ANALYSIS OF ACOUSTIC FEATURES FOR AUTOMATED MULTI-TRACK MIXING

Jeffrey Scott, Youngmoo E. Kim
Music and Entertainment Technology Laboratory (MET-lab)
Electrical and Computer Engineering, Drexel University
{jjscott, ykim}@drexel.edu

ABSTRACT

The capability of the average person to generate digital music content has rapidly expanded over the past several decades. While the mechanics of creating a multi-track recording are relatively straightforward, using the available tools to create professional quality work requires substantial training and experience. We address one of the most fundamental processes to creating a finished product, namely determining the relative gain levels of each track to produce a final, mixed song. By modeling the time-varying mixing coefficients with a linear dynamical system, we train models that predict a weight vector for a given instrument using features extracted from the audio content of all of the tracks.

1. INTRODUCTION

Digital audio production tools have revolutionized the way we consume, produce and interact with music on a daily basis. Consumers have the ability to create quality recordings in a home studio with a relatively limited amount of equipment. Although there exists a myriad of complex software suites and audio editing environments, they all perform the same fundamental task of multi-track recording. This paper focuses on one of the most essential steps in music production: multi-track mixing. The relative levels between the various instruments in a song significantly determine the overall sonic quality of the piece.

In a previous paper we introduced a supervised machine learning approach for automatically mixing a set of unknown source tracks into a coherent, well-balanced instrument mixture using a small number of acoustic features [1]. We modeled the mixing coefficients as the hidden states of a linear dynamical system and used acoustic features extracted from the audio as the output of the model. After estimating the parameters of the model on the training data, we predicted the time-varying weights of each instrument for an unknown song using Kalman filtering [2].

We extend that approach in this paper by reducing the constraints on the model and generalizing it to a larger number of instruments. One modification to the system includes modeling the weights of an individual instrument and their first and second derivatives instead of jointly estimating the weights for all of the instrument tracks at once. This removes the restriction that the test song must contain all instrument types that the model was trained on.

Additionally, we explore an extended feature set within this framework and analyze the performance of each individual feature as well as combinations of features. The features are chosen to contain information about the total energy of the signal, energy within various frequency bands, spectral shape and dynamic spectral evolution.

2. BACKGROUND

Much research in the area of automatic audio signal mixing is devoted to applications in the context of a live performance or event. Initial research on the subject was oriented toward broadcast, live panel discussion and similar environments dealing with the human voice as the primary audio source [3]. These systems analyze the amplitude of the audio signal and apply adaptive gating and thresholding to each input signal to create a coherent sound source mixture of the individual tracks in addition to preventing feedback.

More recent work incorporates perceptual features (e.g, loudness) into systems designed for live automatic gain control and cross-adaptive equalization [4, 5]. The implementation of the former focuses on adapting the fader level of each channel with the goal of achieving the same average loudness per channel. The latter is designed for use in live settings as a tool for inexperienced users or to reduce equipment setup time. The system attempts to dynamically filter various frequency bands in each channel so that all channels are heard equally well.

Structured audio is the representation of sound content with semantic information or algorithmic models [6]. This

form of encoding allows for much higher data transmission rates as well as retrieval and manipulation of audio based on perceptual models. Currently, professional music post-production is performed by a highly skilled engineer with years of training. Using structured techniques, a parameterized, generative version of this process that is applicable to a variety of source audio is feasible.

More recent efforts focus on determining the parameters used in common linear signal processing effects such as equalization and reverb as well as dynamic level compression [7]. The authors also present a method for determining static fader values for an entire song for each track in a multi-track recording session. An interface for assisting users in creating mix-downs of user generated content from examples of mixes produced by professional engineers is presented in [8].

Other related work seeks to equalize an audio input based on a set of descriptive perceptual terms such as *bright* or *warm* [9]. Rather than attempt to navigate the complex network of sliders and knobs in an audio interface, a user can specify a high level term that describes the desired sound quality, and an appropriate equalization curve will be applied. The system was developed through collecting user ratings for audio examples and performing linear regression to find a weighting function for a particular instrument/timbre pair.

3. MODELING FRAMEWORK

The dataset we use in our experiments consists of 48 multi-track songs from the RockBand® video game. Each song contains both mono and stereo tracks for a basic rock instrumentation including guitar, bass, drums and vocals. Many songs may also include keyboards, horns, percussion, backing vocals, strings or other instruments. Often these backing instruments are contained in one audio track, making modeling each instrument separately rather difficult. To facilitate comparison between the data of each song, we first preprocess the tracks to obtain a set of five instrument tracks – bass, drums, guitar, vocals and a backup track that contains all other instruments. A detailed explanation of this process is given in [1].

3.1 Weight Estimation

Since we do not have the DAW sessions used to create each song, the actual fader values of the individual tracks are unknown and must be estimated. To do this, the digital audio output of the gaming console was recorded and aligned in a DAW session with the multi-track data of the corresponding song. The spectrum of a frame of the output mix is assumed to be a linear combination of the individual input tracks according to

$$\alpha_{1t}U_{1t} + \alpha_{2t}U_{2t} + \cdots + \alpha_{kt}U_{kt} = V_t \qquad (1)$$

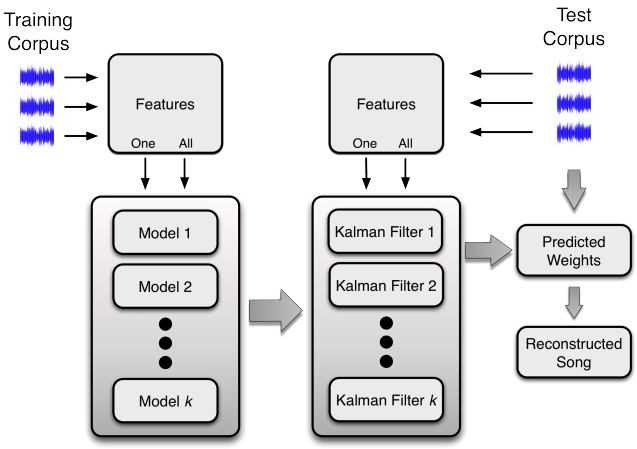

Figure 1. System diagram detailing the 'One Vs. All' method for mixing coefficient prediction.

where V_t is the spectrum of the mixed track and $U_{\{1,\ldots,k\}t}$ represents the spectra of the individual instrument tracks. We vectorize the spectrogram of each frame and use non-negative least squares (NNLS) to find the mixing coefficients. We use NNLS as opposed to unconstrained least squares estimation because multi-track mixing is an additive process.

The noise in the weights is reduced through Kalman smoothing [10]. It is significant to note that while these coefficients produce a mix that is perceptually similar to the original track, they are not the actual ground truth weights. Audio examples of the original song and the reconstructed mix using the estimated weights are available online [1] .

3.2 Weight Prediction

We use the weights estimated in Section 3.1 as labels in a supervised machine learning task. We first briefly outline the previous work we performed using this framework, then elaborate on a modified version of the model.

In [1] we treat the α values as the hidden states of a linear dynamical system and our acoustic features as the output of the system whose mathematical representation is

$$\boldsymbol{\alpha}_t = \mathbf{A}\boldsymbol{\alpha}_{t-1} + \mathbf{w}_t, \qquad (2)$$

$$\mathbf{y}_t = \mathbf{C}\boldsymbol{\alpha}_t + \mathbf{v}_t \qquad (3)$$

The dynamics matrix $\mathbf{A}$ controls the temporal evolution of the hidden states and $\mathbf{C}$ projects the hidden states into our observation space (feature domain). The driving and observation noise sources, $\mathbf{w}_t$ and $\mathbf{v}_t$, respectively are zero mean Gaussian random variables with covariances $\mathbf{Q}$ and $\mathbf{R}$.

[1] http://music.ece.drexel.edu/research/AutoMix

Track	All Tracks	One Vs. All	Best Features
backup	0.0126	0.0110	0.0087
bass	0.0191	0.0163	0.0088
drums	0.1452	0.1283	0.0489
guitar	0.0158	0.0151	0.0115
vocal	0.0188	0.0160	0.0108

Table 1. Results for LOOCV on the database. The MSE for each track across all songs is shown for the All Tracks method and the One Versus All approach.The Best Features column is the result from sequential feature selection.

Our state vector is the weights of each instrument at time step t

$$\boldsymbol{\alpha}_t = [\alpha_1 \alpha_2 \dots \alpha_k]^T \tag{4}$$

and the structure of the output vector is

$$\mathbf{y}_t = \left[F_1^{(1)} \dots F_m^{(1)}\ F_1^{(2)} \dots F_m^{(2)}\ F_1^{(k)} \dots F_m^{(k)}\right]^T \tag{5}$$

where k indexes the instrument and m is the feature index.

To train the model we estimate $\mathbf{A}$ and $\mathbf{C}$ through constraint generation and least squares, respectively and compute the covariances $\mathbf{Q}$ and $\mathbf{R}$ from the residuals of $\mathbf{A}$ and $\mathbf{C}$ [11]. In this framework, we are constrained in terms of the number and type of instruments we can use the automatic mixing system for. Since each α_k is associated with a specific instrument, omitting or adding tracks changes the dimension of the hidden state vector and in turn makes predicting weights for a set of tracks that are not explicitly in the form described in (4) and (5) intractable.

3.3 Modified Prediction Scheme

Instead of modeling the time varying mixing coefficients of all tracks as the hidden states of the LDS, we consider only one instrument at a time. Our new state vector consists of the weight for the jth track and its first and second derivatives

$$\boldsymbol{\alpha}_t = \left[\alpha_j\ \dot{\alpha_j}\ \ddot{\alpha_j}\right]^T \tag{6}$$

The derivatives of the weight vector are used to provide the model with more information about the dynamic evolution of the mixing coefficients. Note that only the weights for one instrument are included in the state vector. By eliminating the weight values of the other instruments, we are training the model to consider only how well the current instrument 'sits' in the mix, not how the weights of all instruments evolve together.

The output vector $\mathbf{y}_t$ is comprised of the feature set for the instrument we are trying to predict stacked with the average of the features from all other instruments

Feature	**Description**
RMS energy	Root mean square energy
Spectral flux	Change in spectral energy
Spectral bandwidth	Range of frequencies where most energy lies
Octave-based sub-bands	Energy in octave spaced frequency bands
MFCC	Mel-Frequency Cepstral Coefficients
Spectral centroid	Mean or center of gravity of the spectrum
Spectral peaks	Energy around a local sub-band maxima
Spectral valleys	Energy around a local sub-band minima
Slope/Intercept	Parameters of a line fit to the spectrum of a frame

Table 2. Spectral and time domain features used in mixing coefficient prediction task.

$$\mathbf{y}_t = \left[F_1^{(j)} \dots F_m^{(j)}\ \tfrac{1}{K-1}\textstyle\sum_{k\neq j}^{K} F_2^{(k)} \dots \tfrac{1}{K-1}\sum_{k\neq j}^{K} F_m^{(k)}\right]^T \tag{7}$$

If $j = 1$, then we are using m features associated with the first track and averaging the features associated with the tracks $k \neq j$, reducing the dimensionality of the feature vector from km to $2m$. Comparing (5) to (7), we observe that in (7) there is no dependency on which position (k) the features for a given instrument are located. The only prior knowledge the model requires is the type of the jth instrument for which we are predicting time-varying weights. As a result, in this framework there is no limitation on the number or type of instruments that can be mixed using the system, provided that there exists training data for the target instrument j. A system diagram showing the new modeling method is shown in Figure 1.

To evaluate the efficacy of this modified estimation approach, we perform the same experiment outlined in [1] and compare the results of the two methods. Using the 48 songs in our dataset, we perform leave-one-out cross-validation (LOOCV), training an LDS on 47 tracks and predicting the weights for the remaining track. We repeat the process using each track as a test song only once and average the mean squared error (MSE) between our estimated ground truth values and our predictions from the LDS. The results are shown in Table 1. We refer to the method described in Section 3.2 as All Tracks (AT) and the modified approach in this section as One Versus All (OVA). The OVA results are are computed using the same feature set {*centroid, RMS, slope, intercept*} that was used in the previous experiment [1].

The table shows an average improvement of 11.66% in terms of MSE for all instrument types in the dataset. The OVA method provides increased performance in terms of the MSE of the weight predictions as well as increased flexibility. The new topology enables the system to mix songs that do not have the same number of tracks as the normalized RockBand dataset we compiled.

Backup		Bass		Drums		Guitar		Vocal	
Feature	**Error**	**Feature**	**Error**	**Feature**	**Error**	**Feature**	**Error**	**Feature**	**Error**
Bandwidth	0.0511	**Flux**	0.0590	**Centroid**	0.7322	**Bandwidth**	0.0756	**Flux**	0.1183
Flux	0.0526	**Bandwidth**	0.0590	**RMS**	0.8415	**Valley**	0.0878	**Centroid**	0.1240
Sub-Bands	0.0580	Slope	0.0618	**Slope**	0.8713	**Intercept**	0.0908	Bandwidth	0.1251
Intercept	0.0587	**Intercept**	0.0622	Bandwidth	0.8861	Slope	0.0920	Valley	0.1262
Slope	0.0589	RMS	0.0716	**Intercept**	0.8932	Flux	0.0936	Peak	0.1302
Peak	0.0607	**Valley**	0.0741	Peak	0.9260	Sub-Bands	0.0974	Intercept	0.1316
RMS	0.0629	Sub-Bands	0.0743	Valley	0.9381	RMS	0.0987	**Sub-Bands**	0.1317
Centroid	0.0636	Peak	0.0752	**Sub-Bands**	0.9649	**Peak**	0.1019	**Slope**	0.1318
MFCC	0.0659	Centroid	0.0801	MFCC	1.1785	Centroid	0.1095	RMS	0.1320
Valley	0.0680	**MFCC**	0.0821	Flux	3.5767	**MFCC**	0.1127	**MFCC**	0.1373

Table 3. Mean squared error for all features and individual instruments. Features for each instrument are listed in order of best performance to worst performance. The best combination of features for each instrument is in boldface.

4. FEATURE ANALYSIS

Having shown that the OVA method outperforms the AT method, we proceed to investigate which features are the most informative. We explore an extended feature set within the framework described in the previous section and analyze the performance of each individual feature as well as combinations of features. Table 2 lists the array of spectral and time domain features we selected for our experiment [12–14]. The features are chosen to contain information about the total energy of the signal, energy within various frequency bands, spectral shape and dynamic spectral evolution. All experiments are performed using LOOCV on the entire dataset. In the first experiment, we test the performance of each individual feature using the average MSE over all songs as our error metric. Table 3 shows the results for each feature for each track type in the dataset. There is no single feature that appears to be dominant for mixing coefficient prediction.

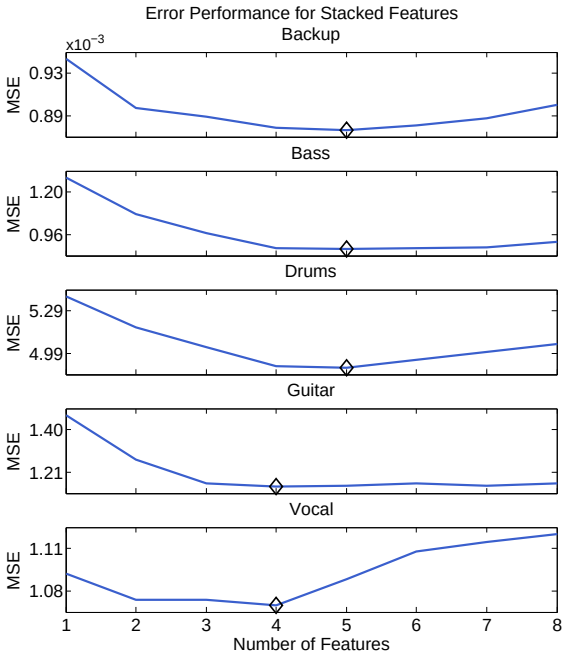

Figure 2. MSE versus the number of stacked features used in training an LDS for each track. Note that the scale of each sub-plot varies. The minimum is indicated for each track.

Using these results, we employ sequential feature selection to increase the performance of our system [15]. The best performing feature for each instrument in Table 3 is stacked with each remaining feature, and the MSE for LOOCV is computed for each combination. The best feature from this result is retained and the process is repeated until all features have been used. The results of this analysis are depicted in Figure 2. The best performing number of features for each instrument is indicated with a diamond. Since some of our features may contain similar information, adding additional features eventually becomes redundant and the increase in the size of the parameter space outweighs the gain in information.

5. RESULTS

The overall results for using the best performing feature ensemble are detailed in Table 1. The table shows that the OVA approach more accurately models the mixing coefficients and the addition of more features greatly improves the results. Mean squared error does not provide any intuition about where each model fails or performs well. Figure 3 shows a comparison between the AT and OVA models. Both

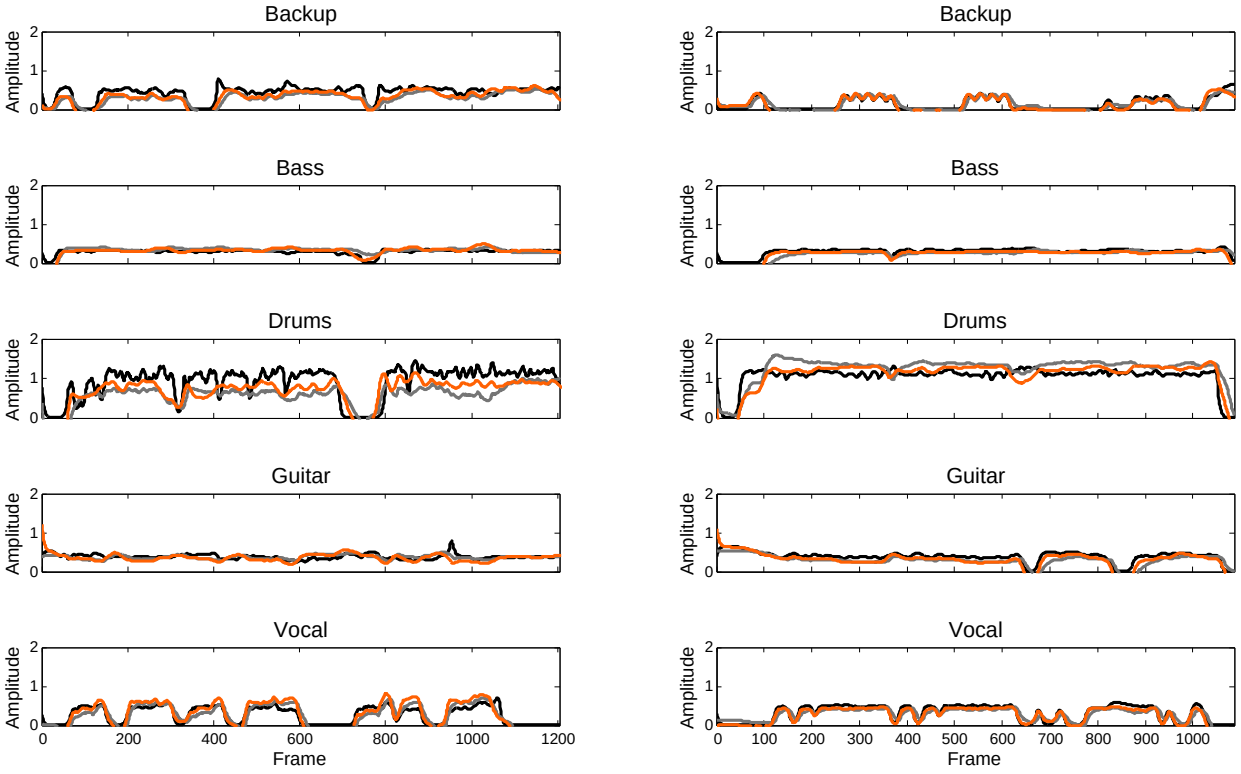

Figure 3. Comparison of ground truth (black) values with AT (gray) and OVA (orange) models. Left: 'More Than A Feeling' by Boston. Right: 'Hammerhead' by The Offspring.

models were trained with the feature set used in [1]. There is relatively small deviation in the bass and guitar predictions for each method on both songs. The most significant difference is in the ability of the OVA model to track the vocal weights as evidenced by the relatively flat predictions from the AT model contrasted with the OVA model predictions that follow the contour of the ground truth weights.

In Figure 4 we observe the effect of increasing the number of features used to train the model. The predictions using the best feature for each instrument from Table 3 are shown in gray and the highest performing ensemble of features is depicted in orange. Adding features creates the most improvement in the drum track where the contour and bias of the predictions closely follows the ground truth for both songs. Although this is only a small sample of the dataset, this representation informs us of improvements that can be made to the system.

6. CONCLUSION

Our automatic multi-track mixing system predicts a set of weighting coefficients for an instrument given an ensemble of acoustic features extracted from audio content. We improve upon our previous modeling framework by training a separate LDS for each instrument rather than modeling all weight vectors within a single system. Applying the One Versus All method of training removes the restrictions imposed by the All Tracks model and yields better performance in predicting the weights for all instruments.

Moreover, we investigate the accuracy of an array of spectral and time-domain features on predicting the mixing coefficients. The improved modeling scheme and feature ensemble chosen through sequential feature selection illustrate marked improvement over our previous results. While this approach to automatic multi-track mixing works well for our small dataset, in the future we plan to develop a larger and more varied corpus of songs to explore how robust the model is.

7. ACKNOWLEDGMENT

This work is supported by National Science Foundation award IIS-0644151.

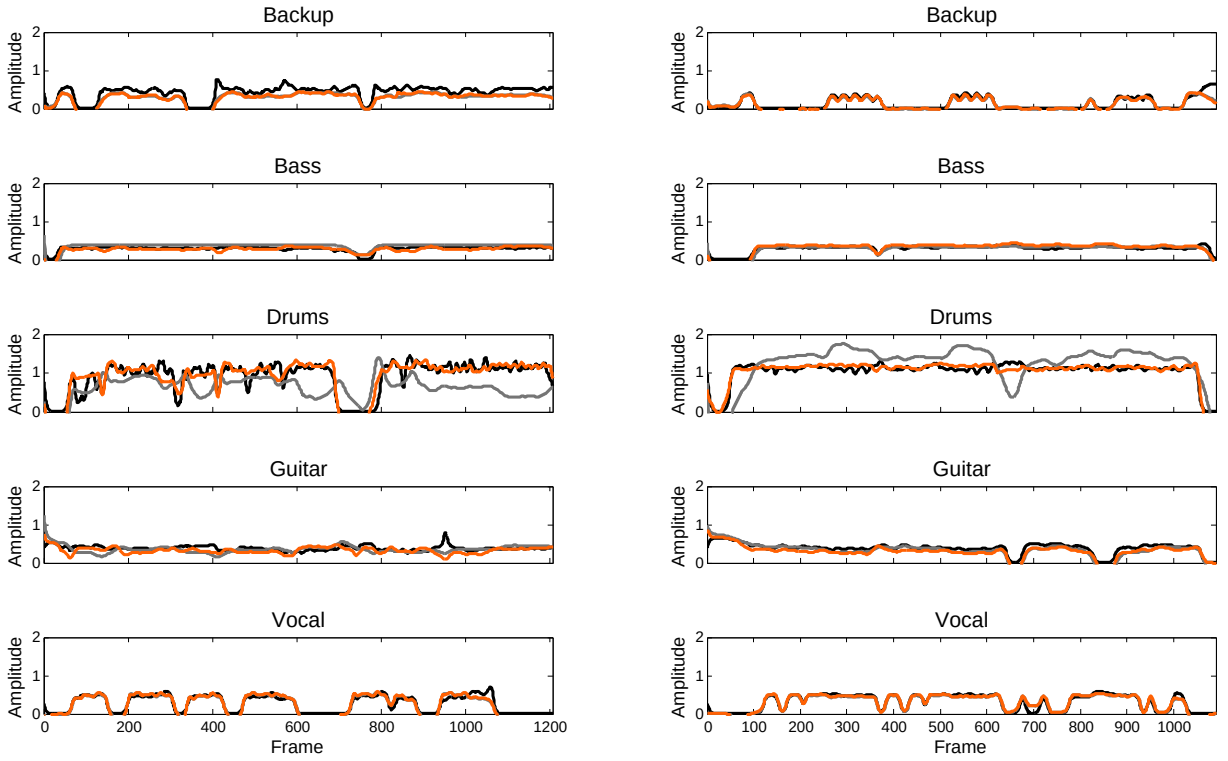

Figure 4. Comparison of ground truth (black) values with OVA model using the single best feature (gray) and using the best combination of features (orange). Left: 'More Than A Feeling' by Boston. Right: 'Hammerhead' by The Offspring.

8. REFERENCES

[1] J. Scott, M. Prockup, E. M. Schmidt, and Y. E. Kim, "Automatic multi-track mixing using linear dynamical systems," in *Proceedings of the 8th Sound and Music Computing Conference*, Padova, Italy, 2011.

[2] E. M. Schmidt and Y. E. Kim, "Prediction of time-varying musical mood distributions using kalman filtering," in *Proceedings of the 2010 IEEE International Conference on Machine Learning and Applications*, Washington D. C., USA, 2010.

[3] D. Dugan, "Automatic microphone mixing," *J. Audio Eng. Soc*, vol. 23, no. 6, pp. 442–449, 1975.

[4] E. Perez Gonzalez and J. D. Reiss, "Automatic gain and fader control for live mixing," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 1–4.

[5] ——, "Automatic equalization of multichannel audio using cross-adaptive methods," in *127th AES Convention*, 2009.

[6] B. Vercoe, W. Gardner, and E. Scheirer, "Structured audio: Creation, transmission, and rendering of parametric sound representations," in *Proceedings of the IEEE*, 1998, pp. 922–940.

[7] D. Barchiesi and J. Reiss, "Reverse engineering of a mix," *Journal of the Audio Engineering Society*, vol. 58, no. 7, pp. 563–576, 2010.

[8] H. Katayose, A. Yatsui, and M. Goto, "A mix-down assistant interface with reuse of examples," in *First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, Florence, Italy, 2005.

[9] A. T. Sabin and B. Pardo, "A method for rapid personalization of audio equalization parameters," *Proceedings of ACM Multimedia*, pp. 769–772, 2009.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[11] S. Siddiqi, B. Boots, and G. Gordon, "A constraint generation approach to learning stable linear dynamical systems," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2008, pp. 1329–1336.

[12] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Lusanne, Switzerland, 2002, pp. 113–116.

[13] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357 – 366, aug 1980.

[14] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293 – 302, jul 2002.

[15] L. Mion and G. D. Poli, "Score-independent audio features for description of music expression," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 2, pp. 458–466, 2008.