Real Time Tempo Analysis of Drum Beats

Author: **Philip Hannant**, Supervisor: **Professor Steve Maybank**

Birkbeck, University of London
Department of Computer Science and Information Systems

Project Report
MSc Computer Science

September, 2016

Contents

Abbreviations

BPM	Beats Per Minute
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
JSON	Javacript Object Notification
IDE	Integrated Development Environment
TDD	Test Driven Development

Definitions

Acoustic Drum Kit	A collection of drums and cymbals which do not have electronic amplification. Typically made up of a bass drum, snare drum, tom-toms, hi-hat and 1 or more cymbals.
Beat	For the purpose of this project a beat will be defined as the sequence of equally spaced pulses used to calculate the tempo being played by the drummer.
Downbeat	Refers to beat one of a measure of music, called a downbeat to correspond to the motion a conductor's arm [1].
Drum Module	The device which serves as a central processing unit for an electronic drum kit, responsible for producing the sounds of the drum kit.
Electronic Drum Kit	An electrical device which is played like an acoustic drum kit, producing sounds from a stored library of instruments and samples.
MIDI	Musical Instrument Digital Interface is a protocol developed in the 1980's to allow electronic instruments and other digital musical tools to communicate with each other [3].
Tempo	The speed at which a piece of music is played [3] and counted in beats per minute (bpm).

1 Introduction and Background

This project report presents my aim to develop a real-time drum beat tempo analysis system using different beat detection algorithms which is able to record the perfomance of each method concurrently when an extensive set of drum samples, representing a real drummer's perfomance, is processed through the system.

1.1 Drumming Training Tools Background

Timing is the fundamental skill any good drummer should possess and is the staple by which they will be judged. For many years the only training tool available to a drummer to improve their timing was the metronome. An instrument used to mark musical tempo, erroenously attributed to Johann Nepomuk Maelzel in 1815 but was actually invented by a Dutchman, Dietrich Nikolaus Winkel a year earlier. The traditional metronome, based on Winkel's original design is a hand-wound clockwork instrument that uses a pendulum swung on a pivot to generate the ticking which depicts the desired tempo [1] is still used today by musicians, as seen in Figure 1.

Figure 1: Traditional Metronome

For drummers however the electronic versions of the metronome are much more widely used, to the point that metronomes are now developed with functionality specifically tailored to a drummers training requirements. The Tama Rhytham Watch (Figure 2) was the first metronome designed specifically for drummers, providing enough volume to be used with real drums as well as allowing for the use of different time signatures and preset set rhythm patterns to help improve perfomance.

Figure 2: Tama Rhythm Watch

Following the development of MIDI driven electronic drum kit came the development of more advanced training tools that now were able to provide live feedback to drummer during any given perfomance. Today the leaders in this field are Roland, their v-drums line provide a variety of tutition packages including the SCOPE and more recently the COACH system provided in the v-drum modules, the v-drum Rhymthm Coach line is an advanced version of the traditional drummers practice pad and the extensive DT-1 V-Drums tutor software package. Roland have even now gamified this field with their latest release, the V-Drums Friend Jam app. The application itself provides the player with live feedback and evaluates each performance in order to provide the player with a score which they can share over social media.

The aim of this project is to investigate whether some of the current beat detection algorithms available would be accurate enough to provide the basis for a training tool for dummers using an acoustic drum kit as opposed to an electronic drum kit.

1.2 Drum Musical Theory

In order to understand the fundamentals of musical timing some theory needs to be examined but beforehand the concept of a drummer playing time must be considered. Time, in a drumming sense is an informal term used to describe the consistent rhythmic pattern that a drummer will play on the hi-hat or ride cymbal [2] and it can be considered one of the most important components of any drum beat.

1.2.1 Notation

Drum music notation is written on staff that is made up of five individual lines, the clef is found on the far left of the staff which indicates the pitch of the notes [3] and as percussion instruments are non-pitched they use the percussion-clef. On traditional musical notation the lines and spaces between represent a tonal where as for drum notation, notes written on lines or spaces indicate a certain drum or cymbal. The staff is seperated into individual measures which are known as bars [4] and it is these bars that are the basis of musical time. For the purpose of this project it is the count of these beats that will be used to calculate the tempo of a certain drum beat.

1.2.2 Time Signatures

Time signatures appear on the staff just after the clef and are written as a fraction where the top number indicates the number of beats that there are in a bar. With the bottom number representing the size of the note that makes up the duration of one beat. For example the straight time four four (4/4) or common time signature indicates four beats in each bar or measure where each beat is made up of one quarter note [4]. Within these bar lines beats can be further divided by using a technique known as subdivision, which is a method for reducing the pulse or rhythm pattern into smaller parts than those originally written, for example counting a four four (4/4) measure in eighth (1/8) or sixteeth (1/16) notes.

1.2.3 Notes

The notes used to represent the percussive instrument to be played also provide the duration it should be played for. Notes come in different lengths and the key values are the whole (1/1), half (1/2), quarter (1/4), eighth (1/8) and sixteeth (1/16). For example two eighth notes represent the same time value as a single quarter note. It is possible to divide a note values by three instead of two, these notes are known as triplets. An eighth note triplet is played fifty percent faster than a normal eighth note, therefore for every two eighth notes there will be three eighth note triplets [4]. An example of the eighth-note triplets being used is in a twelve eight (12/8) jazz shuffle, the time element played on the ride cymbal or hi-hat is charaterised by playing the first and third triplet of an eighth-note triplet grouping [2].

1.2.4 Playing Basics

With the basics of drum theory covered it is now possible to discuss the key elements of a drum beat, typically for a straight four four (4/4) bar the bass drum will be played on the first and third beat and the snare drum will be played on the second and forth beat both as quarter notes. This is more commonly known as a back beat [2]. This just leaves the time element which will usually be played on the ride cymbal or hi-hat, this too could be played using quarter notes on the first, second, third and forth beats. However, in order to make the drum pattern more dynamic the time element will usually be played using subdivisions, typically using eighth note subdivisions. The ride cymbal or hi-hat will therefore be played on the first, second, third and forth beats as well as the eighth notes inbetween each quarter note. This can be demostrasted by counting the one-and-two-and-three-and-four-and, where the and represents the subdivided eighth note. Additonally to this technique a drummer will usually ensure that there is difference in the volume of the eighth notes being played on the quarter notes and those being played on the and. This technique of emphasising certain beats is known as accenting.

1.3 Beat Detection Background

Most of the early work on beat detection was a by-product of research directed at other areas of musicical understanding. The earliest work in this field can be attributed to H. C. Longuet-Higgins, who in 1976 while researching the psychological theory of how Western musicians perceive rhythmic and tonal relationships between notes. Produced an algorithm that was able to follow the beat of a performance and adjust the perceived tempo accordingly based on whether a note started earlier or later than expected [5]. Longuet-Higgins' work was built on the premise that in order to percieve the rhythmic structure of a melody it is first necessary to identify the time at which each beat

occurs [6], otherwise know as onset detection. The onset of a note is the instant which marks start of the variation in the frequency of a signal, a visualisation of this can be seen in Figure 3. Once detected it can then be used to measure the onset times of sonic events[1] within a piece of music[10]. These onset times are then used within a beat detection algorithm in order to calculate the piece of music's tempo.

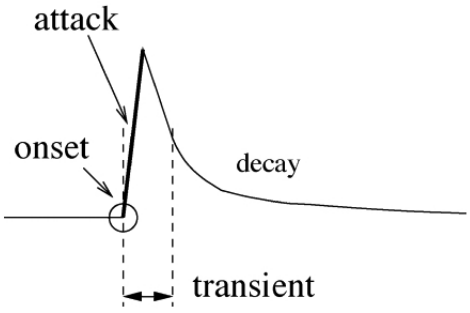

Figure 3: The onset of a note is the instant which marks start of the variation in the frequency of a signal (Image Source: [12])

Since Longuet-Higgins' first work the are of beat detection has expanded rapidly, in 2005 the first annual Music Information Retrieval Evaluation eXchange (MIREX) was held in 2005. MIREX includes a contest with the goal of comparing state-of-the-art algorithms for music information retrieval [13]. The topics to be evaluated are proposed by the participants. In the first year, three of the nine topics concerned beat detection, including audio onset detection and since it's first inclusion it has been an evalulated topic of all but one of the last twelve contests [10]. The beat detection algorithms proposed to perform the tempo analysis for this project, the Beatroot system developed by Simon Dixon [7] and a development on the original audio analysis using the Discrete Wavelet Transform (DWT) by Tzanetakis, Essel and Cook [14]. Are both former entrants the MIREX contest, with the beatroot system receiving the highest score in the 2006 Audio Beat Tracking task [15]

1.4 Project Aims

The primary aim of this project is to investigation whether some of the currently avaliable beat detection algorthims are accurate enough to form the basis for a training tool to be used by drummers practicing on an acoustic drum kit. In order to achieve this the developed software package, hereafter referred to as RTT_Analyser[2], will need to process enough live audio in form of preconstructed drum beats and record each of the chosen beat detection algorithms accuracy.

The original core features of the RTT_Analyser developed for this project are:

1. A live audio tempo analysis tool, that compares and records the performance of selected beat detection algorithms
2. The RTT_Analyser implements an adapted version of the beat tracking system Beatroot and Discrete Wavelet Transform algorithm described by Tzanetakis *et al*'s[14] which process live audio as opposed to originally designed off-line audio files.

[1] A sonic event is a singular feature of a piece music which can be made up of one source or many[11], e.g. the hitting of a drum

[2] Where RTT stands for Real Time Tempo

3. RTT_Analyser implements a concurrent system that provides the same captured live audio data to the chosen beat detection algorithms in order for the tempo to be calculated simultaneously.
4. While processing live audio the RTT_Analyser stores a predetermined data set in order to allow for performance analysis of the beat detection algorithms.
5. The RTT_Analyser will provide the user with real time feedback of the most recent tempo calculation returned
6. In order to An extensive sample set of drum beats will need to be created to ensure the system is tested sufficiently

2 Solution Design

The basic premised for the RTT_Analyser is to provide enable the user to play a live drum beat through the system and the tempo of the live audio is returned to user as well as being stored for future analysis. Initially, the RTT_Analyser opens the inbuilt microphone of the device upon which the software is being run. After establishing the live audio stream the RTT_Analyser beat detection algorithms are sent the live audio data in the format of a byte array. These live audio bytes are then decoded according to the individual algorithm's requirements before being processed and the tempo calculated. A general schematic of the work-flow of the RTT_Analyser can be seen in Figure (add in).

2.1 Live Audio Processing

2.2 RTT_Analyser Beat Detection

The original proposed solution incorporated two beat detection algorithms, Beatroot system [7] and the DWT method [14]. However during the development the adaptation of the Beatroot system to be used with live audio took longer than the proposed time-frame. This meant that an alternative system needed to be found in order to mitigate this issue, conveniently the Beatroot software package also contained another beat detection system, the Performance Worm [9]. In order to ensure the project remained on track it was decided to substitute the Performance Worm for the Beatroot system. The intention was, if time allowed, to resume the adaptation of the Beatroot system to work with live audio once the project was ahead of schedule and this was successfully completed prior to the development of the user interface.

The project report will now discuss all of the beat detection algorithms that were incorporated in the RTT_Analyser as well as the system used to allow for simultaneous tempo calculation.

2.2.1 Beatroot

Beatroot is a beat detection software package created by Simon Dixon [7], which was originally designed extract musical expression information musical recordings[9]. Beatroot was included in the RTT_Analyser as the algorithm designed by Dixon was considered to be sufficiently fast enough to be implemented as part of a real-time system[16].

Beatroot works by first obtaining a time-frequency representation of the signal based on a Short Time Fourier Transform (STFT) using a Hamming window[8]. The STFT is a form of Fourier

transform (FT), which can be used to find out how much of each frequency exists in a signal. Th A negative of the FT is that it is unable to provide any details of when a frequency component occurs in time for non-stationary signals[3]. A solution to this is to split a non-stationary signal up into a number of smaller segments using a window function, which effectively created a series of stationary[4] signals which the FT could then be applied to. By splitting the signal into smaller segments the STFT is able to apply the DFT to these segments and essentially express a signal as a linear combination of elementary signals that are easily manipulated. The DFT returns a spectrum that contains information about how the energy held within the signal is distributed in the time and frequency domains[19]. The use of a Hamming window function in the STFT employed by Beatroot is an attempt to reduce the amount of additional frequencies appearing in the returned DFT spectrum, known as spectral leakage, which is caused when the steep sloped of the a rectangular window causes the frequecies to become distorted. The Hamming window counteracts this by using a bell shape, which has the effect of reducing amplitudes of its sidelobes[18] and ensures the spectrum returned is less spread out and closer to the ideal theoretical result[19]. A visualisation of a Hamming window can be seen in Figure 4.

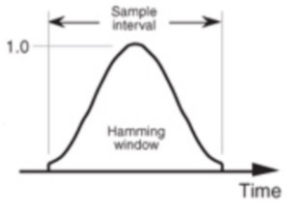

Figure 4: (Image Source: [18])

Once the time-frequency representation is returned the next stage is to the spectral flux onset detection function, which is a method for measuring the change in magnitude of each returned frequency bin[8]. The onsets are then selected from the spectral flux onset detection function by a peak-picking algorithm which finds local maxima within the detection function. The next stage is to apply the tempo induction algorithm which is used to compute clusters of inter-onset intervals (IOI) by using the calculated onset times. Each cluster represents a hypothetical tempo, in seconds per beat [7]. The clustering algorithm works by assigning an IOI to a cluster if its difference from the cluster is less than 25ms. The cluster information is then combined by recognising the approximate integer relationships between clusters. An example of this can be seen in Figure 5 where cluster C2 is twice as long as C1 and C4 is twice that of C2. This information along with the number of IOIs within a cluster is then used to weight each cluster which is then returned as a ranked list of tempo hypotheses[16].

The multiple agent architecture of Beatroot's beat tracking subsystem is then employed to find the sequences of events that closest match the original tempo hypotheses, each of these sequences is then rated and the most likely set of beat times is determined. Each of the agents in initialised with a tempo or beat rate hypothesis and an onset time. Further beats are then predicted by the agent based on these parameters. Any onsets corresponding with the predicted beat times are taken as a beat time, while those that fall outside of the are considered not to be a beat time, although the possibility that the onset is not on the beat is considered. Then agents then rate themselves based on an evaluatio function which looks at how evenly spaced the beat times are, the number of predicted beats which relate to actual events, and the salience[5] of the matched onsets. The agent

[3]Non-stationary signals are signals whose frequency contents changes over time[17]

[4]The frequency contents of a stationary signal does not change over time

[5]The salience is a measure of the note duration, density, pitch and density[7], which is calculated from the spectral flux of the onset[16]

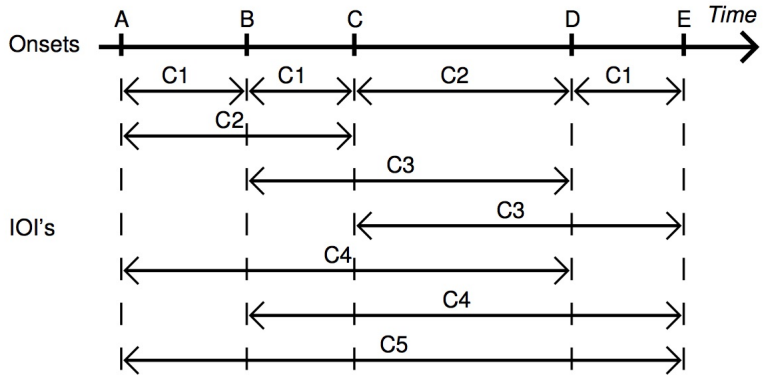

Figure 5: (Image Source: [16])

with the highest score is then returned as the sequence of beats corresponding to the processed audio[16]. It is from the sequence of beats which this agent returns that overall tempo of the audio can be determined by using the "inter-beat intervals, measured in beats per second"[7] to calculate beats per minute (bpm) of the audio.

2.2.2 Discrete Wavelet Transform and Beat Detection Method

The first literature regarding the wavelet was provided by the mathematician Albert Haar in 1909 [18]. The wavelet transform is a technique for analysing signals which was developed as an alternative to the STFT[14]. Like the STFT, the DWT is able to provide time and frequency information, however, unlike the STFT the DWT is able to do this without the need for a window function. The DWT can essentially be considered to be a filterbank, where a filterbank is a system used to separate subbands by using an array or bank of filters, where each of the filters corresponds to half frequency range of the closest centre higher frequency. Thus each filter will have half or twice the bandwith of any of its adjacent filters.

In 2001, Tzanetakis *et al* described how the Discrete Wavelet Transform (DWT) could be used to extract information from non-speech audio[14]. Their beat detection algorithm was based on the detecting the most prominent signals which are repeated over a period of time within the analysed audio. The first stage is to process is to split the signal into a number of octave[6] frequency bands with the DWT. This allows for the time domain amplitude envelopes of each frequency band to be extracted separately. The extraction of these envelopes is completed in the following three steps:

1. Full Wave Rectification - process of converting the amplitude of each frequency band to one polarity[22], which can be either positive or negative. A visual representation can be seen in Figure 6.
2. Low Pass Filtering - Low pass filtering is a signal processing technique which is designed to allow frequencies below a cutoff frequency through but blocks any frequencies above the cutoff frequency[21].
3. Downsampling - Due to the large perodicities that can occur in beat analysis, downsampling the signal reduces the computation time of the autocorrelation stage without causing any negative effects on the performance or the algorithm[20]

After these steps each frequency band os normalised through a method of mean removal in order

[6] define an octave

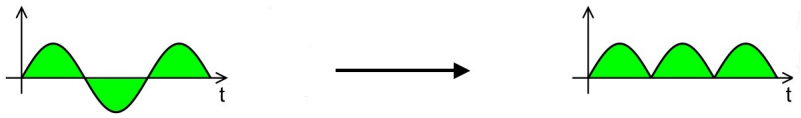

Figure 6: Visual representation of Full Wave Rectification (diagram adaptedfrom https://en.wikipedia.org/wiki/Rectifier)

to ensure the signal is centred at zero for the autocorrelation stage. The autocorrelation function is then applied to each frequency band and its peaks correspond to the various periodicities of that signal's envelope. The first five peaks of the function and their corresponding periodicities are then calculated in beats per minute and added to a histogram, this process is repeated while iterating over the signal. The estimated tempo of the audio signal is then retrieved from the periodicity that corresponds to the highest peak within the histogram[14].

2.2.3 Performance Worm

The final beat detection algorithm employed in the RTT_Analyser is the Performance Worm (PW) system, designed by Simon Dixon, Werner Goebl and Gerhard Widmer[23]. The PW is based on a real time algorithm which is able to determine the tempo of the input raw audio, while keeping track of other possible tempo hypotheses that are rated and updated dynamically. Allowing for the most recently highest ranked tempo hypothesis to be returned to the user[23].

First the PW processes the raw audio which can either taken from a static recording or directly from a live input with a smoothing filter in order to obtain the RMS amplitude of the signal taken from a 40ms window (explain RMS and smoothing filter). The note onsets are then calculated by the event detection module that finds the slope of the smoothed amplitude and then calculates the set of local peaks which are taken as the note onset times[23].

The signal is then processed by the multiple tempo tracking subsytem which uses a similar approach to the beatroot system. The time intervals (inter-onset intervals or IOIs) of event pairs are first calculated. A clustering algorithm (Figure 7) is then applied in order to determine the significant clusters of IOIs, which are subsequently assumed to be the musical units held within the signal. As in the beatroot system, these clusters are then used as the bases of the tempo hypotheses produced by the tempo tracking subsystem. While running the clustering algorithm keeps 5 seconds of onset times within memory and begins processing by determining all IOIs between the onsets in memory[23].

For IOI times t from 100ms to 2500ms in 10ms steps

2.2.4 Akka Actors

3 Implementation

3.1 Live Audio Processing

The live audio will be processed using the Javax Sound package. The audio will be captured using a stereo microphone and processed to match CD quality with the Javax Sound AudioFormat class. The Beatroot system was not originally intended to be used as a real time system [19] so currently

For each new onset
    For IOI times t from 100ms to 2500ms in 10ms steps
        Find pairs of onsets which are t apart
        Sum the mean amplitude of these onset pairs
    Loop until all IOI time points are used
        For times t from 100ms to 2500ms in 10ms steps
            Calculate window size s as function of t
            Find average amplitude of IOIs in window $[t, t + s]$
            Store t which gives maximum average amplitude
        Create a cluster containing the stored maximum window
        Mark the IOIs in the cluster as used
    For each cluster
        Find related clusters (multiples or divisors)
        Combine related clusters using weighted average
    Match combined clusters to tempo hypotheses and update

Figure 7: Clustering algorithm used in the Performance Worm Multiple Tempo Tracking Subsystem[23]

only works with prerecorded audio. It will therefore be will need to be modified in order for it to work with live audio.

- Encoding - This will be set to "PCM.signed", representing audio encoded to the native linear pulse code modulation, where quantization levels are linearly uniform [15].
- Sample Rate - 44,100, set to match CD quality for the number of analog samples which will be analysed per second.
- Sample Size in Bits - 24, based on a sound card with a 24 bit sample depth.
- Channels - 2, audio will be captured using a stereo microphone.
- Frame Size - 6, where the frame size is the number of bytes in a sample multiplied by the number of channels [17].
- Frame Rate - 44,100, same as sample rate.
- Big Endian (boolean) - false, as the project will be developed on an Intel core which uses a little-endian architecture[7].

[7]Endianess refers to the order of bytes which make up a digital word. Big endianess stores the most significant byte at a certain memory address and the remaining bytes being stored in the following higher memory addresses. The little-endian formate reverses the order storing the least significant at the lowest and most significant at the highest memory address [16].

References

[1] *https://www.britannica.com/art/metronome*

[2] Mick Berry and Jason Gianni *The Drummer's Bible: How to Play Every Drum Style from Afro-Cuban to Zydeco, Second Edition, 2004, See Sharp Press*

[3] Alison Latham *The Oxford Companion to Music, 2002, Oxford University Press*

[4] *http://www.drummagazine.com/lessons/post/drumkey/*

[5] Allen and Dannenberg *Tracking Musical Beats in Real Time, International Computer Music Conference, International Computer Music Association, 1990, pp. 140-143*

[6] H. C Longuet-Higgins *Perception of melodies, Nature Vol. 263, 1976, pp. 646-653*

[7] Simon Dixon *Automatic Extraction of Tempo and Beat from Expressive Performances. Journal of New Music Research, 30 (1), 2001, pp 39-58*

[8] Simon Dixon *Onset Detection Revisited, Proceedings of the 9th International Conference on Digital Audio Effects, Montreal, September 2006, pp 133-137*

[9] Simon Dixon *On the Analysis of Musical Expression in Audio Signals. Storage and Retrieval for Media Databases, SPIE-IS&T Electronic Imaging, SPIE Vol. 5021, 2003 pp 122-132*

[10] *http://www.music-ir.org/mirex/wiki/2016:Audio_Onset_Detection*

[11] *http://www.ieor.berkeley.edu/ ieor170/sp15/files/Intro-to_Sonic_Events_Campion.pdf*

[12] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler *A Tutorial on Onset Detection in Music Signals, IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 13, NO. 5, 2005, pp. 1035 - 1047*

[13] *http://www.music-ir.org/mirex/wiki/2005:Main_Page*

[14] George Tzanetakis, Georg Essl and Perry Cook *Audio Analysis using the Discrete Wavelet Transform, Proc. WSES International Conference on Acoustics and Music: Theory and Applications (AMTA), 2001*

[15] *http://www.music-ir.org/mirex/wiki/2006:Audio_Beat_Tracking_Results*

[16] Simon Dixon *Evaluation of the Audio Beat Tracking System BeatRoot. Journal of New Music Research, 36, 1, 2007, pp 39-50*

[17] *http://users.rowan.edu/ polikar/WAVELETS/WTpart2.html*

[18] Lyons *Understanding Digital Signal Processing*

[19] Tao Li, Mitsunori Ogihara, George Tzanetakis *Music Data Mining, CRC Press, 2002, pp 45-53*

[20] George Tzanetakis and Perry Cook *Musical Genre Classification of Audio Signals*

[21] Steven W. Smith *The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 2011, Chapter 3*

[22] Ramn Palls-Areny and John G. Webster *Analog Signal Processing, Wiley, 199, pp. 231*

[23] Simon Dixon, Werner Goebl and Gerhard Widmer *The Performance Worm: Real Time Visualisation of Expression based on Langners Tempo-Loudness Animation, International Computer Music Conference, 16 - 21 September 2002, Gteborg, Sweden, pp 361-364.*