SOUND TRANSFORMATION BY DESCRIPTOR USING AN ANALYTIC DOMAIN

Graham Coleman
Music Technology Group,
Universitat Pompeu Fabra
Barcelona, Spain
gcoleman@iua.upf.edu

Jordi Bonada
Music Technology Group,
Universitat Pompeu Fabra
Barcelona, Spain
jbonada@iua.upf.edu

ABSTRACT

In many applications of sound transformation, such as sound design, mixing, mastering, and composition the user interactively searches for appropriate parameters. However, automatic applications of sound transformation, such as mosaicing, may require choosing parameters without user intervention. When the target can be specified by its synthesis context, or by example (from features of the example), "adaptive effects" can provide such control. But there exist few general strategies for building adaptive effects from arbitrary sets of transformations and descriptor targets. In this study, we decouple the usually direct link between analysis and transformation in adaptive effects, attempting to include more diverse transformations and descriptors in adaptive transformation, if at the cost of additional complexity or difficulty. We build an analytic model of a deliberately simple transformation-descriptor (TD) domain, and show some preliminary results.

1. INTRODUCTION

Sound transformations are practically used by sound and music producers in a variety of contexts: mixing, mastering, synthesis, composition, sound design for varying media. Effects are typically modeled as mathematical functions transforming one or more input audio signals into output signals according to a parameter set. These parameters usually are tuned either interactively or according to some knowledge of the transformation domain. Because this process can be immediate and interactive, it is usually fast and effective for a user to find parameters which correspond to their target percepts for the input sounds in question. The assumption that parameters can be effectively manually tuned breaks down under several conditions: the parameter space is too large (in terms of cardinality), too complex to be interactively searched, or needs fine detail in time for the desired result.

For example, consider an automatic mosaicing system that selects "source" sound samples from a database to match input "target" samples, then composites them into a score. We could transform the retrieved sounds to be more similar to their targets, for example, but we would need to select transformation parameters without human input.

Adaptive effects "in which controls are derived from sound features" [1] [1] are often implemented directly via analysis-synthesis. This offers a direct route to control effects by descriptors, usually by exploiting mathematical properties of transforms such as the STFT or the source filter model in such a way that allows independent algorithmic modification of several properties of the sound. In these systems, the analysis of the target is coupled directly with the transformation of the input material. However, this requires that we develop such a transformation model, fixed to the target descriptor set, if it exists.

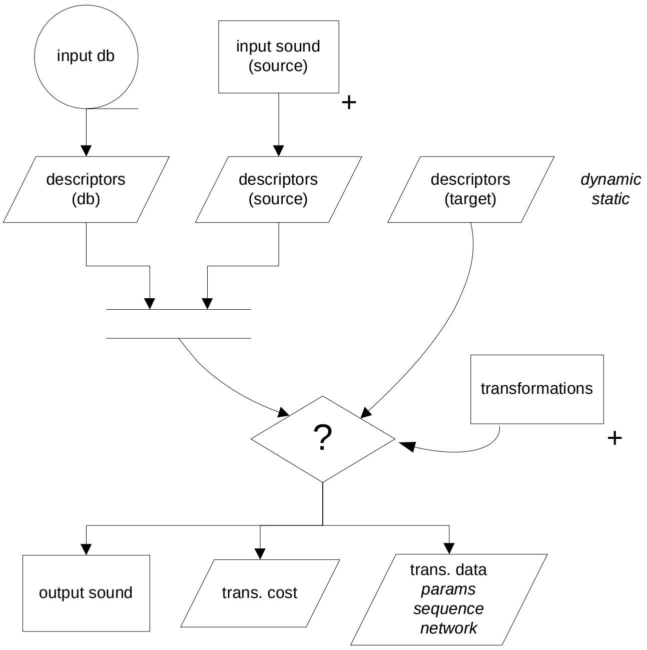

Figure 1: An ideal transformation by descriptor system, which uses available transformations to bring input sounds close to a target, and also discriminates between cantidate source sounds in a database.

An alternative approach would allow consideration of any target descriptors of interest, using the set of transformations that are available. By breaking the link between analysis and transformation, we intend to allow a wider and more complex set of criteria to be considered.

Instead of using a transformation domain that allows direct modification according to a target, we propose building models of the transformation-descriptor (TD) space. By determining the relationship between the input sound, the transformation parameters, and the output descriptors, we provide a map of the space

[1] In the field of pattern matching, *features* are statistics computed from examples used for classification, etc. From more recent computer audio literature, *descriptors* refer to information about media content to aid processing, presentation, etc. Here we use them interchangeably.

which can be used for finding parameters that meet the target. By using numerical optimization techniques (search) on this model informed by the input descriptors, we provide suitable parameters for the transformation, thus reconnecting the analysis-transformation chain.

2. SOME RELATED WORK

Several previous work develop transformations controlled by other signals, including Verfaille and Depalle [1], which introduces a taxonomy for distinguishing these so-called Adaptive Effects. In the examples provided, the STFT and source filter model form a basis over which some aspects of the sound (at once descriptors and parameters) can be independently modified.

As we include arbitrary or more complex sets of descriptors, the chance of extending or discovering such models lessens. We propose modeling a domain of parametric transformations with respect to target descriptors, and searching for links between the two. We will refer to this as a Transformation-Descriptor (TD) domain. The cost of this more flexible and general approach is that if chosen inappropriately, the connection between transformations and descriptors may be poor.

Concatenative synthesis synthesizers ranging from a single-instrument to audio mosaicing use the similar techniques ([2], [3]). These systems generate sequences with different target trajectories from a limited sample database, and could likewise be extended by transforming the input sound closer to its intended target. In this application context, we would also like to be able to select source samples that are most easily transformed to the targets. Figure 1 illustrates Transformation by Descriptor in a mosaicing context.

Perhaps the closest work and a current inspiration is the parellel work being done in synthesis by Hoffman and Cook [4], which also uses analysis-synthesis indirection and numerical optimization to control parametric synthesizers from frame-based audio features, and thus explores a similar technique. But we expect that the specifics of working in the transformation domain, of potentially working at different time scales, of differing input signals, make the problems and questions faced different enough to merit separate investigation.

3. PREDICTING DESCRIPTORS UNDER TRANSFORMATION

By modeling our transformation space, we intend to guide the search process towards its intended target. We model each transformation as a function mapping an input sound, represented by its descriptors, and a parameter vector, to an output sound again represented by descriptors.

Feature vector of input sound : $\vec{d_{in}}$
Hopefully something that predicts $\vec{d_{tr}}$ well.

Feature vector of transformed sound : $\vec{d}_{tr\,i=1\ldots D}$
We have D descriptors of interest, need not match $\vec{d_{in}}$.

Vector of transformation parameters : $\vec{w}$

The transformation function : $\vec{t}(\vec{d_{in}}, \vec{w}) = \vec{d_{tr}}$

Model of transformation : $\hat{t}(\vec{d_{in}}, \vec{w}) = \vec{d_{tr}} + e$
In other words, an approximation.

Target feature vector : $\vec{a}_{i=1\ldots D}$, matches $\vec{d_{tr}}$

For these experiments, we choose a simple TD domain to model changes by simple transformations on Fourier domain descriptors. Our descriptors are statistical moments of the spectrum, currently spectral centroid and standard deviation; with later tests on skewness, kurtosis, and higher central moments:

$$\vec{a} = (a_{mean}, a_{std}, a_{skew}, a_{kurt}, a_{cm5}, \ldots) \qquad (1)$$

and our transformations are bandlimited interpolation (resampling) and linearly spaced bandpass equalization. In this case, the transformation parameters are the resampling factor w_L, and the gains for B equalization bands w_j:

$$\vec{w} = (w_L, \vec{w}_{j=1\ldots B}) = (w_L, w_1, w_2, \ldots, w_B) \qquad (2)$$

We will introduce a model in which effects of both transformations in the spectrum can be predicted precisely. We omit modeling two phenomena which become sources of error in predicting the effects of real transformation. One source of error is the bandlimited interpolation which relies on a bank of filters to reconstruct the ideal sinc function, and the other source of error comes from using a reduced version of the input spectrum (a filter bank) to approximate the full spectrum.

3.1. Resampling

To understand the basic workings of upsampling and downsampling, we can use a few Fourier theorems. First, in a consequence of the Periodic Interpolation (Spectral Zero Padding) theorem, [5]:

$$Interp_L(x) \longleftrightarrow ZeroPad_{LN}(X) \qquad (3)$$

we see that ideal interpolation in the time domain is equivalent to zero padding in the spectral domain. In practice, the reconstruction filters are less than ideal but we omit specific consideration of these filters for the simplicity of our model. [2]

For our model of t_r, this means for $w_L > 1$ (upsampling), resampling simply means using the same set of Fourier coefficients, just adding zero coefficients, and reinterpreting the existing ones at lower frequencies.

For downsampling ($w_L < 1$), the picture gets more complicated. From the Downsampling Theorem [5] we have:

$$DownSample_L(x) \longleftrightarrow \frac{1}{L} Alias_L(X) \qquad (4)$$

Since aliasing is an unwanted outcome, we must bandlimit the input spectrum so the output will not overlap. In both cases, we scale the input spectrum frequencies according to the resampling factor w_L, bandlimiting the input if we are performing a downsampling ($w_L < 1$).

Using the spectrum as our input descriptor (or an approximation by filter banks) we model the moments of the upsampled or downsampled signals, each moment specified by an $m(k)$:

$$\hat{t}_{r,m(k)}(X, \vec{w}) = \frac{\sum_{k=1}^{K} |X_k| \cdot bstep(\frac{k}{K \cdot w_L}) \cdot m(\frac{k}{w_L})}{\sum_{k=1}^{K} |X_k| \cdot bstep(\frac{k}{K \cdot w_L})} \qquad (5)$$

[2] for further detail see Bandlimited Interpolation of Time-Limited Signals [5]

where $bstep$ is derived from the Heaviside step function H:

$$bstep(x) = 1 - H(x-1) = \begin{cases} 1 & x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and $m(k)$ is the enclosed function in any expectation of the spectral distribution, such as $m_m(k) = k$ for the mean, $m_{var}(k) = (k-\mu)^2$ for the variance, and $m_n(k) = (k-\mu)^n$ for higher central moments.

3.1.1. Sigmoids

However, a step function is discontinuous, and thus less than ideal for analysis; its derivative is zero everywhere except at the discontinuity. Thus it is less useful in a local search technique, in which we rely on derivatives to tell us about the local behavior of the function.

Instead we use a surrogate composed with the sigmoid function:

$$\begin{aligned} bsmooth_\alpha(x) &= 1 - P(\alpha(x-1)) & (7) \\ &= 1 - \frac{1}{1+e^{\alpha(x-1)}} & (8) \end{aligned}$$

where the sigmoid function is defined as:

$$P(x) = \frac{1}{1+e^{-x}} \tag{9}$$

with the derivative:

$$\frac{dP}{dx} = P(1-P). \tag{10}$$

With a sigmoid, we can have a smooth function to optimize. (Indeed, in this manner they facilitate backpropogation, gradient descent derived for feedforward neural networks.)

This does not completely solve the problem of the local minima created by the resampling, it just allows the search to find the local minima easier by giving them directional cues.

3.2. Equalization

We model equalization as rectangular filters that partition the spectrum into B bands, apply non-negative gains $\vec{g}_j$, and then add the scaled signals. In practice, the filters used will overlap and contain small amounts of energy from other bands.

We can model each transformed Fourier coefficient as being scaled by the appropriate gain:

$$\hat{t}_{eq,X_k}(X, \vec{g}) = g_{j(k)} \cdot X_k \tag{11}$$

where $j(k)$ is just the filter bank that corresponds to the spectral bin k.

One way of constraining the gains to be non-negative is to use log-domain (exponents) to control the gains, such as $g_j = 2^{w_j}$, giving us:

$$\hat{t}_{eq,X_k}(X, \vec{w_j}) = 2^{w_{j(k)}} \cdot X_k \tag{12}$$

As well, exponents are frequently used as gains in the form of dB controls, to which could convert our w_L by multiplying by $\frac{20}{\log_2(10)}$.

In this experiment, we divide the spectrum evenly into linearly-spaced bands.

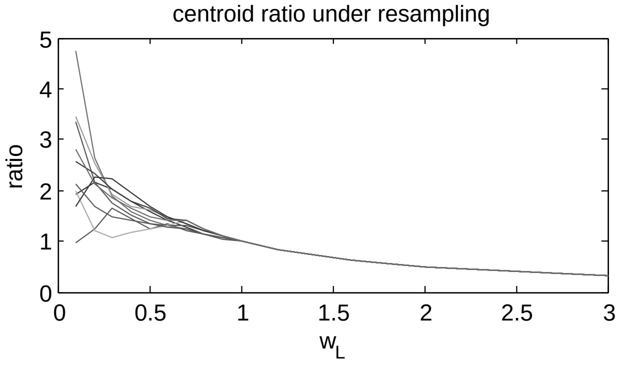

Figure 2: Centroid ratios (wrt the centroid at w_L) diverge when $w_L < 1$.

3.3. Composition

For a combined model, we can compose the two transformations like so:

$$\hat{t}(X, \vec{w}) = \hat{t}_r \circ \hat{t}_{eq} = \hat{t}_r(\hat{t}_{eq}(X, \vec{w}_j), w_L) \tag{13}$$

This helps us coordinate the combined effects of the transformation. Conceptually, independent sets or bands of Fourier coefficients are scaled by the equalization, which are then shifted and possibly bandlimited by the resampling. If we composed them the other way, membership in a given equalization band would vary with resampling parameter w_L.

Composing the two transformations into one function gives us:

$$\hat{t}_{m(k)}(X, \vec{w}) = \frac{\sum_{j=1}^{B} 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \cdot \sum_{k \epsilon k(j)} |X_k| \cdot m(\frac{k}{w_L})}{\sum_{j=1}^{B} 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \cdot \sum_{k \epsilon k(j)} |X_k|} \tag{14}$$

3.4. Behavior of Transformation / Model

To give an intuition for the shape of the space we wish to model and search, we examine several plots. The first, in figure 2, shows the effect of resampling on a set of 10 sounds. On the right side of the function, when $w_L \geq 1$, all transformed centroids have the same behavior, which is an even $\frac{trans.centroid}{inputcentroid}$ ratio with respect to w_L. In strong contrast, to the left side, when $w_L < 1$, the function behavior depends completely on the spectrum of the sound as different pockets of energy are bandlimited away causing fluctuations in the transformed centroid.

The next figure (3) shows a grid sampling of a cross-section of the real transformation space of resampling and equalization for the standard deviation as a descriptor. We see the undulations in the resampling dimension, but looking at slices of the equalization parameters look much smoother, almost like sigmoids!

As these moments are weighted sums over the spectrum, the position of the band determines the value it will contribute to the sum independently of the energy in that band, which determines

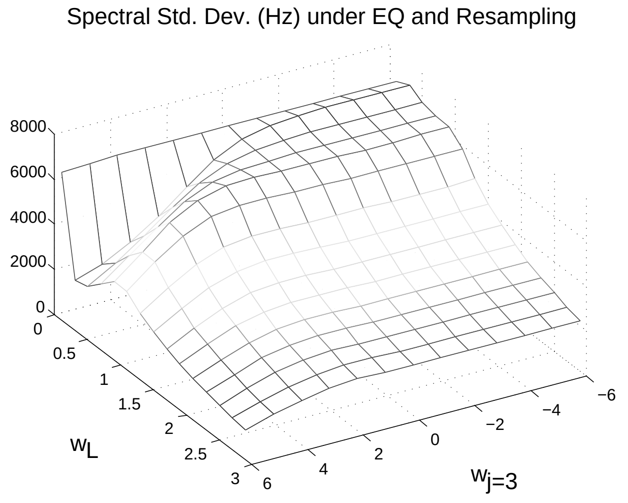

Figure 3: Cross-section of transformation space for one variable band (out of 16) and a resampling parameter.

the strength with which it will pull towards that value, making them monotonic with respect to the weighted spectrum. When you amplify a band far above the rest of the bands, or far below, it either predominates or becomes insignificant in the summation, giving it the slow limiting effect as seen. This gives us hope that in the error space will be relatively smooth and easy to follow the gradient in these dimensions, with the bumpiness being confined to the resampling dimension.

4. MODEL OPTIMIZATION

For each sound we which to transform, we have input descriptors $\vec{d_{in}}$, a transformation function $\hat{t}(\vec{d_{in}}, \vec{w})$, and a target descriptor vector $\vec{a}$. When we use a model to approximate the effect of the transformation, we replace $\vec{t}$ with $\hat{t}$ and an error term e. We define the residual as distance from the target for each descriptor of interest:

$$\vec{r}(d_{in}, \vec{w}) = \hat{t}(\vec{d_{in}}, \vec{w}) - \vec{a} + e. \tag{15}$$

To optimize the transformation parameters, we then minimize some function of the residual. For mathematical convenience, the sum of squares of the individual terms (least-squares) is often chosen. [6]

$$f(\vec{d_{in}}, \vec{w}) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(\vec{d_{in}}, \vec{w}) \tag{16}$$

For a particular input sound $\vec{d_{in}}$ will be fixed, so we will disregard it in the optimization notation.

4.1. Partial Derivatives

Partial derivatives of the objective function f are a basic ingredient of local search techniques (such as gradient descent). The gradient is defined as the vector of partial derivatives with respect to each of the parameters:

$$\nabla f(\vec{w}) = (\frac{\partial f}{\partial w_L}, \frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_B}) \tag{17}$$

Under the least-squares error criteria, the gradient reduces to:

$$\nabla f(\vec{w}) = \sum_{j=1}^{m} r_j(\vec{w}) \nabla r_j(\vec{w}) = J(\vec{w})^T \vec{r}(\vec{w}) \tag{18}$$

where J, the Jacobean, is the partial derivative of the residual in each parameter:

$$J(\vec{w}) = \begin{bmatrix} \frac{\partial r_1}{\partial w_L} & \frac{\partial r_1}{\partial w_1} & \cdots & \frac{\partial r_1}{\partial w_B} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_D}{\partial w_L} & \frac{\partial r_D}{\partial w_1} & \cdots & \frac{\partial r_D}{\partial w_B} \end{bmatrix} \tag{19}$$

So, to find the gradient for our model, we need only formulate the Jacobean, or the partial derivatives of the residual with respect to our model.

Since the target $\vec{a}$ is constant and the error term e can be assumed to be independent, the partial derivatives of the residual are simply the partial derivatives of the transformation model $\hat{t}$, have the following form (derivative of a quotient):

$$\frac{\partial \vec{r}}{\partial \vec{w}} = \frac{\partial \hat{t}}{\partial \vec{w}} = \frac{\frac{\partial top}{\partial \vec{w}} bot - top \frac{\partial bot}{\partial \vec{w}}}{bot^2}, \tag{20}$$

given subexpressions for the top and bottom:

$$top = \sum_{j=1}^{B} 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \sum |X_k| \cdot m(\frac{k}{w_L}) \tag{21}$$

$$bot = \sum_{j=1}^{B} 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \sum |X_k|. \tag{22}$$

We define a subexpression for the summation over each band:

$$band_j = \sum_{k \epsilon B_j} |X_k| \cdot m(\frac{k}{w_L}) \tag{23}$$

This gives us the following partial derivatives for w_L:

$$\frac{\partial band_j}{\partial w_L} = \sum_{k \epsilon B_j} |X_k| \cdot m'(\frac{k}{w_L}) \cdot \frac{-k}{w_L^2} \tag{24}$$

$$\begin{aligned} \frac{\partial top}{\partial w_L} = \sum_{j=1}^{B} 2^{w_j} \cdot (bsmooth'(\frac{j}{B \cdot w_L}) \cdot \frac{-j}{B \cdot w_L^2} \cdot band_j + \\ bsmooth(\frac{j}{B \cdot w_L}) \cdot \frac{\partial band_j}{\partial w_L}) \end{aligned} \tag{25}$$

where:

$$bsmooth'_\alpha(x) = -\alpha \cdot P(\alpha(x-1)) \cdot (1 - P(\alpha(x-1))). \tag{26}$$

The bottom expression is similar to the top, but the constant m, makes the partial term disappear:

$$\frac{\partial bot}{\partial w_L} = \sum_{j=1}^{B} 2^{w_j} \cdot bsmooth'(\frac{j}{B \cdot w_L}) \cdot \frac{-j}{B \cdot w_L^2} \cdot band_j \tag{27}$$

and the following partial derivatives for w_j:

$$\frac{\partial top}{\partial w_j} = 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \sum |X_k| \cdot m(\frac{k}{w_L}) \qquad (28)$$

$$\frac{\partial bot}{\partial w_j} = 2^{w_j} \cdot bsmooth(\frac{j}{B \cdot w_L}) \sum |X_k| \qquad (29)$$

The derivatives above should be sufficient for the centroid, and moments about the mean. For the standard deviation, we must add a square root to our expression:

$$\frac{\partial t_{std}}{\partial \vec{w}} = \frac{\partial \sqrt{t_{var}}}{\partial \vec{w}} = \frac{\frac{\partial t_{var}}{\partial w}}{2 \cdot \sqrt{t_{var}}} \qquad (30)$$

For standardized moments, such as skewness, kurtosis, and beyond, we must divide by powers of the standard deviation:

$$\begin{aligned} \frac{\partial t_{\frac{\mu^n}{\sigma^n}}}{\partial \vec{w}} &= \frac{\partial}{\partial \vec{w}}(\frac{t_{\mu^n}}{\sigma^n}) \\ &= \frac{\frac{\partial t_{\mu^n}}{\partial w} \cdot \sigma^n - t_{\mu^n} \cdot (n\sigma^{(n-1)} \cdot \frac{\partial \sigma}{\partial w})}{\sigma^{2n}}. \end{aligned} \qquad (31)$$

5. NUMERICAL SEARCH

Real optimization methods typically assume f is smooth, then use its derivatives to navigate around the space. In *line search*, a relatively well-understood iterative method, you choose a search direction, choose a step size, then repeat. In most cases the search direction should be a *descent direction*, or a direction in which the function is decreasing.

For simplicity of development, we have used the normalized gradient descent with backtracking, with the normalized gradient descent itself as our search direction.

As our error surface is likely non-convex (and thus has local minima) the iterated line search will only return one of several local minima. To get around this problem, we can start the search in different places to sample different local minima, known as randomized gradient descent.

5.1. Penalty Terms

To encourage less extreme parameter transformations when possible, we added penalty terms for the amount of transformation. While not always encouraged in terms of convergence properties, this kind of constraint has the added bonus of allowing to favor one type of transformation in the potential solutions over another.

6. BASIC EXPERIMENTS

6.1. Development Database

We assembled a small database of 10 sounds of different types of audio signals to test the predictors and basic optimization tests. The sounds were collected from Freesound [7] and included speech (adult, baby), sounds (dishes, mouth pop), musical instruments (harmonica, gong), several synthetic electronic beats, and environmental noise. A minority of the sounds were less than one second long, we truncated the longer ones to maximum duration one second before analysis.

6.2. Predictors

In the first experiment we attempted to predict a variety of simple time and fourier domain descriptors under either resampling or bandpass filtering. Results are discussed below.

6.3. Transformation by Descriptor

To test prediction and target-based optimization end-to-end, we used each of the sounds as an input sound, and likewise each of the sounds as a target, using the extracted spectral centroid and standard deviation, for a total of 10x10 trials, including the identity trials, to serve as an interesting sanity check. Free parameters for the experiment include constants for the parameter penalty terms, choice of sigmoid steepness α, and parameters of randomized line search (trials, starting distribution, step size, contraction rate, etc), all of which were chosen by hand.

Once parameters are chosen for each set of targets, we transform the input sound according to those parameters. Then we can measure the descriptors of the transformed sound and compare it to the original target descriptors. A set of input trials against a particular target is shown in figure 5.

The experiment gives us two forms of error, the model error, or the distance from the target after model optimization, which can explain the difficulty of search in the model error space, or the adequacy of a particular search method; and the transformation error, the real distance from the target of the transformed sound, which can explain potential errors of the model in describing the real transformation in descriptor space.

Centroid and std. deviation to describe the targets of the first transformation experiment, for which numerical results below are reported. Shortly after, using formulations in 4.1 we used a larger set of spectral moments to attempt a more general spectral warping.

7. PRELIMINARY RESULTS

In developing our models of resampling and equalization, we were able to predict the change in descriptors, starting over one sound, then generalizing to our small 10 sound database. We tested over a range of parameters and a set of descriptors to fair accuracy (around 10%), as shown by Figure 4.

On the same set of sounds used as targets to each of the input sounds, we optimized the transformation parameters to an average of ±45 Hz on the model.

When we actually used these parameters for transformation and testing, we got an average of about ±190 Hz in spectral centroid and ±140 Hz in standard deviation. This is a strictly numerical error and should probably be supplemented by perceptual measures in the future.

7.1. Optimization Efficiency

Simple penalty terms added to the model made the search take longer, but did return solutions with less extreme parameters. These added two more free parameters to the optimization, effectively creating a tradeoff between squared residual error, eq sharpness, and potential resampling rates.

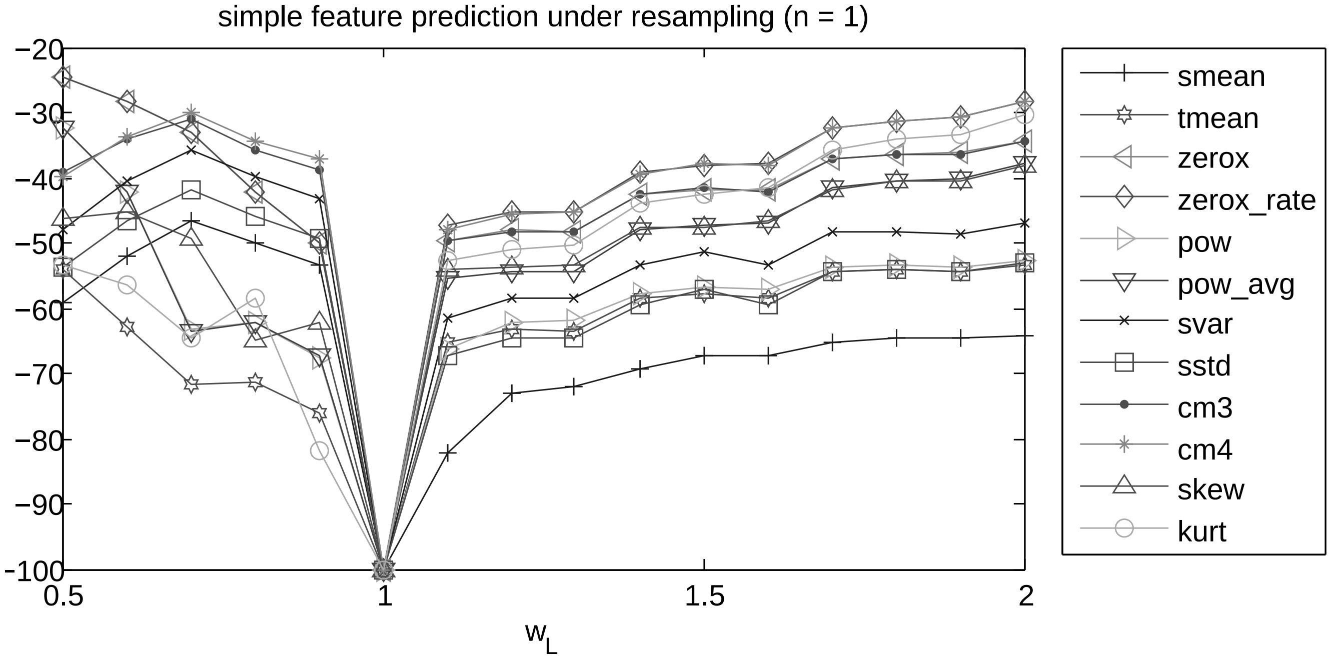

Figure 4: A sound is resampled at different rates, and descriptors such as spectral mean, variance, skewness, kurtosis, power, temporal centroid, etc are predicted given the input descriptors and the resampling rate w_L. Error is given as a ratio (predicted value over actual value) in the dB log domain; -20dB is equal to 10%. At $w_L = 1$ we have no transformation, thus no error.

7.2. Qualitative Analysis

After the optimization experiment, an investigator listened to the groups of transformed sounds to qualitatively evaluate the basic rendered result. By listening: within an input sound group, you get an idea of the range of transformation as a sound is transformed to hit different targets, and within a target group you see how different input sounds are transformed to hit the same target. An impression of similarity within the same target is present but not predominant. This would be due to many variations in the sounds that are not described by the two dimensions of spectral centroid and standard deviation, which are only a rough shape of the spectral distribution of a sound. One hopes that by adding other descriptors to the target, other spectral shape coefficients, temporal shape coefficients (along with transformations that effect them), and particularly descriptors what are strongly perceptually grounded, that the future synthesis results will be stronger for within-target similarity.

What we can confirm from listening is this: that combining transformations along with a penalty term can produce cooperation between them in reaching a target. Using either resampling or equalization, we can certainly formulate a more direct and efficient method of making an input sound like a target, for example computing the spectral envelope of a target and then adjusting the gains of the input directly to have the same envelope, but this solution can be characterized by its severity of transformation and its brittleness to subsequent transformations that may destroy the correspondence with the direct target.

8. FURTHER DISCUSSION

Optimization as a solution for this type of problem, when we wish to learn a function from descriptors to parameters or vice versa, has several clear disadvantages. First, numerical optimization is sensitive to the algorithm parameters, the shape of the function itself, and may take a long time to converge to what may be a local minima. Second, if we want to prelearn a mapping from descriptors to parameters, we cannot simply perform the optimization over a grid of descriptor targets, because solutions chosen by isolated optimizations may be in wildly different parts of the parameter space, so this mapping would not be smooth.

9. FUTURE WORK

At the outset, it seems we may have traded an inflexible signal manipulation model requiring detailed knowledge about the descriptors and transformations for an analytic model requiring the same knowledge and perhaps just as inflexible. Then, to proceed in building general TxD systems, we will need to use models able to accomodate more diverse sets of feature sets and transformations, while hopefully continuing to use all the knowledge from our current models.

10. ACKNOWLEDGMENTS

Thanks to MarC Vinyes for suggesting the use of sigmoids used for optimizing the transformation model. Thank you to both MarC and Mike for proofing and suggestions on the notation, and to Mark and Alex for additional comments.

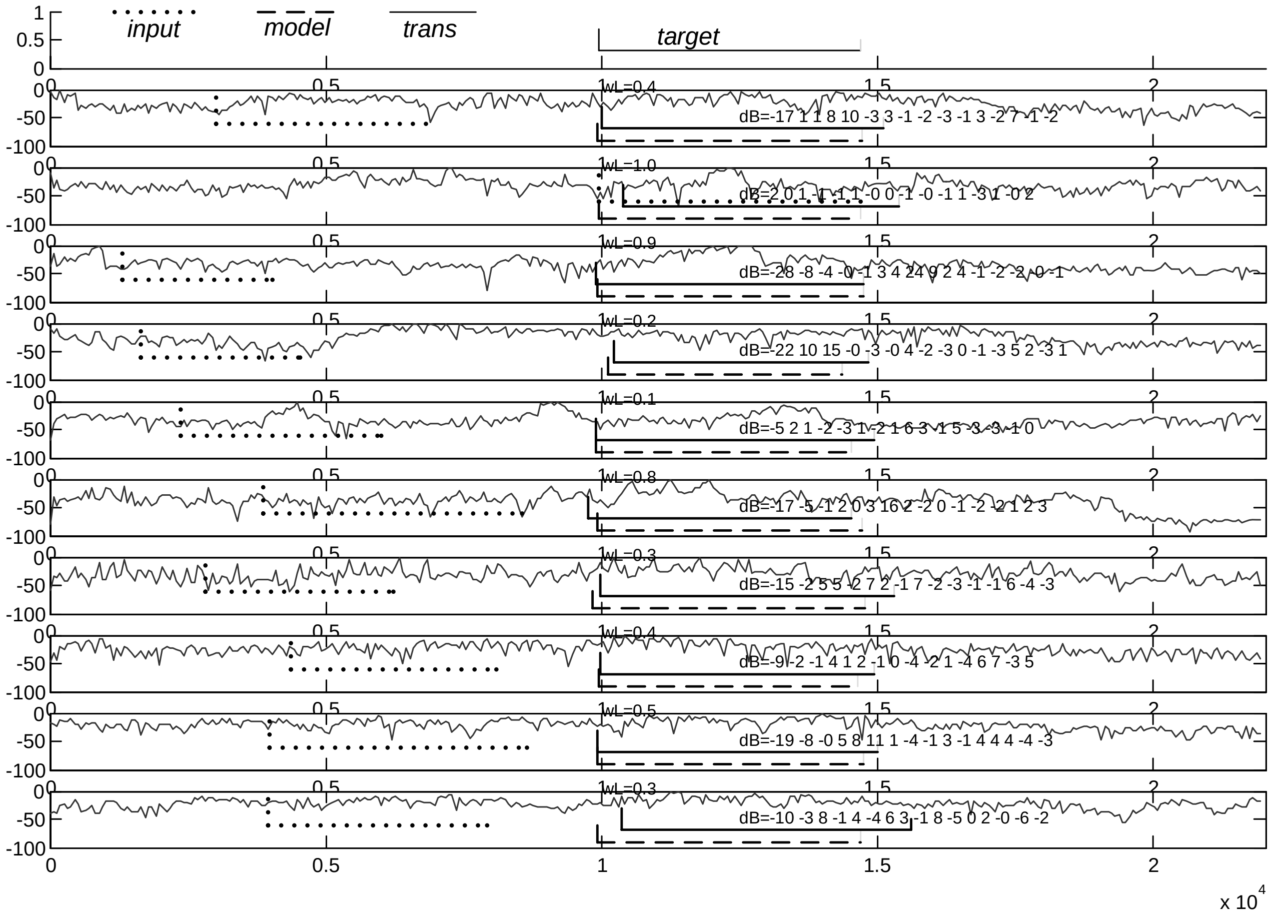

Figure 5: One of the targets from the preliminary transformation experiment. 10 input sounds are transformed to match a target, shown in the first row. Each bar represents a descriptor vector, where the left notch is the centroid in Hz, and the length is the std. dev. Descriptors for input sound are shown in the dotted bars, optimized model descriptors in the dashed bars, and descriptors after transformation in solid bars, against the transformed spectra in dB, with the transformation parameters w_L and w_j (in dB) overlayed on the sound.

11. REFERENCES

[1] V. Verfaille and P. Depalle, "Adaptive effects based on stft, using a source-filter model," *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, pp. 296–301, 2004.

[2] J. Bonada and Xavier Serra, "Synthesis of the singing voice by performance sampling and spectral models," *Signal Processing Magazine, IEEE*, vol. 24, pp. 67–79, 2007.

[3] Diemo Schwarz, "Corpus-based concatenative synthesis," *Signal Processing Magazine, IEEE*, vol. 24, pp. 92–104, 2007.

[4] Matthew Hoffman and Perry Cook, "The featsynth framework for feature-based synthesis: Design and applications," in *International Computer Music Conference*, 2007, vol. II, pp. 184–187.

[5] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second Edition*, http://ccrma.stanford.edu/~jos/mdft/.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.

[7] "Freesound," http://freesound.org/.