# Feature Extraction from Audio and their Application in Music Organization and transient Enhancement in Recorded Music

Jakob Frank, Thomas Lidy, Andreas Rauber
Vienna Univ. of Technology
Austria
Vincenzo Di Salvo, Massimo Magrini, Graziano Bertini
CNR – ISTI
Pisa, Italy

## Motivation and Goals

- Extract information from audio
- Aggregate it to higher-level semantic information
- Apply this information to different applications such as music browsing or improvement of acoustic perception

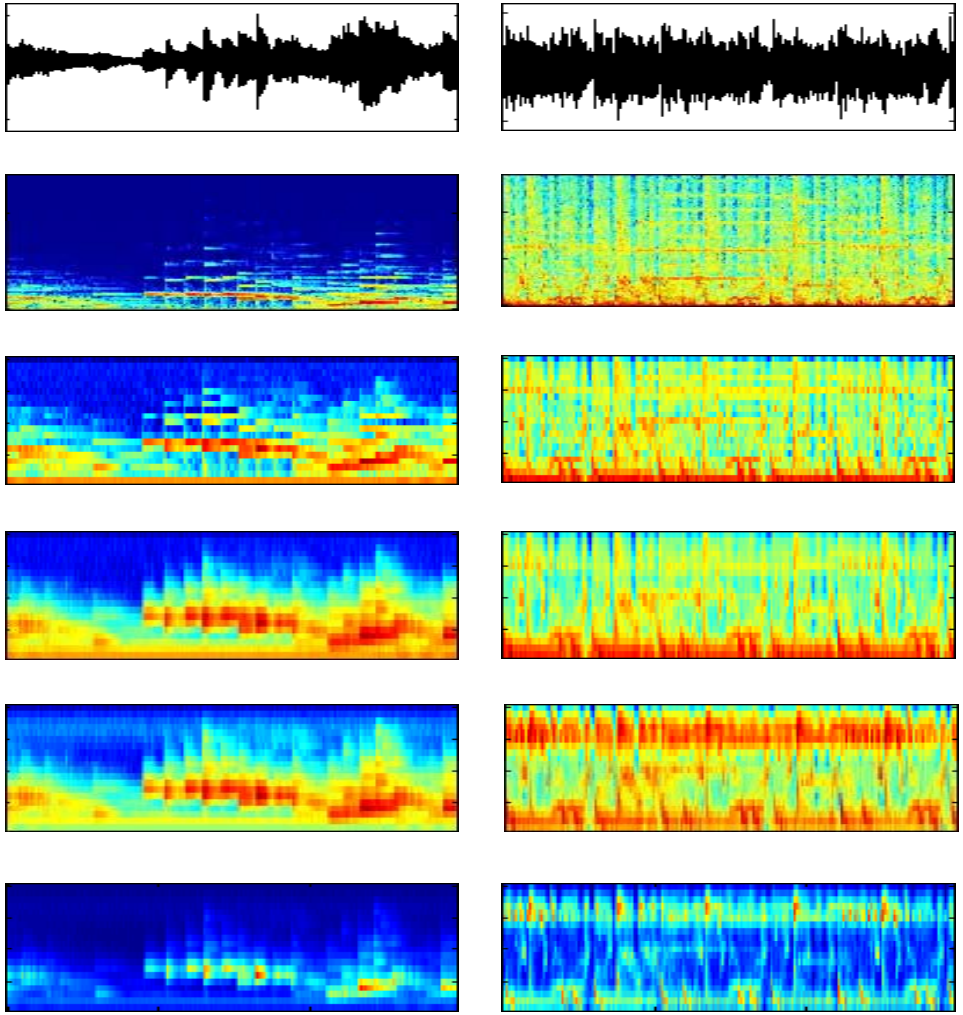
## Outline

- Feature Extraction:
  - RP, RH, SSD
  - Web Services for Feature Extraction
- Applications
  - SOMeJB/PocketSOM: Browsing Music Collections
  - ARIA: Improving sound quality
- Conclusions

# Audio Features: RP 1/2

Classical

Metal



PCM Audio Signal

Power Spectrum

Frequency Bands

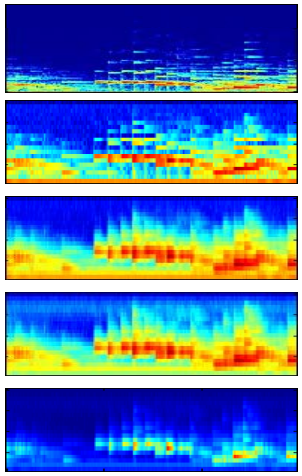
Masking Effects

Phon

Sone

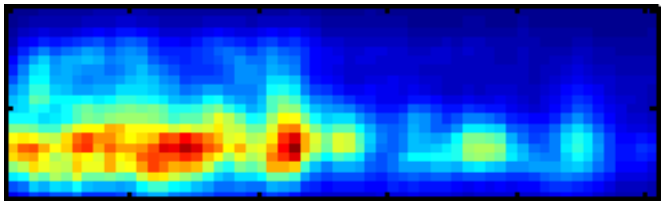
# Audio Features: SSD

Statistical Spectrum Descriptors (SSD)



**SSD**: 24*7=168-dimensional vector

24
critical
bands

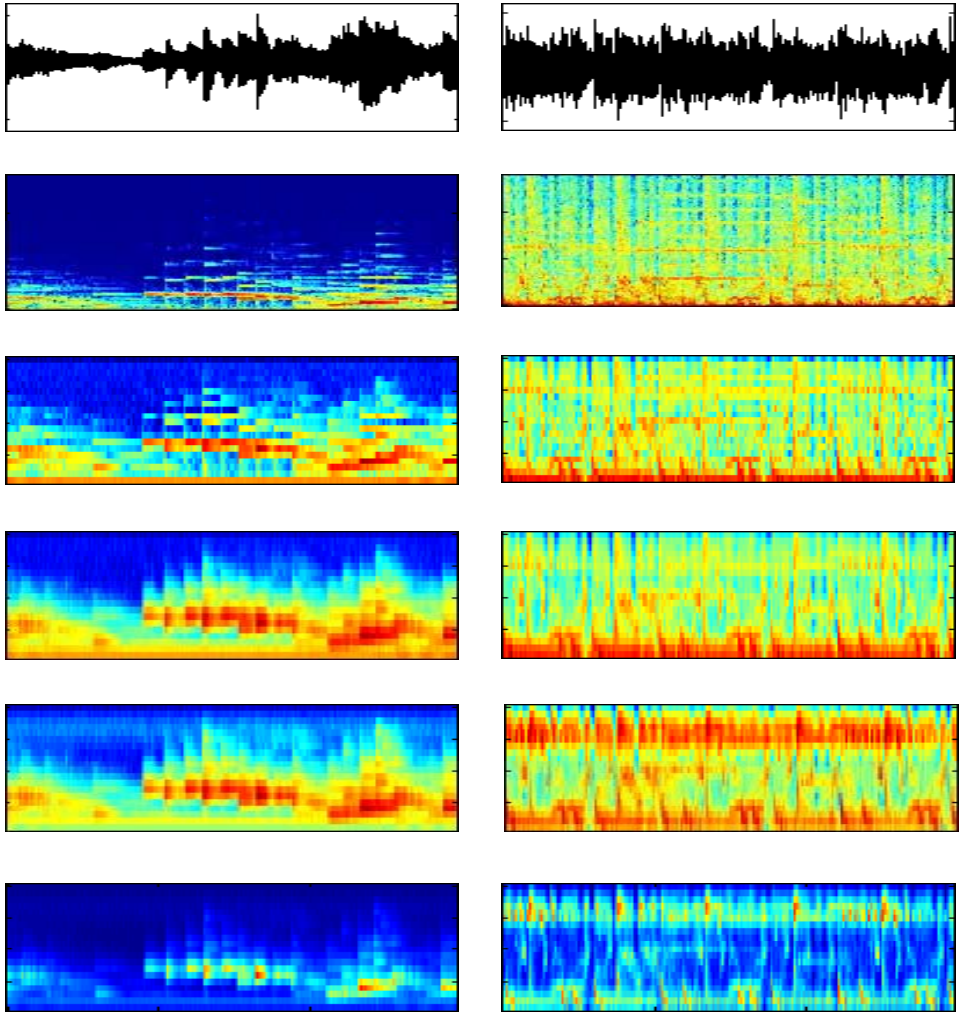


mean
median
variance
skewness
kurtosis
min
max

# Audio Features: RP 1/2

Classical

Metal



PCM Audio Signal

Power Spectrum

Frequency Bands

Masking Effects

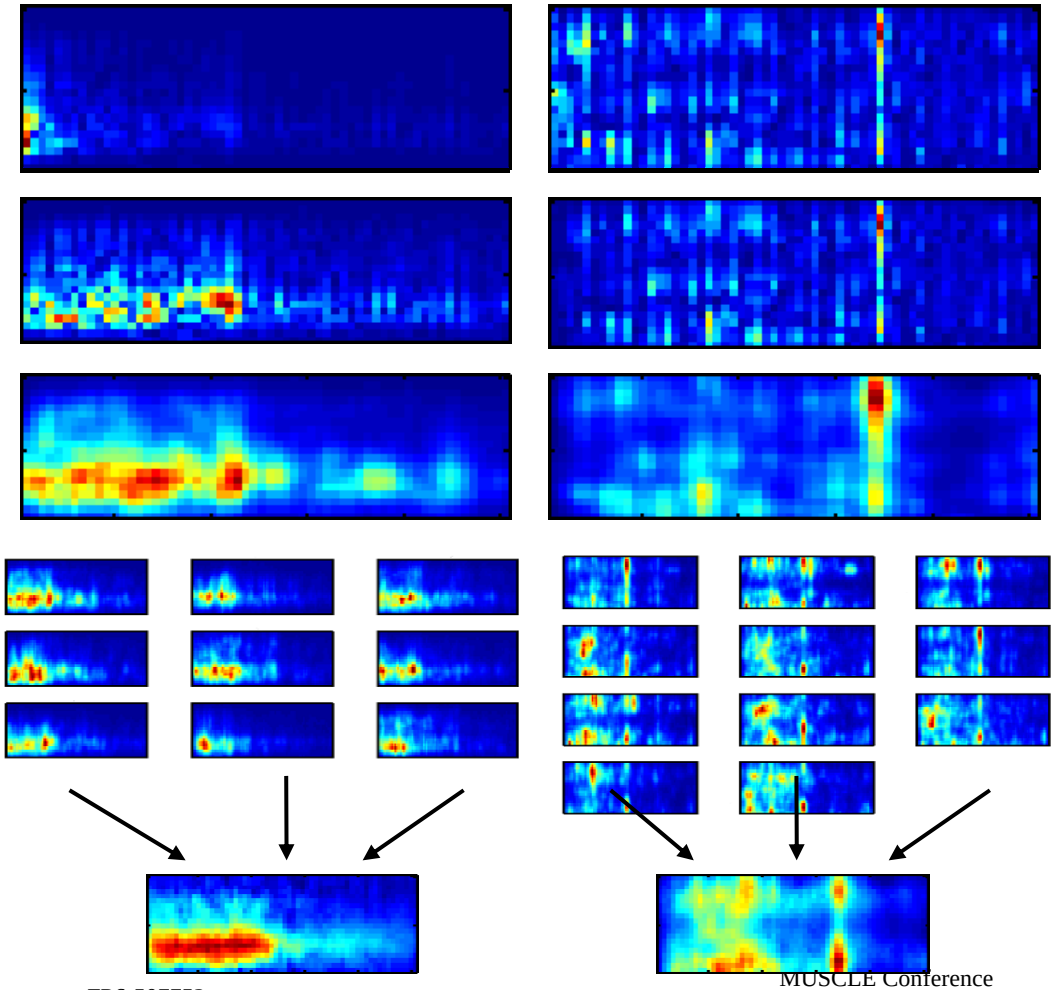
Phon

Sone

# Audio Features: RP 2/2

Classical

Metal



Loudness Modulation Amplitude (60 Values)

Fluctuation Strength

Filter (Gradient, Gauss)

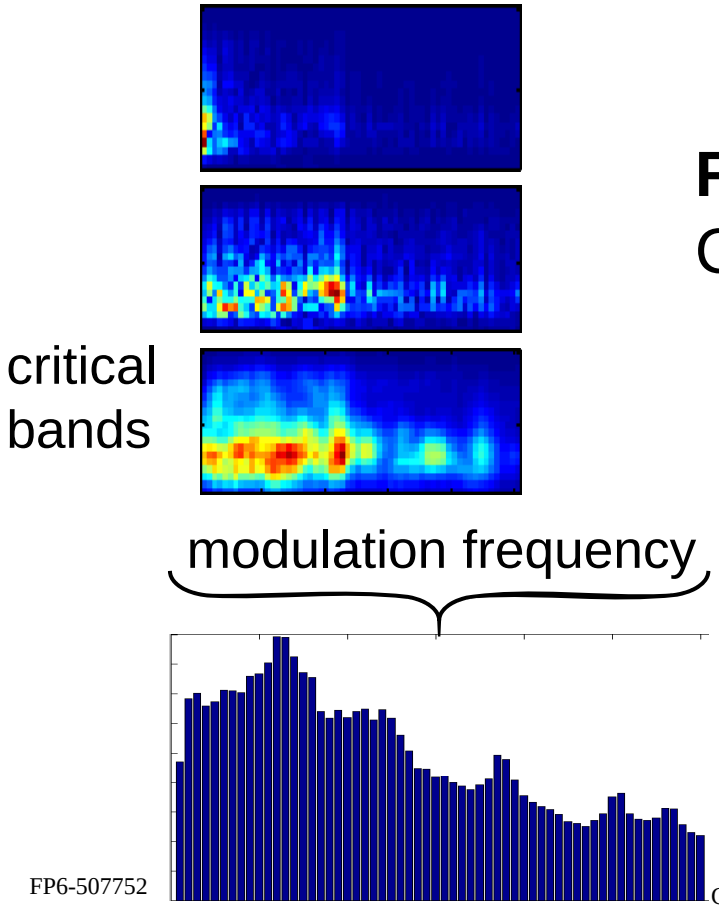
Median

24*60=

1440-dim feature vector

# Audio Features: RH

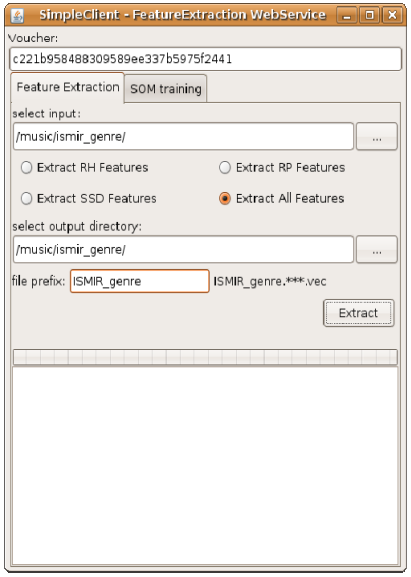
Rhythm Histograms (RH)



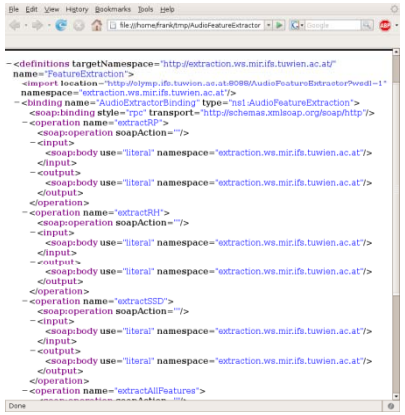

**RH**: 60 dimensions
Captures rhythmic events

# Web Service

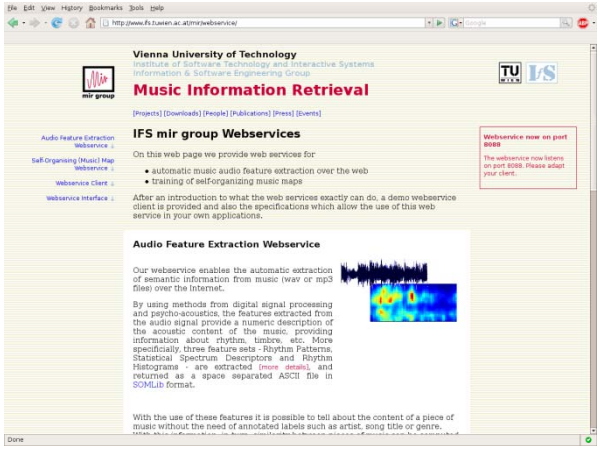
## Basic Concept




Client



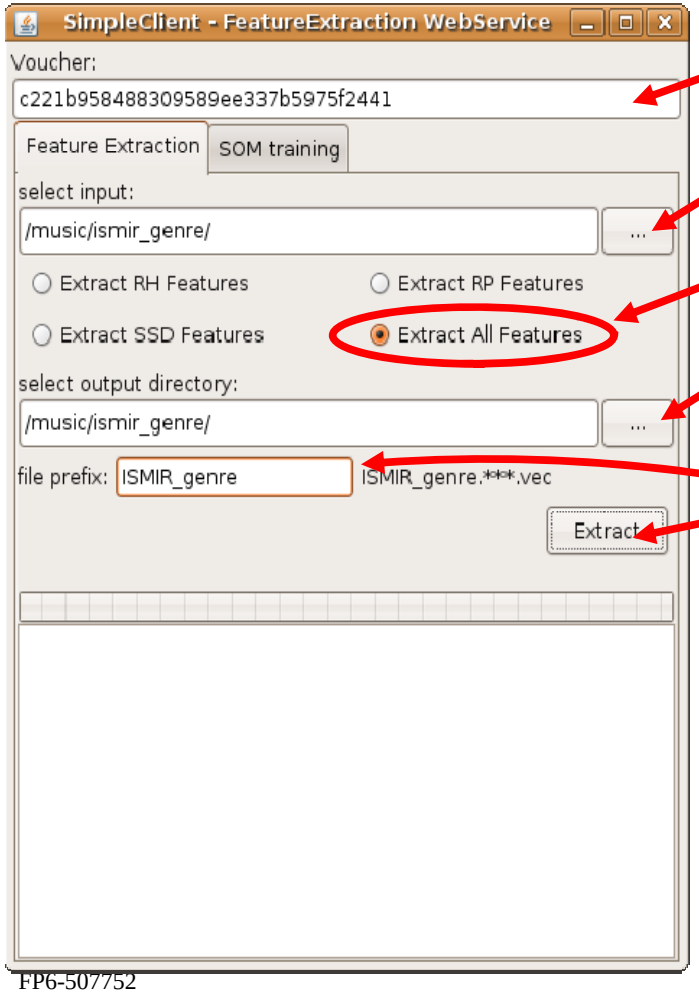
WSDL



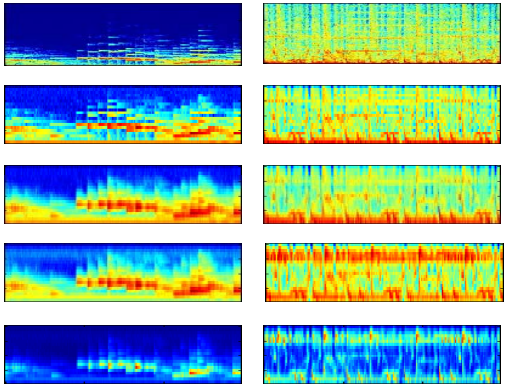

Web Service

# Web Service

## Audio Feature Extraction

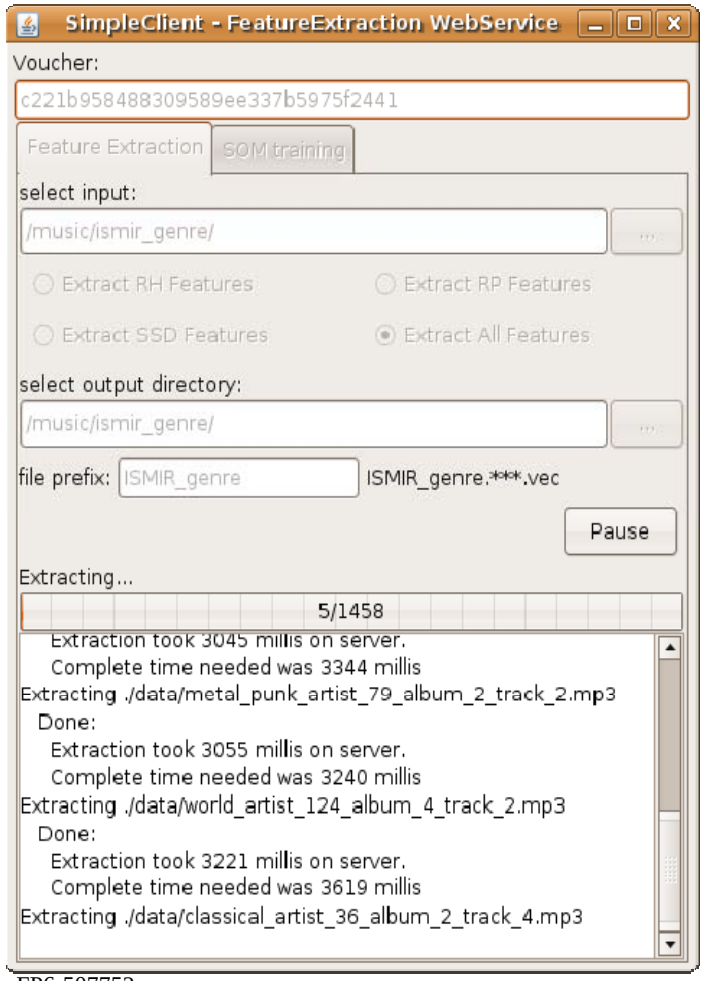



- voucher
- source directory
- “Extract All Features”
- output directory
- file prefix
- GO



# Web Service

## Audio Feature Extraction




- in progress...

## Outline

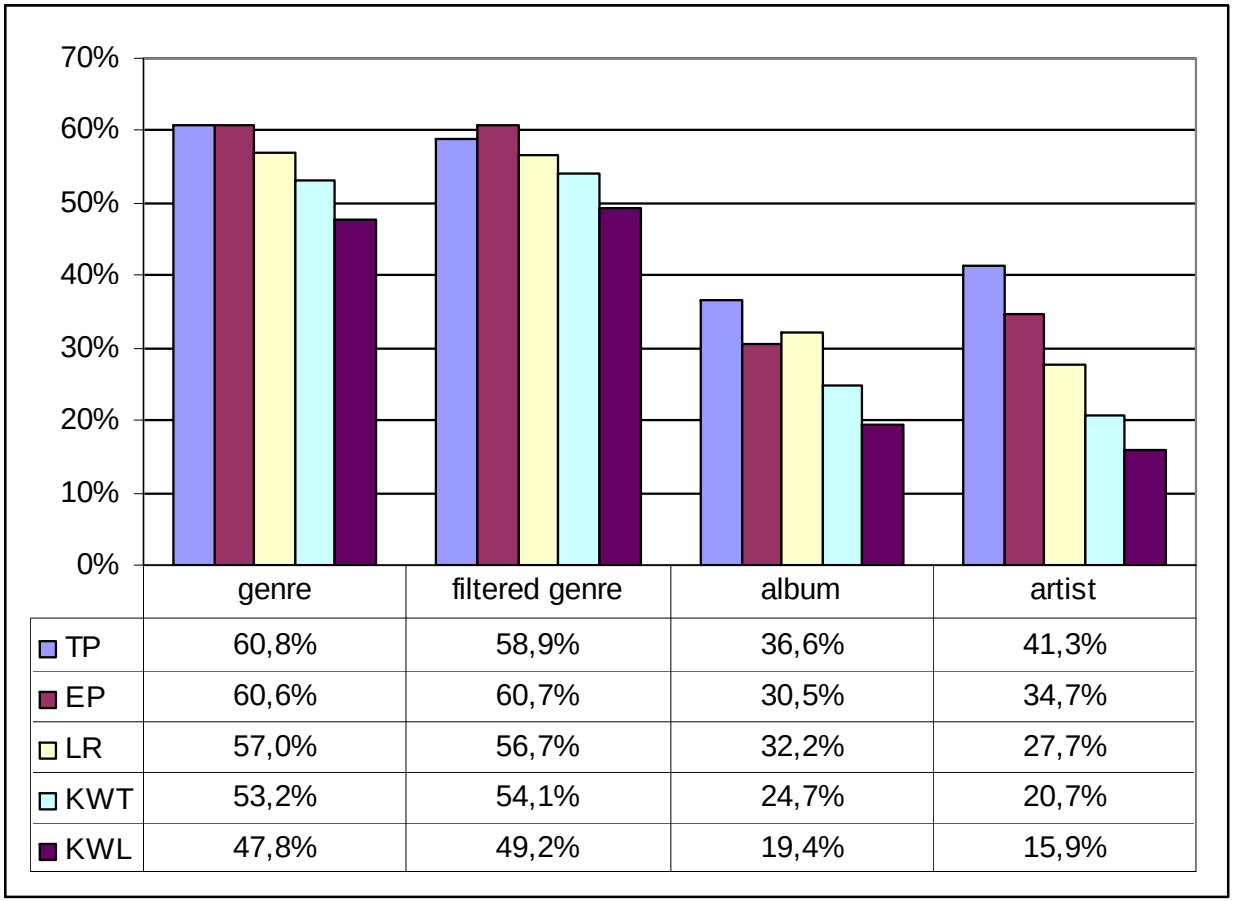
- Feature Extraction:
  - RP, RH, SSD
  - Web Services for Feature Extraction
- Applications
  - SOMeJB/PocketSOM: Browsing Music Collections
  - ARIA: Improving sound quality
- Conclusions

# Application

- Low-level features describe audio content
- Need to be aggregated to provide higher-level semantic information
  - genre classification
  - mood/emotional analysis
  - artist identification
- This, in turn, can be used for further applications, such as
  - Browsing applications for music collections
  - Parameter tuning for improving transients, countering the effect of overcompression

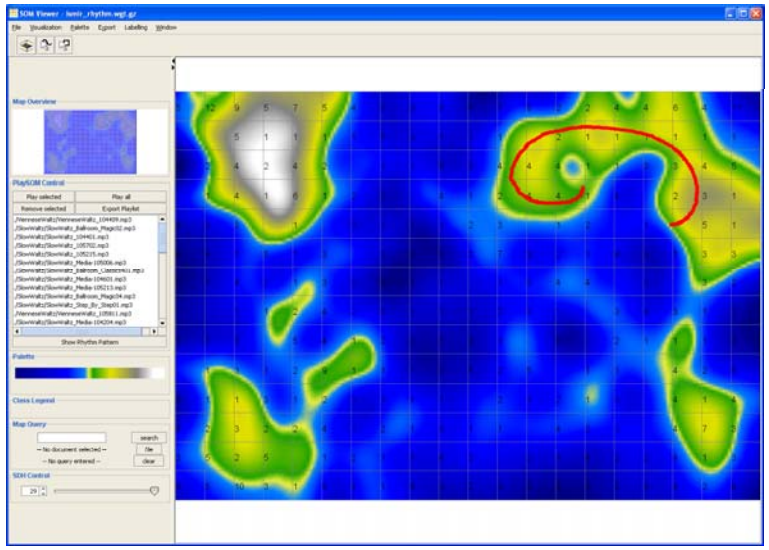
# Classification

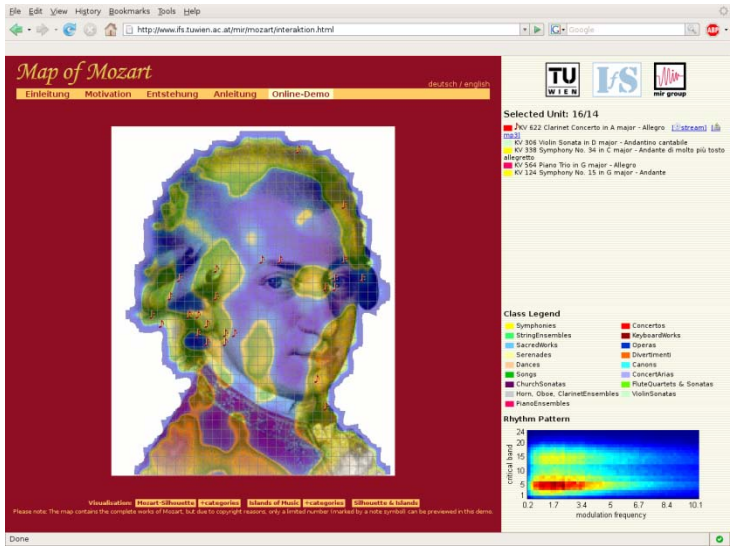
- Classifying Music into Genres
- MIREX 2006 Benchmark



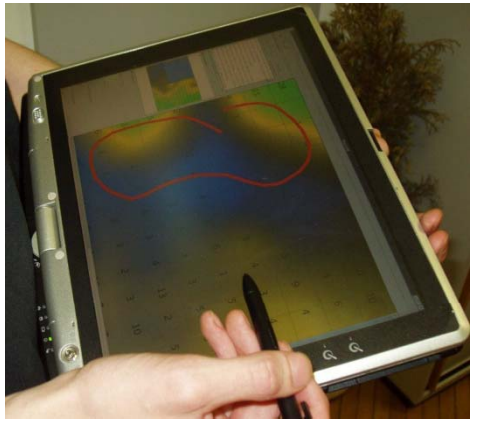

| | genre | filtered genre | album | artist |
|---|---|---|---|---|
| TP | 60,8% | 58,9% | 36,6% | 41,3% |
| EP | 60,6% | 60,7% | 30,5% | 34,7% |
| LR | 57,0% | 56,7% | 32,2% | 27,7% |
| KWT | 53,2% | 54,1% | 24,7% | 20,7% |
| KWL | 47,8% | 49,2% | 19,4% | 15,9% |

# Browsing Music Collections: PlaySOM






- Overview to large Music-Collections
- Based on the Self-Organising Map
- Music that sounds similar is located together
- “Genres” form Islands
- Intuitive Trajectory Selection
- PocketSOM

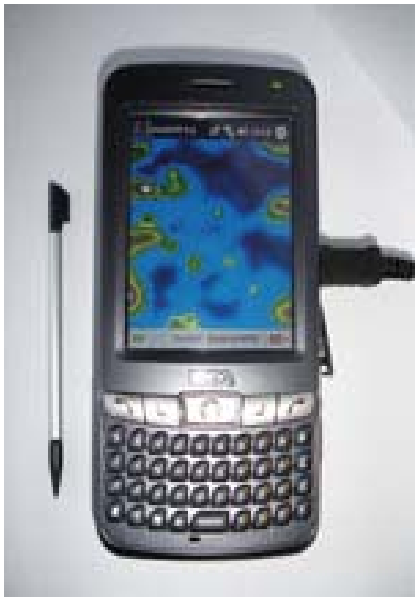


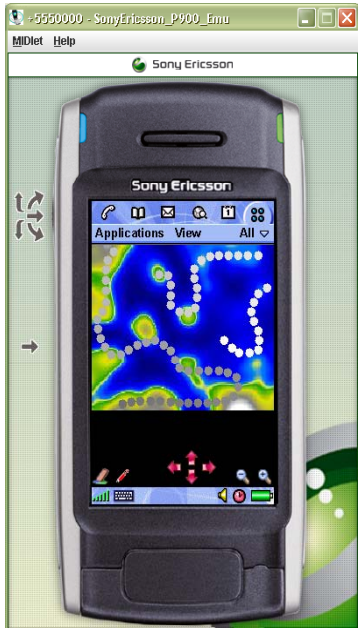
# Browsing Music Collections: PocketSOM

- Application for Mobile Devices
- Streaming Audio
- Remote Control

http://www.ifs.tuwien.ac.at/mir/pocketsom/



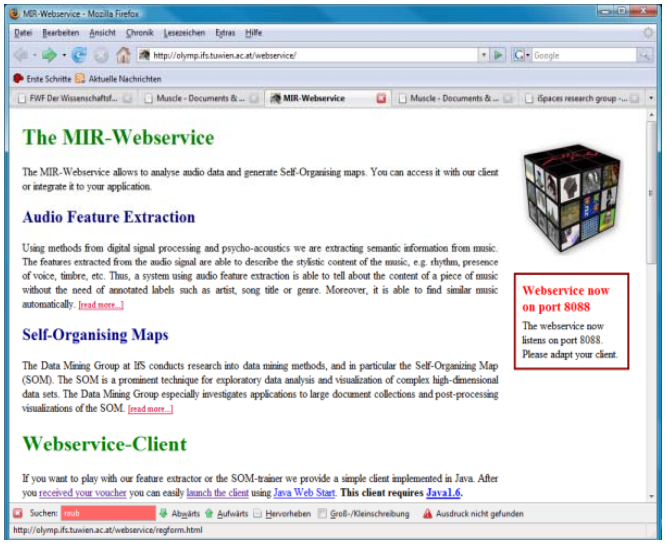





# Map Interfaces to Music

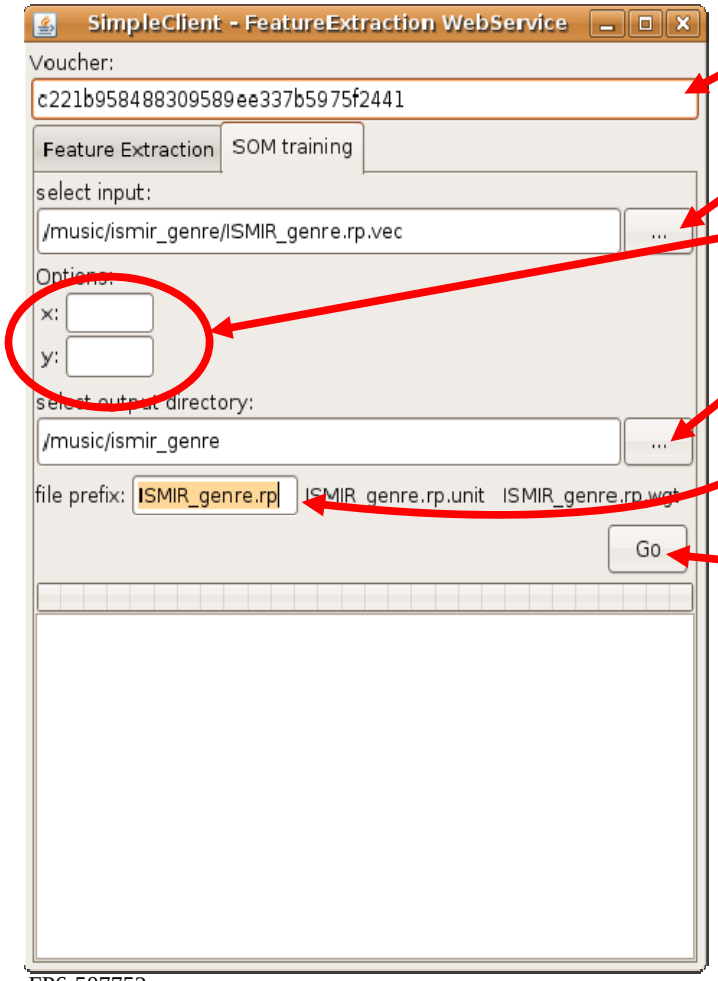
- Desktop Viewer
- 4 different viewer applications for different mobile devices
  - iPocketSOM: for iPAQ
  - ePocketSOM: for eweVM
  - mPocketSOM: for JavaME5
  - PocketSOM.NET: for Windows Mobile

  http://www.ifs.tuwien.ac.at/mir/pocketsom/
- Web Service for SOM Training
- To be presented at CeBIT 2008




# Web Service

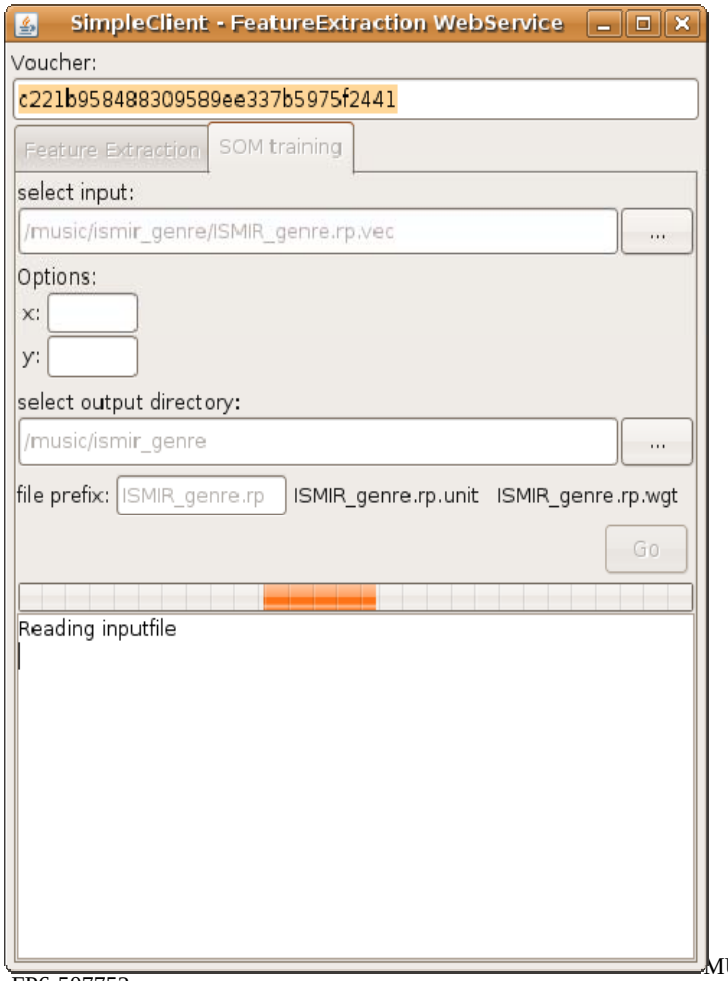
## Organizing Music – SOM Training




- voucher
- vector file
- map-dimension (optional)
- output directory
- file prefix
- GO

# Web Service

## Organizing Music – SOM Training




- in progress...

## Outline

- Feature Extraction:
  - RP, RH, SSD
  - Web Services for Feature Extraction
- Applications
  - SOMeJB/PocketSOM: Browsing Music Collections
  - ARIA: Improving sound quality
- Conclusions

# ARIA Algorithm: background

**Dynamic Range Compression** techniques has been adopted in order to improve the quality of the music recording technology i.e. tape, vinil etc.(for better S/N ratio etc.).

From ’80s compression has been used in FM radio broadcasting and other applications mainly to allow a better hearing in noise environments.

**An over-compression** is today used in the CD audio mastering: this reduces heavily the amount of audio transients, resulting in a loud but “flat” sound, so the music fidelity is strongly affected. Over the time this manner of listening the music can produce hearing loss.

**ARIA DDS** v.1* is a method to restore a “contrast” effect in the compressed music tracks, expanding transients like in live performances.

**ARIA** has been improved during the MUSCLE –NoE activity, in the framework of ET9-ET7 e-Team (toward ARIA v.2 ).

**(*www.aria99.com by M. Magrini e G. Biagiotti with the support of ISTI-CNR)**

# ARIA Algorithm: background

## Audio Dynamics

Dynamic range of a *musical track* **:**
estimate of ratio between **loudest** and **softest** (lowest) signal
*(depends on the recorded signal)*

Dynamic range of a *recording media*:
depends on the **kind of media** (and coding)
*(vinyl, tape, CD, DVD etc.)*

Compressor parameters:

**Threshold**: Compression begins above it

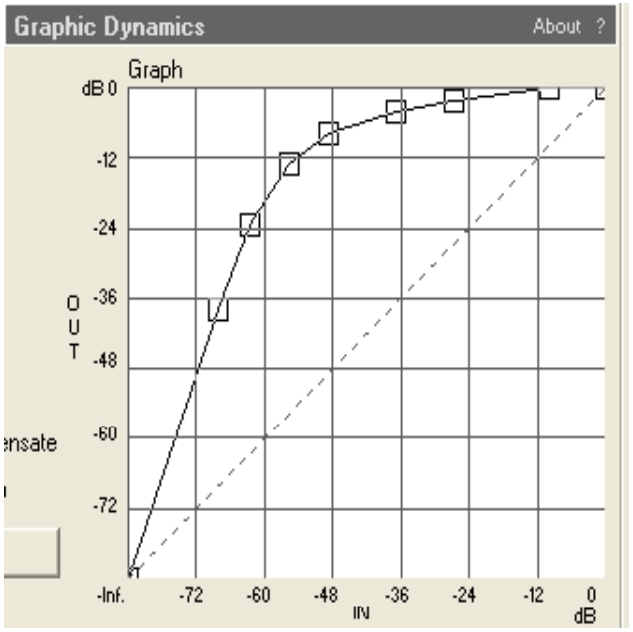
**Attack** - **release time**: how fast compression starts /stops

**Ratio**: How much the gain is reduced

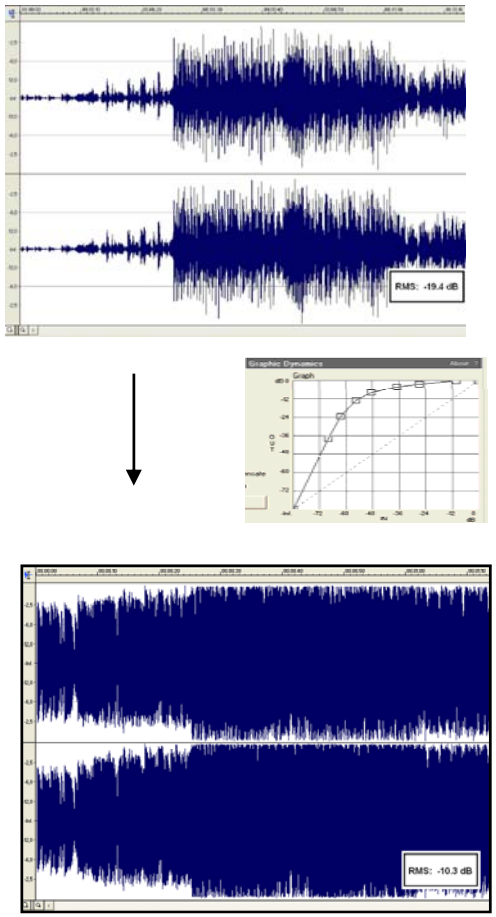
Other special compression functions:

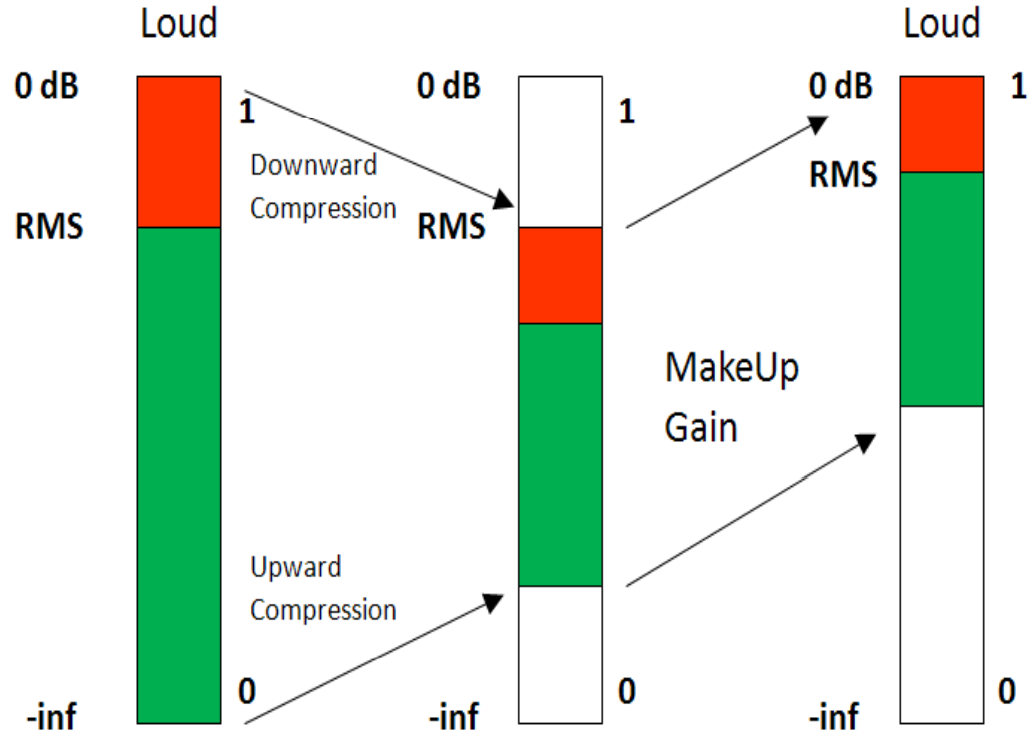
***Limiter****: fast attack , high compression ratio*

***More advanced limiters:*** *Look Ahead peak limiter, Maximizer*



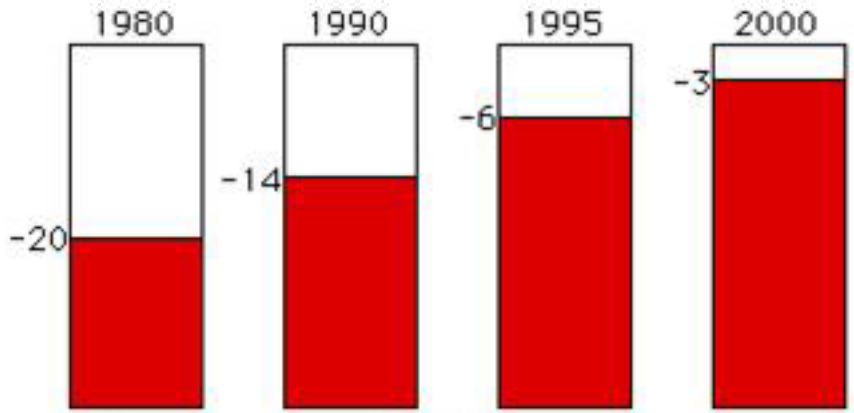
# ARIA Algorithm: background - dynamic range compression procedure



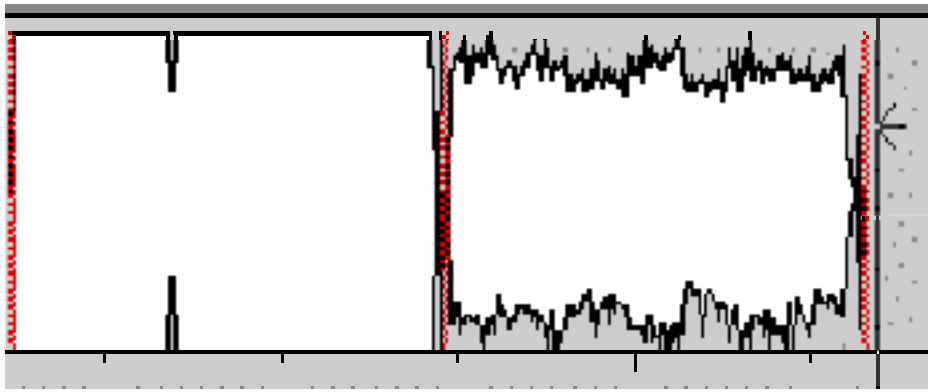

No information are given about the compression parameters used

# ARIA Algorithm: background - trend




Typical difference of perceived loudness in commercial recordings, 1980 → 2000

- red - signal power
- white – dynamic range



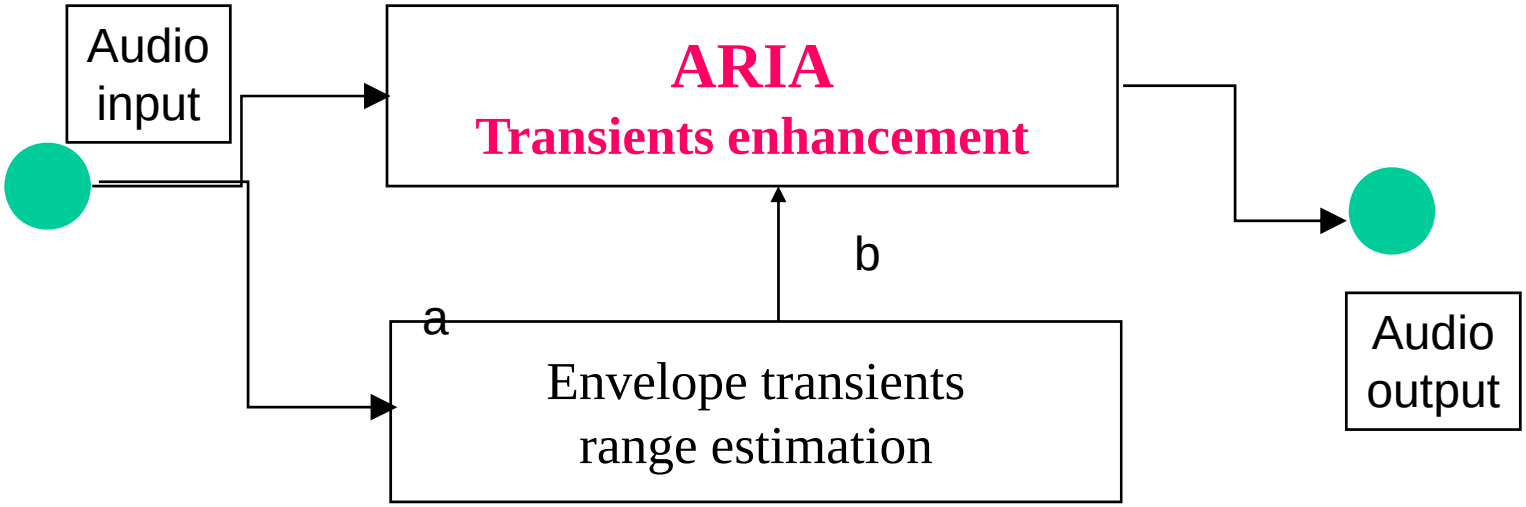
2000's pop song (Ricky Martin) vs. 1990 pop song(Mellencamp)

- time domain comparison

# ARIA Algorithm: functionality schema

a) incoming signal is analyzed in real time estimating the suitable features E

b) depending on this features a gain factor is computed to controls the transients, i.e the amplitude of output signals

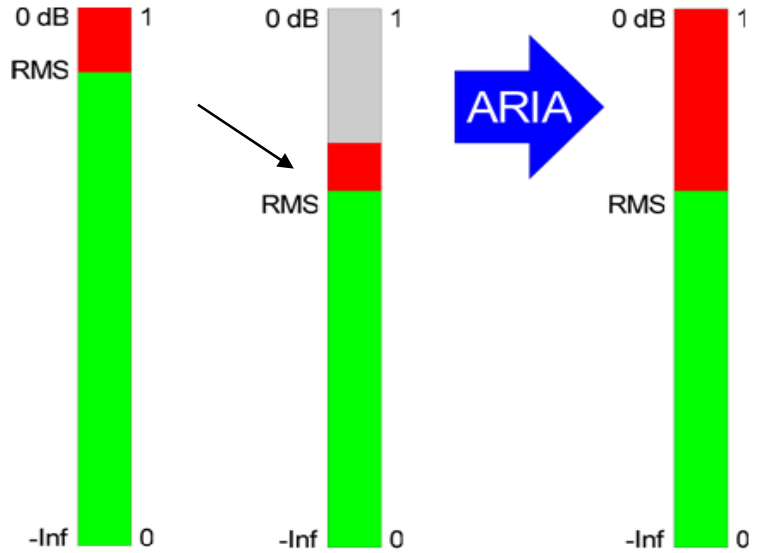



Gain factor can be modified depending on different music genres, compression levels etc.

# ARIA Algorithm: main functionality

ARIA performs the inverse of compression procedures:

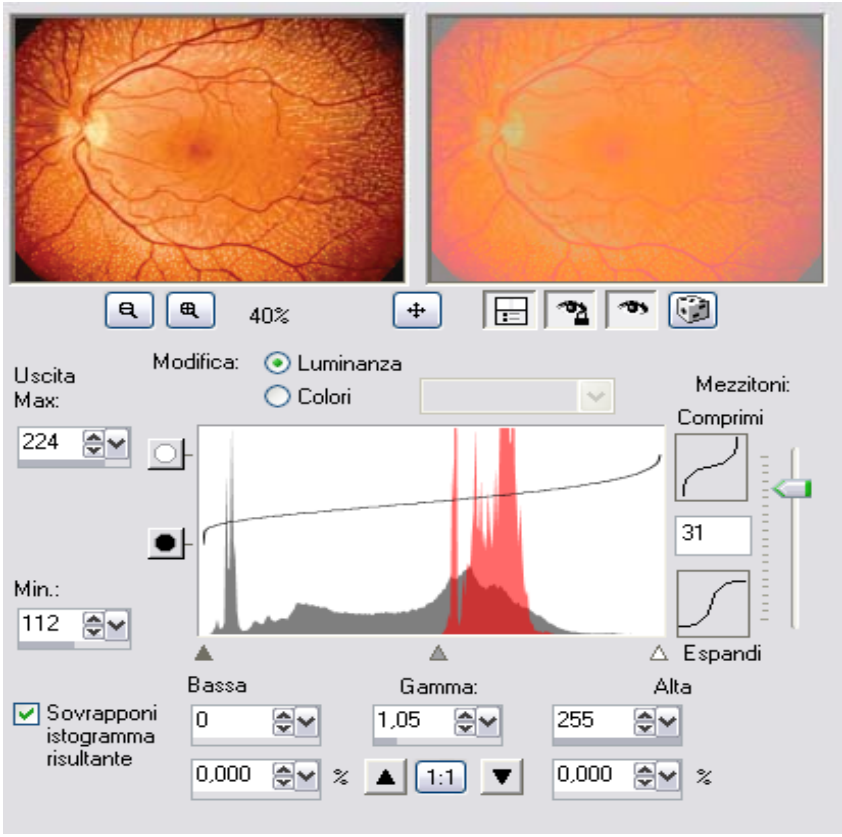
a) reducing the incoming signal of 6 dB in order to recovering a suitable headroom
b) operating the transients enhancement




compressed track

a) ARIA first computation

b) ARIA final track

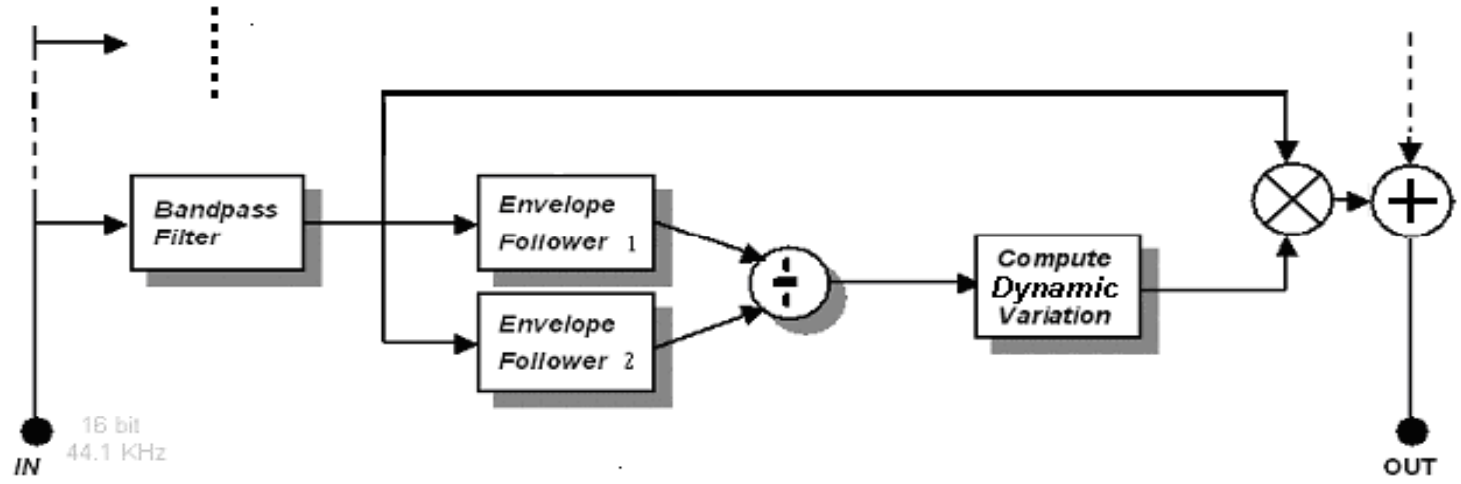


# ARIA Algorithm: basic steps of the algorithm

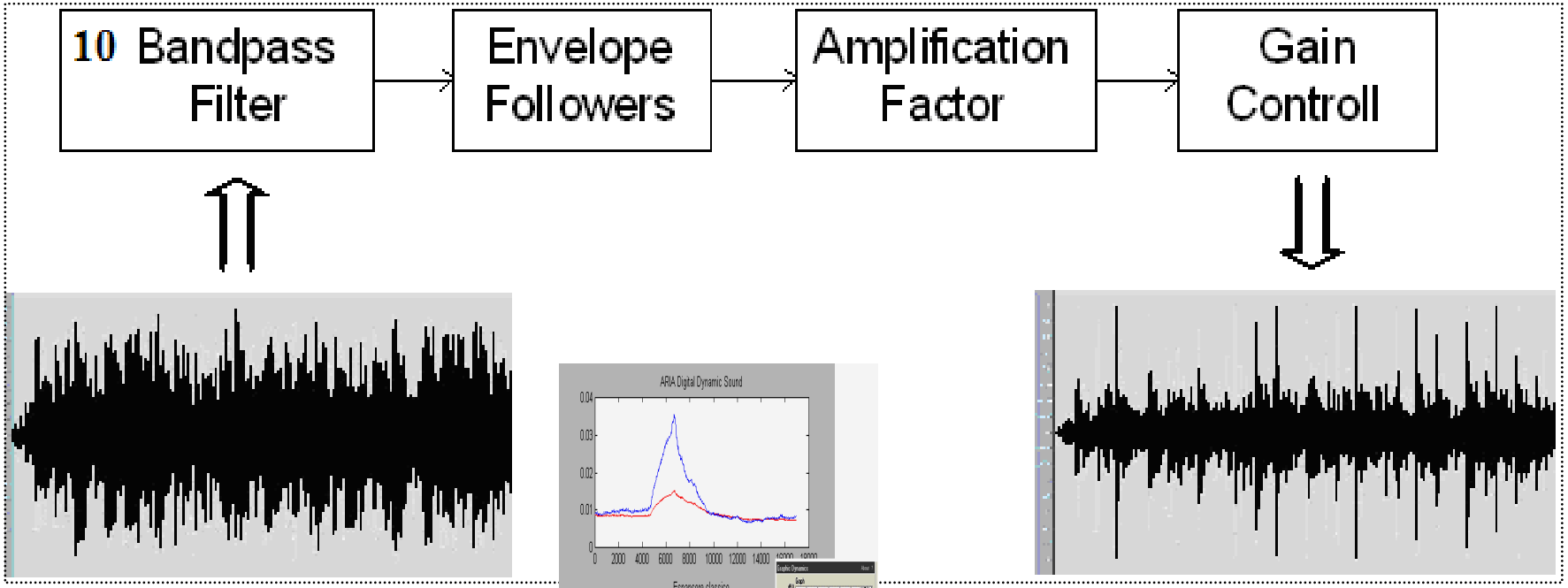
- Audio is sampled and splitted in n bands by a filter banks (n= 10)
- Long and short term amplitude envelopes are computed for each band
- A variable amplification factor, based on the ratio of the two envelopes

  is computed and then used to dynamically control the sub-band signal envelope (enhancement factor R[n])
- Singnals are then added to obtain the overall effect

E (enhancement control parameters is computed with low latency : 1-2 sec)



# ARIA Algorithm: basic steps of the algorithm




Music Signal with Compressed Dynamic Range

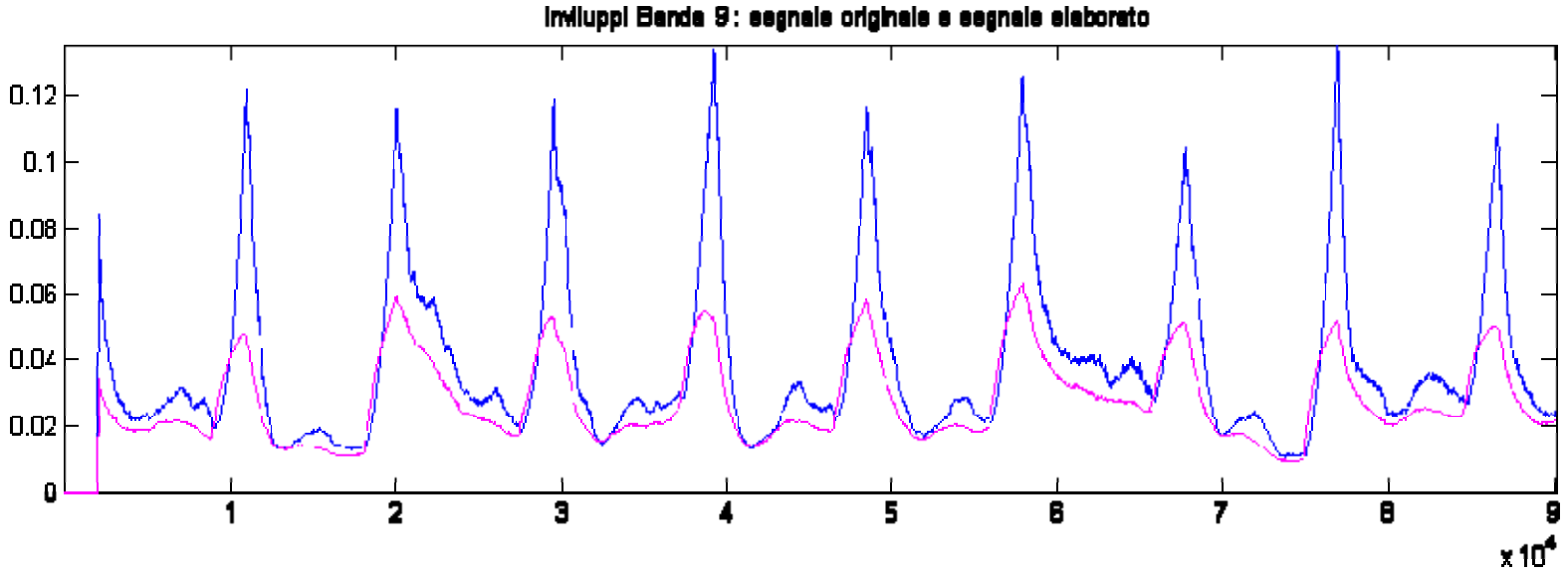
Enhanced Signal

(similar to the original one)

ARIA v.1, TM (2003)

# ARIA Algorithm: basic steps of the algorithm

- The time domain representation clearly shows that the transients of sub-band amplitude envelope (in blue) have been enhanced versus the input signal one's (in red):
- The resulting sound is much more vivid and "live".



# ARIA Algorithm: characteristics

Differences respect to dynamic expander:

- **ARIA slightest affects the overall timbre**:

*It acts only on fast (user defined) transitions.*

- **ARIA is independent from the overall volume**:

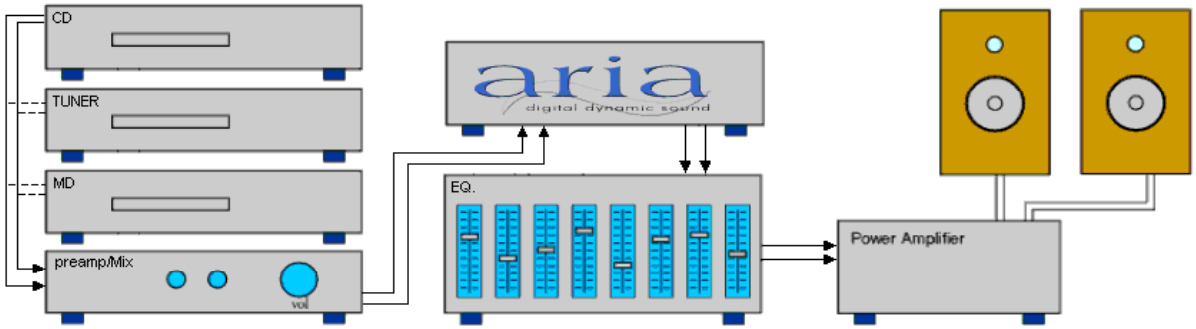
*classic expander really works only over some volume threshold*

Applications (to be focused):

- **Stand alone box** in the HI-FI chain (before the power amplifier)

  (ARIA v.1 TM)

- **Hard-coded** (embedded): in portable mp3 player, Car audio..
- **SW plug-in for audio player applications** (Winamp, iTunes etc.)



# ARIA v.1→ ARIA v.2: improvements in the MUSCLE framework

Aria v.1™ had fixed settings, often no optimal. The upgrade is addressed to modify some characteristics:

- The Enhancement factor computation has been modified so to gradually reach the maximum value in the envelope gain control.

- A user-variable level control (ranging from 1 to 6) has been introduced, for setting different effect amounts, in order to obtain the optimal level for each musical track (obj-subj). It basically works as a multiplication constant for the variable index k * R[n] .

- A sw plugin for Winamp has been developed.

- For each sub-band the enhancement factor can also be multiplied by a specific constant considering the Fletcher and Munson iso-phonic curves.

# ARIA v.2: plugin interface

An ARIA v.2 version is implemented as a plugin for the popular Winamp media player.


The plugin interface is provided with an activation switch and an intensity control slider with six level.

# ARIA v.2: pre-setting

ARIA can be set from 1 to 6, representing the dB level of the transients enhancement.

The level can be set knowing a special feature we have defined as the **Dynamic Index** (Di) of the track.

*This feature may be easily added to the feature set used by TU-WIEN in the PlaySom System:*

- so PlaySom may automatically set the the optimal setting for an archived track

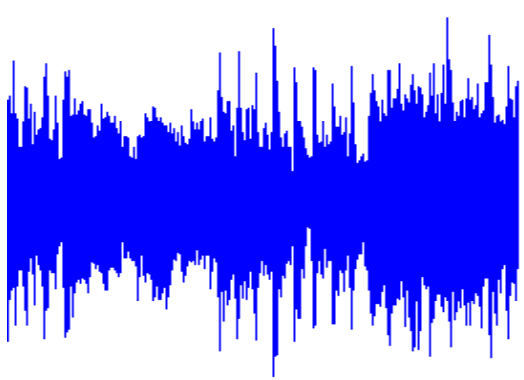
- the PlaySom genre knowledge may be useful for enabling/disabling ARIA according the genre (e.g. ARIA is not necessary for classical music)

# ARIA Algorithm: Dynamic index

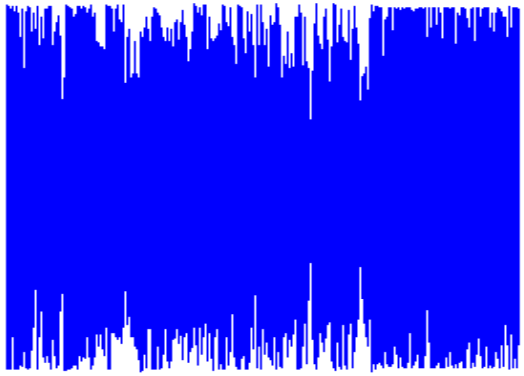
**Dynamic range** of a musical signal is defined as the difference (in dB) of max versus lower RMS signal's level, but dealing with non stationary signals no standard rule is known for its computation.

The method we developed is based on $RMS_{max}$ / $RMS_{average}$ computed in time windows taking into account psycho-acoustics and averaging it on the whole track: it is summarized in one feature, the **Dynamic Index (Di),** a sort of *crest factor*, which represents the energetic transients range of a musical track.

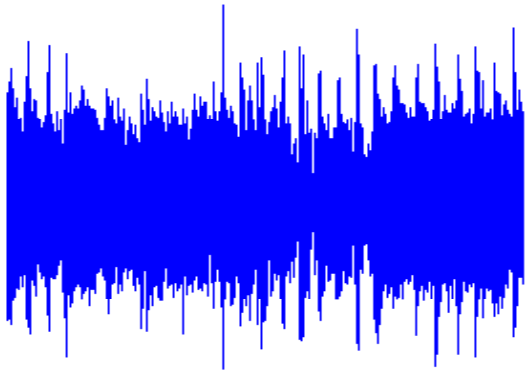
Example of Di (track n° 6 referred into the table following)



Original track 'Di'=2,02



Compressed track 'Di'



enhanced track with ARIA 'Di'=2,01

# ARIA Algorithm: Dynamic index

Using a set of tracks available in two forms **before** and **after** an usual mastering compression*:

•ARIA has been applied to the compressed track, and set in a way that the Di of the processed signal approaches the Di of the uncompressed musical track

•The level found in this way matched or is close to the subjective optimal level

| Track | original 'Di' | compress 'Di' | decompress 'Di' | optimal aria level |
|---|---|---|---|---|
| Track1 | 2,072 | 1,76 | 2,11 | 2 |
| Track2 | 2,35 | 2,04 | 2,39 | 2 |
| Track3 | 2,67 | 2,19 | 2,61 | 1 |
| Track4 | 1,94 | 1,81 | 2,011 | 2 |
| Track5 | 2,26 | 1,99 | 2,017 | 2 |
| Track6 | 2,02 | 1,53 | 2,01 | 6 |
| Track7 | 1,87 | 1,71 | 1,86 | 2 |
| Track8 | 2,205 | 1,87 | 2,27 | 2 |
| Track9 | 2,1302 | 1,857 | 2,1095 | 2 |

* with the support of Studio Lab di Sergio Taglioni (Cascina, Pisa)

# ARIA Algorithm: Dynamic index table

Using this feature we have made a table of ARIA settings vs Di of the incoming track:
lower value → high compression; higher value → low or no compression

| ARIA levels chart conversion for compress tracks. Based on 'Di' coefficient | | |
|---|---|---|
| **from** | **to** | **ARIA recommended level** |
| 2,05 | > 2,05 | **ARIA 1** |
| 1,7 | 2,05 | **ARIA 2** |
| 1,65 | 1,7 | **ARIA 3** |
| 1,6 | 1,65 | **ARIA 4** |
| 1,55 | 1,6 | **ARIA 5** |
| < 1,55 | 1,55 | **ARIA 6** |

The table shows the ARIA level versus the dynamic coefficient.
Example: a compressed audio track that has a Di of 1.9 would require an ARIA level 2.

# ARIA Algorithm: objective preset

We tested this rule on a set of tracks selected from the ISMIR database, with some session of subjective tests. These tests made with it reported that, in general, the listener reports that the listening pleasure of the musical track has been improved: tracks are more dynamically contrasted without artefact

| ISMIR samples tracks and genre | compress 'Di' | 'Di' at each level of 'ARIA' | | | | | | optimal aria level |
|---|---|---|---|---|---|---|---|---|
| | | a1 | a2 | a3 | a4 | a5 | a6 | |
| **ELETRONIC** | | | | | | | | |
| 1-believe | 1,7572 | 1,8648 | 2,0259 | 2,1529 | 2,2518 | 2,329 | 2,3909 | 2 |
| 10-brilliant_day | 1,6198 | 1,6556 | 1,7277 | 1,8266 | 1,8195 | 1,8422 | 1,8599 | 4 |
| 11-first_melstorm | 1,9553 | 2,123 | 2,2223 | 2,3157 | 2,3901 | 2,4505 | 2,4995 | 2 |
| **JAZZ&BLUES** | | | | | | | | |
| 1-like_white_on_ric | 1,9788 | 2,0433 | 2,2691 | 2,4433 | 2,5758 | 2,679 | 2,7621 | 2 |
| 10-abbey_rhodes | 2,5669 | 2,889 | 3,368 | 3,7069 | 3,952 | 4,1363 | 4,281 | 1 |
| 11-jag's_rag | 2,0091 | 2,2417 | 2,4845 | 2,6575 | 2,7833 | 2,8763 | 2,9447 | 2 |
| **METAL&PUNK** | | | | | | | | |
| 1-by_mourning | 1,8468 | 1,973 | 2,1088 | 2,2126 | 2,2934 | 2,357 | 2,4083 | 2 |
| 11-donkey_punch | 1,6009 | 1,7441 | 1,8832 | 1,9923 | 2,0781 | 2,1458 | 2,2001 | 4 |
| 2-angel's_blood | 1,5031 | 1,6282 | 1,7063 | 1,7677 | 1,8164 | 1,8558 | 1,8875 | 6 |
| **ROCK&POP** | | | | | | | | |
| 6-coda | 1,5142 | 1,8011 | 1,9159 | 2,0056 | 2,0748 | 2,1303 | 2,1756 | 6 |
| 10-at_least_you've_be | 1,6351 | 1,8314 | 1,8878 | 1,9377 | 1,9782 | 2,0111 | 2,0392 | 4 |
| 12-a_little_part_of_m | 2,1339 | 2,3191 | 2,5882 | 2,7835 | 2,9293 | 3,0415 | 3,1314 | 1 |
| **WORLD** | | | | | | | | |
| 14-milk_shake | 1,7166 | 1,7942 | 1,8942 | 1,9677 | 2,0244 | 2,0703 | 2,1077 | 2 |
| 3-arizona | 1,7799 | 1,8609 | 2,034 | 2,1924 | 2,3029 | 2,3507 | 2,4005 | 2 |
| 4-beledi_2_4_4 | 2,8856 | 3,2594 | 3,9309 | 4,3853 | 4,7166 | 4,9604 | 5,1359 | 1 |
| **For CLASSICAL tracks the use of 'ARIA' isn't recomended** | | | | | | | | |

**This is only a part of about one houndred tracks tested from us**

# ARIA Algorithm: subj. vs obj. tests

| ISMIR Collection Tracks | Di | | ARIA effect level <->Di | | | | | Obj. | Subj. |
|---|---|---|---|---|---|---|---|---|---|
| n. tracks and titles | compr. | a1 | a2 | a3 | a4 | a5 | a6 | tab | i |
| 15.electronic_1-believe.wav | 1,7572 | 1,865 | 2,026 | 2,153 | 2,252 | 2,329 | 2,391 | 2 | 2 |
| 16.electronic_10-brilliant_day.wav | 1,6198 | 1,656 | 1,728 | 1,827 | 1,82 | 1,8422 | 1,86 | 4 | 4 |
| 17.electronic_11-first_melstorm.wav | 1,9553 | 2,123 | 2,222 | 2,316 | 2,39 | 2,4505 | 2,5 | 2 | 6 |
| 18.electronic_12-never_leave.wav | 1,5001 | 1,537 | 1,554 | 1,57 | 1,584 | 1,5969 | 1,609 | 6 | 1- 0 |
| 19.electronic_14-ones.wav | 2,357 | 2,633 | 3,132 | 3,535 | 3,86 | 4,1102 | 4,281 | 1 | 1 |
| 20.electronic_15-chord2.wav | 1,7696 | 1,791 | 1,827 | 1,861 | 1,889 | 1,9107 | 1,929 | 2 | 2 |
| 21electronic_16-ambicious.wav | 1,8661 | 2,12 | 2,21 | 2,283 | 2,34 | 2,3844 | 2,42 | 2 | 2 |
| 22.electronic_2-blue_days.wav | 2,6976 | 3,162 | 3,663 | 4,009 | 4,256 | 4,4377 | 4,573 | 1 | 1- 0 |
| 23.electronic_3-d_b_l.wav | 1,5633 | 1,717 | 1,796 | 1,861 | 1,913 | 1,9534 | 1,988 | 6 | 5 |
| 24.electronic_4-black_mystery.wav | 1,7427 | 1,975 | 2,076 | 2,157 | 2,219 | 2,2682 | 2,307 | 2 | 2 |
| 25.electronic_5-130.wav | 1,9441 | 2,034 | 2,196 | 2,333 | 2,436 | 2,519 | 2,585 | 2 | 2 - 3 |
| 26.electronic_6-angels.wav | 1,9861 | 2,227 | 2,409 | 2,544 | 2,648 | 2,7287 | 2,793 | 2 | 2 - 1 |
| 27.electronic_7-call_me_snake.wav | 2,1765 | 2,205 | 2,411 | 2,563 | 2,677 | 2,7664 | 2,837 | 1 | 1 |
| 28.electronic_8-ambriel.wav | 2,1617 | 2,406 | 2,669 | 2,858 | 2,995 | 3,0975 | 3,177 | 1 | 1 |
| 29.electronic_9-give_me_some_ | 1,6367 | 1,804 | 1,891 | 1,956 | 2,01 | 2,0515 | 2,085 | 4 | 3 - 4 |
| 30.jazz_blues_1-like_white_on_ric.wav | 1,9788 | 2,043 | 2,269 | 2,443 | 2,576 | 2,679 | 2,762 | 2 | 1- 2 |
| 31jazz_blues_10-abbey_rhodes.wav | 2,5669 | 2,889 | 3,368 | 3,707 | 3,952 | 4,1363 | 4,281 | 1 | off - 1 |

# ARIA Algorithm: subj. tests form

Name_____
Age____
Date___

Test 1 original song (o), compressed (c) , effetted by ARIA (A)

| tracks | close vs original | Preferred |
|---|---|---|
| | c or A | o, c or A |
| | | |

Preliminary Results

- Experts: ~ 95% prefer ARIA, 5% not
- Non-Experts: ~ 80% distinguish A vs. c; 10% don't distinguish; 10% prefer c

Test 2 song retrived from ISMIR collection (k)

| ISMIR tracks | More dynamic | Preferred |
|---|---|---|
| | k or A | k or A |
| | | |

Preliminary Results

- Experts: 100% prefer ARIA.
- Non-Experts: ~ 80% distinguish A vs. k; 10% don't distinguish; 10% prefer k

# ARIA Algorithm: Dynamic index problems

**Drawbacks:**

• Dynamic index cannot be easily computed “on the fly” (requires some seconds for a precise computation)

• Musical genres originally without heavy transients (some kind of classical, ethnical music) show a low Di but they do not need to be ARIA processed

**ARIA processor Integration in the TU-Wien PlaySom system:**

• If Di is added in the system feature set PlaySom could automatically set the the optimal ARIA setting playing an archived track, without any delay.

• the PlaySom genre knowledge may be useful for enabling/disabling ARIA according the genre (e.g. ARIA is not necessary for classical music)

**Works in progress:**

•Extensive obj-subj evaluation (using ISMIR and other tracks)

•ARIA v.2 to ARIA v.3 (better sub-band filter design etc.)

•Commercial applications proposal: studio recording, audio processingetc

# Conclusion

- Range of features to be extracted from audio
- Web Service available for extraction
- Classification to obtain higher-level semantic information
- Applications in different domains
  - Browsing music collections
  - Improving sound quality

http://www.ifs.tuwien.ac.at/mir

http://www.isti.cnr.it/

http://www.aria99.com