subjEQt: Controlling an Equalizer through Subjective Terms

Sebastian Mecklenburg
Hochschule Bremen
University of Applied Sciences
Flughafenallee 10
28199 Bremen, Germany
sebastian.mecklenburg@hs-bremen.de

Jörn Loviscach
Hochschule Bremen
University of Applied Sciences
Flughafenallee 10
28199 Bremen, Germany
jlovisca@informatik.hs-bremen.de

This research was funded by grant 1742A04 of the German Ministry of Education and Research (BMBF). The views and conclusions contained in this document are those of the authors.

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.
ACM x-xxxxx-xxx-x/xx/xxxx.

Abstract
Equalizing is one of the most important tasks in audio engineering. It also is a task that requires technical and auditory training to achieve the desired results. We propose to simplify the use of an equalizer by providing a visual arrangement of subjective terms such as 'warm', 'present', 'boomy' instead of the standard controls that closely correspond to the underlying technology.

Keywords
Music production, similarity measure, scattered data interpolation

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces—Graphical user interfaces (GUI); H5.5. Information interfaces and presentation (e.g., HCI): Sound and Music Computing—Methodologies and techniques.

General Terms: Design, Experimentation, Human Factors

Introduction

In the realm of audio, equalization means to emphasize or to suppress certain frequencies in an audio recording such as music or speech. Equalizing is used in almost all but the most simple audio production projects. It may occur at different stages of the process: The raw audio recordings are usually equalized before they are merged into a stereo track (a process called mixing) and then the final mix is equalized again (a process called mastering). Of course, mixing and mastering usually contain a lot more work than just equalizing, namely tasks such as stereo panning, compressing, limiting, exciting, applying reverb or distortion effects. But the most basic and most frequently used processing step is equalization.

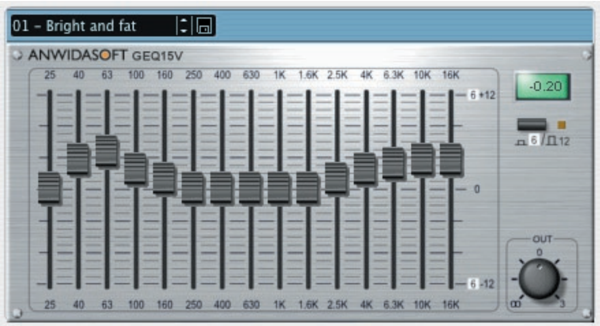

Figure 1: A standard 15-band graphic equalizer [1]

Figure 1 shows the usual interface of a software graphic equalizer, which is closely modelled after a hardware equalizer. It includes a preset menu, containing subjective terms, to make it easier for the user to realize the desired modifications of the sound.

Figure 2 shows the proposed new equalizer interface. It contains only subjective terms and computes the equalizer setting from a point the user selects on its 2D field. It is possible to adjust the sound only by subjective terms without being distracted by a technical interface. A graphic equalizer is included anyway, althoug hidden by default, so users may take a look at the resulting equalizer setting.

Subjective terms

There is a number of subjective terms that are frequently used to describe sound: boomy, boxy, impact, warm, present, nasal, muddy, cutting, edgy, airy, sizzly, etc. Interestingly, a lot of these terms can be described by the absence or presence of certain

Figure 2: The basic subjEQt interface

frequencies [2]. For example, through attenuation of the frequencies in the range from 4 to 7 kHz, a piece of music will sound less edgy. If one boosts the range between 200 and 600 Hz, it will sound warmer. Figure 3 shows a chart that relates the subjective terms to frequency ranges.

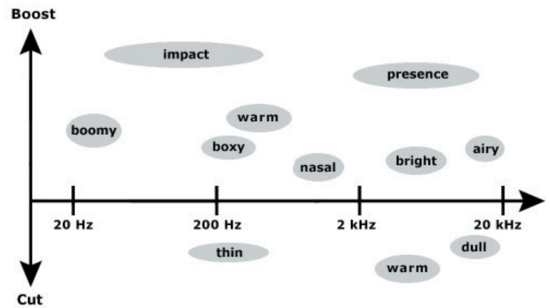

Figure 3: Subjective terms mirror peaks and valleys in the frequency spektrum. Data according to [2]

Digital Filters

The concept of a filter stems from electrical engineering, where filters are used to process signals, in particular to eliminate (filter out) certain frequencies in a signal. A peak filter, which is the type of filter we use here, can be described by its center frequency, its gain (amplification or attenuation) at this frequency and a quality (Q factor), which determines the width of the filter. The action of a filter can be described by a frequency response. This is a curve that specifies the amplification or attenuation at every frequency in decibels (dB). Figure 4 shows the frequency response of a filter that adds more warmth (cf. Figure 3). A graphic equalizer is basically a number of equally shaped, equally spaced narrow filters, with each filter amplifying or attenuating a distinct frequency band. In contrast to a graphic equalizer, a parametric equalizer comprises only a handful of filters, but allows to adjust the shape and the center frequency of each of these.

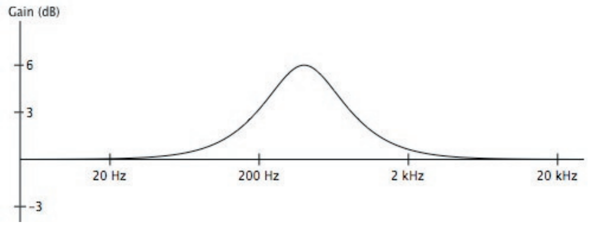

Figure 4: A slight boost at 400 Hz adds more warmth to the sound.

Similarities

One can define a number of subjective terms and specify the according filter for each. We want to place these terms on a two-dimensional area on the computer's screen. Their relative position should reflect the perceived distance: Filters with a similar effect are to be placed close to another. To accomplish this, one needs to find a similarity measure that generates a distance value for every pair of frequency responses. A straightforward mathematical solution such as computing the area between the two filter curves is unsuitable for this purpose: Two frequency responses with a given area between them may have a very different or a very similar effect.

Figure 5, 6, and 7 show three pairs of filters, where the area between each pair of curves is almost equal.

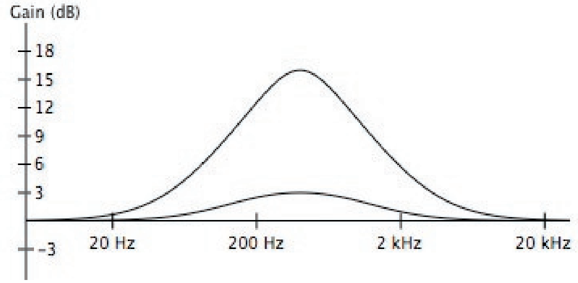

Figure 5: Two filters displaying the same center frequency, but different gain and Q.

Nevertheless, Figure 5 displays a pair of very similar responses; the responses in Figure 6 are more or less unrelated to each other; and one filter from the pair in Figure 7 has the opposite effect of the other. Thus, the two filters of Figure 5 should be close together on our equalizer interface; the two filters of Figure 6 should not be too far away from each other. The two filters

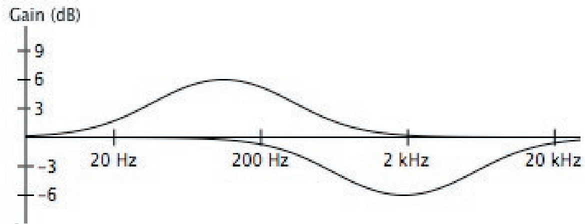

Figure 6: Different center frequency, same magnitude of gain but with different signs, same Q.

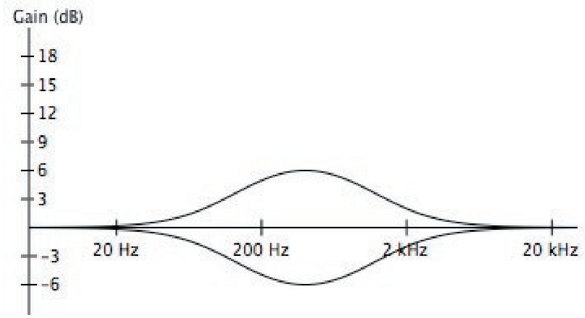

Figure 7: Same center frequency, same Q, same absolute gain, but with different signs.

of Figure 7, however, should be placed as far apart as possible. The reason for this is as follows: We want to allow the user to point anywhere with the mouse in the 2D field of subjective terms. If she points at an intermediate position, the action of the filters in the vicinity should be blended accordingly. However, it would not make much sense to blend the two filters of Figure 7: They would cancel each other, thus leaving some area of the 2D field without effect and reducing the overall number of available meaningful filter combinations. Placing such filters far apart from each other alleviates this problem.

This line of thought boils down to the following rules:

- Two filters that perform a similar action should be close together in the 2D field.
- Two filters that cancel each other (like a high cut and a high boost) should be far away from each other.
- Two filters that can be combined into a meaningful result (like a low cut and a high boost) should have a medium distance.

Thus, we came up with the following heuristic constraints to compute the similarity measure:

- If two filters are equal their similarity is 1.
- If two filters have the same Q, the same center frequency and maximum absolute gain but one is the negative of the other their similarity is 0.
- If two filters have the same Q and maximum absolute gain but their center frequencies are at the opposite ends of the frequency spectrum, their similarity is 0.5, no matter if they both amplify, both attenuate or if one amplifies and the other one attenuates.
- If the filters have the same gain, the same center frequency but different Q, the similarity is set to a value between 0.75 and 1.0, depending on how different the Q factor is. This mirrors the fact that varying the shape of the filter has a less audible effect as compared to varying its gain or its center frequency.

The distance of an arbitrary pair of filters is computed as a linear interpolation between these constraints.

Self-Organizing Maps

To arrange the subjective terms on a two-dimensional grid we use a self organizing map [3]. The self-organizing map (SOM) is a method for unsupervised learning, based on a grid of artificial neurons. It is employed to visualize high-dimensional data with a complex, nonlinear statistical relationship as simple geometrical arrangement on a low-dimensional display. As input vectors and feature vectors for the node weights we use the triple {center frequency, gain, Q-factor}. The similarity of two triples is computed using the similarity measure for the corresponding frequency response. On a grid with 40 by 40 nodes and after an SOM computation of 1000 generations this results in a visually pleasing order of the subjective terms.

Natural neighbor interpolation
Given the set of filters arranged in the 2D field, we want to allow the user to select any point and get a meaningful combination of the filters in the neighborhood. This is a special case of the well-known problem called scattered-data interpolation. A widely accepted solution to this problem is natural neighbor interpolation [4]. It works by creating a Voronoi tessellation from the data points and computing another potential Voronoi cell at the query point. The amount of influence—called the natural neighbor coordinate—of each neighbor then is the ratio of the area of overlap with the potential cell at the query point and the area of the neighbor's Voronoi cell. In our prototype we use the implementation provided in [5].

Resulting equalizer setting
To perform the actual processing, we use a standard 31 band graphic equalizer, in this case Apple's equalizer audio unit that ships with the Mac OS X operating system. To compute the final equalizer setting, we form the average of the frequency responses of the neighboring filters, weighting them by their natural neighbor coordinates. This results in a more or less complex frequency response curve. To compute the gain for each of the 31 bands of the graphic equalizer, we sample that curve at the corresponding distinct frequencies. Figure 8 shows a screenshot of the final interface and the attached equalizer.

Evaluation
It turns out that professional sound engineers and complete laymen, that is, people who "only" listen to music, find this software most interesting. What the pros liked was the ability to try out more complex equalizer settings quickly; the laymen liked the ability to learn about the effect an equalizer has. Amateur musicians, however, were more skeptical. They considered the terms still too technical and would like to be able to rename them or to insert new ones. One hypothesis to explain this preliminary observation is be the following: The terms are very common to professionals; they get what they expect. Furthermore, the terms are novel to non-musicians; they do not object to what they get. Amateur musicians, however, may have heard some of the terms already, but have formed their own ideas of what they mean. So they may get different results from what they expect, which is a frustrating experience, in particular for this kind of interface which relies on agreed-upon meanings for subjective terms.

Another problem that surfaced in our first informal tests was the empty space between the terms and the way in which intermediate equalizer settings are computed. The seemingly empty space is considered less important and is less frequently used, quite contrary to the way the interface is supposed to work. The interesting and important settings are definitely between the terms.

Future Work
An equalizer can only emphasize or suppress frequencies that are already present in an audio reording. So there are ways to alter the sound of an audio reording that cannot be done by equalizing. For example, sound can also be made warmer by adding selective harmonics. This could be achieved by implementing an harmonic distortion effect in addition to the equalizer.

Advanced users should be able to customize the interface, that is, to insert new terms with an associated filter and to rename the existing ones. Although the provided terms are common sense in audio engineering, they are not some objective truth or fact.

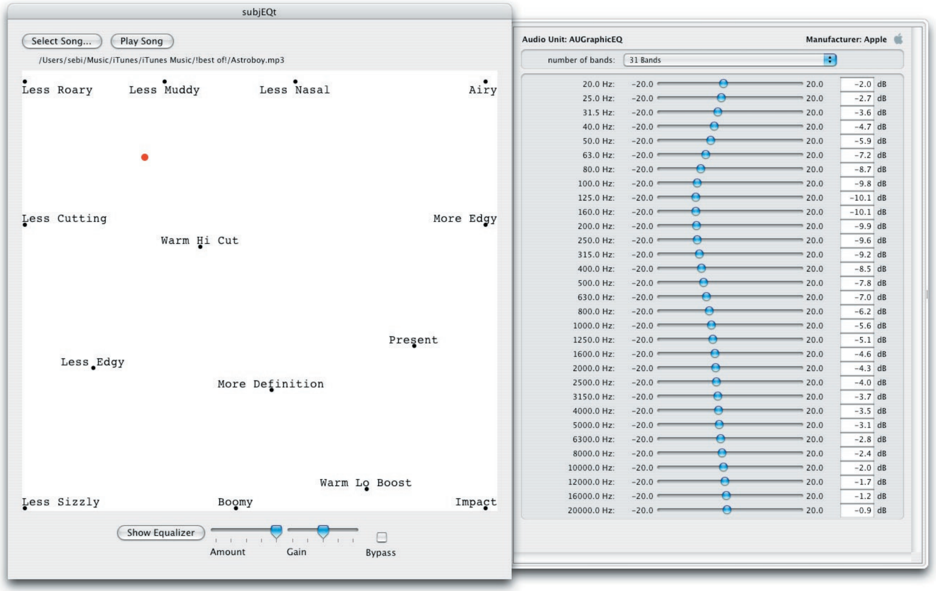

Figure 8: subjEQt in action: making a sound warmer and less muddy by reducing the low frequencies around 160 Hz.

A 31-band graphic equalizer has to provide 31 narrow filters, which is not computationally efficient and causes a loss in quality due to a complex phase response. Furthermore, a graphic equalizer works at distinct, equally spaced frequencies. So it is not able to accurately mirror a boost or cut at arbitrary intermediate frequencies. Due to all of these drawbacks it would be better to use a parametric equalizer, that is, an equalizer with only two or three filters, each with adjustable center frequency, gain, and width. The problem here is to find the right parameters for these filters so that the result approximates the user's choice best. This leads to a complex optimization problem.

Instead of displaying a solid color background it could be fruitful to provide a background with color gradients that hint at the interpolation of the intermediate settings between the terms.

Finally, it could be useful to simultaneously select several points on the UI. The corresponding filters would be applied in a serial connection, which translates into simply adding the per-band gain settings of the graphic equalizer. This way, the user can create more complex equalizer settings, at the price of an interface that is more difficult to handle. One could even provide a two-mouse input system, so the user would be able to manipulate two points at the same time. This could make it easier to achieve some desired result—or just to explore the settings.

References

[1] ANWIDA's GEQ15V VST plugin, http://www.anwida.com/

[2] Katz, B. (2002), Mastering Audio, the Art and the Science, Focal Press

[3] Kohonen, T. (2001), Self-Organizing Maps, Springer

[4] Sibson, R., (1981), A Brief Description of Natural Neighbor Interpolation, Interpreting Multivariate Data, John Wiley & Sons, New York, pp. 21–36

[5] Computational Geometry Algorithms Library, http://www.cgal.org