Tony: a tool for melody transcription - Feature #926
Plot normalized waveform

2014-04-09 03:48 AM - Rachel Bittner

Status:	Closed	**Start date:**	2014-04-09
Priority:	Urgent	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The view of the waveform has proven to be very useful - in particular for onsets/offsets.
However, it doesn't appear to be normalized because in quiet tracks you don't see much.

History

#1 - 2014-05-20 02:02 PM - Matthias Mauch

- Priority changed from Normal to High

This does not just apply to the waveform visualisation, but also to spectrogram visualisation and the actual PYIN processing, so I think normalisation is actually an important issue. Just discussing this with Simon: we would need that for our project.

So I set to high priority.

#2 - 2014-05-21 12:23 PM - Matthias Mauch

So I've been looking around a little bit to get a feel how the normalisation should work.

Conceptually I think it's always going to look like this (sorry if this is obvious):

1. measure input gain
2. amplify so that new gain = x (where x is our chosen level)

Re 1: I think we should follow the Replay Gain guys, who calculate the gain of the original signal as the 95%ile of framewise (50ms) loudness measurements. This is apparently grounded in perception theory: http://wiki.hydrogenaudio.org/index.php?title=ReplayGain_specification
For simplicity we could just use the RMS on frames as our loudness measurement, so by taking the 95%ile of that we have our input gain.

Re 2: We need to decide how conservative we want to be on the output gain. If we do normalisation internally in Tony, then it's not a big issue, and we can adjust the floating point representation to look and sound right, e.g x = -10 dB, and abs values > 1 are fine. If we want to write a new file, which clips at abs values > 1, then we might need to be more conservative. x = -20dB seems to be what Brecht suggested, and in a few informal experiments on our singing data no clipping happened in that case.

#3 - 2014-06-03 05:14 PM - Matthias Mauch

- Priority changed from High to Urgent

#4 - 2014-06-13 02:27 PM - Matthias Mauch

A quick fix could be to simply normalise to maximum level, i.e. in Matlabby code:

```
x = x/max(abs(x));
```

That would, in many cases, be enough, and would introduce no clipping. (It might introduce some quantisation error, but I assume that's negligible.)

#5 - 2014-06-13 04:35 PM - Chris Cannam

OK, as of commit:ed9296a27a27 we normalise to max level == 1. Try it out.

#6 - 2014-06-13 04:35 PM - Chris Cannam

- Status changed from New to Resolved

#7 - 2014-06-13 05:22 PM - Matthias Mauch

- Status changed from Resolved to Closed

very nice. works.