uno dei file

Disabilitare l'audio prima di disconnettersi!