Audio Metaphor 2.0: An Improved Classification and Segmentation Pipeline for Generative Sound Design Systems

Joshua Kranabetter[1], Craig Carpenter[1], Renaud Bougueng Tchemeube[2], Philippe Pasquier[2], and Miles Thorogood[1]

[1]University of British Columbia
[2]Simon Fraser University

ABSTRACT

Soundscape composition and design is the creative practice of processing and combining sound recordings to evoke auditory associations and memories within a listener. We present a new set of classification and segmentation algorithms as part of Audio Metaphor (AUME), a generative system for creating novel soundscape compositions. Audio Metaphor processes natural language queries from a user to retrieve semantically linked sound recordings from a database containing 395,541 audio files. Building off previous work, we implemented a new audio feature extractor and conducted experiments to test the accuracy of the updated system. We then classified audio files based on general soundscape composition categories, improved emotion prediction, and refined our segmentation algorithm. The model maintains a good accuracy in segment classification, and we significantly improved valence and arousal prediction models - as noted by the r-squared (72.2% and 92.0%) and mean squared error values (0.09 and 0.03) in valence and arousal respectively. An empirical analysis, among other improvements, finds that the new system provides better segmentation results.

1. INTRODUCTION

Soundscape composers aim at creating a type of electroacoustic music that is ”characterized by the presence of recognizable environmental sounds and contexts, the purpose being to evoke listeners associations, memories, and imagination related to the soundscape” [1]. Figure 1 demonstrates the relationships of these sound design contexts on the continuum moving from realistic to abstracted. Computationally assistive tools for sound design and soundscape composition production focus on soundscapes trending toward the real end of this continuum.

Recent advancements of our Audio Metaphor (AUME) generative audio model expand the system’s ability to produce sonic experiences with depth and clarity. As sound designers and soundscape composers use creative and technical strategies to communicate the sense of a place as perceived by a listener, an important factor of a soundscape or sound design includes valence and arousal in people’s perception. Russel introduces these factors in a circumplex model that facilitates evaluation and analysis of a stimulus such as sound [2]. AUME expands on previous work in soundscape emotion recognition with a unique interface to modulate valence and arousal.

The system’s most recent iteration draws on a database of 395,541 hand-tagged files. Updated segment description algorithms improve emotion recognition significantly while maintaining classification accuracy. Background-foreground segmentation and composition processes are automated through simple natural language queries in AUME. A smoothing algorithm now produces superior boundary delineation of the emotive audio signal while maintaining temporal resolution. This level of automation affords sound designers the ability to create long, complex soundscapes with a simple text query. These improvements, combined with the interactive speed of AUME, can provide state-of-the-art solutions in sound design and soundscape composition.

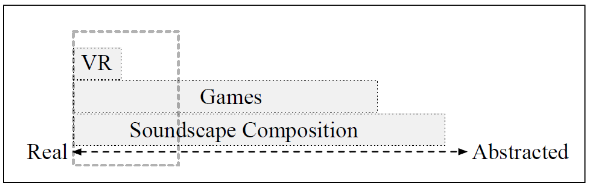

Figure 1. Continuum of ambience focus from real sounding environments, to more abstracted spaces. Production contexts range across the continuum. Figure from Thorogood et al. [3].

2. RELATED WORK

Generative soundscape models objective is to reduce the time-consuming practice in sound design of searching, importing, editing, sequencing, and arranging audio files from often unwieldy, massive online audio databases. Birchfield et al. describe a system that uses an adaptive user model for context-aware soundscape composition [4]. In their work, the system has a small set of hand-selected and hand-labeled audio recordings that were autonomously mixed with minimal processing. Similarly, Eigenfeldt and Pasquier employ a set of hand-selected and hand-labeled environmental sound recordings for the retrieval of sounds from a database by autonomous software agents [5]. In their work, agents analyze audio when selecting sounds to mix based on low-level audio features.

Like AUME, Teixeira et al. demonstrate a model that bypasses the tagging step through automating the browsing and retrieval of audio from large databases by employing Russel's widely used model of valence and arousal [6]. The MScaper system [6], integrated into Ableton Live DAW, similarly supports the adoption of the valence and arousal model for retrieving soundscapes. The valence and arousal model to facilitate soundscape generation also shares synergies with automated feature extraction developed by Music Information Retrieval systems (MIR). Bellisario and Pijanowski explore contributions of MIR to the emergent field of soundscape ecology and demonstrate how methodologies of automated feature extraction, sound classification and labeling using machine learning, and data visualization might be extended to soundscapes [7]. In further studies, Teixeira et al. have mapped how MScaper performs by using crowd-sourced affective annotations from the Emo-Soundscapes dataset and mapped affective dimensions and low-level audio descriptors [6]. MScaper supports audio database navigation through a valence and arousal model, generating an adaptive soundscape according to emotional states. While AUME similarly employs a valence and arousal model, its unique composition engine utilizes a combination of Natural Language Processing (NLP), background and foreground classification, and segmentation to set it apart from related models in soundscape generation.

3. PREVIOUS WORK

Audio Metaphor (AUME) is a system for creating computer-assisted creation of soundscape compositions. Before this system update, the system performed better than random baseline in pleasantness, eventfulness, believability, and semantics [3]. The authors demonstrated AUME in the form of an art installation that derived natural language queries from Twitter. The queries were then used to generate soundscapes to reflect trending events in the social landscape. In previous research on Audio Metaphor, we established a framework for the system [8] (presented in Figure 2) and continue to describe each element throughout this section.

3.1 Corpus

The audio corpus consists of 395,541 sourced sound files from which we generate soundscapes. We save the audio files from source to disk to create the database. As described in the following sections, segmentation is required to obtain the short audio segments used for the automatic soundscape composition.

3.2 Crawler

The crawler is responsible for building the AUME database. It implements our feature extraction, classification, emotion recognition, and segmentation on a collection of audio files. The crawler finds an audio file, runs the pipeline, then saves the resulting segments into the AUME database. Each data entry in our database contains a file path to the raw MP3 audio data, associated file tags, and segment data. This segmentation and classification section, shown at the top of Figure 2, only runs once to build the AUME database. After the crawler retrieves an audio file, the first job in the pipeline is feature extraction.

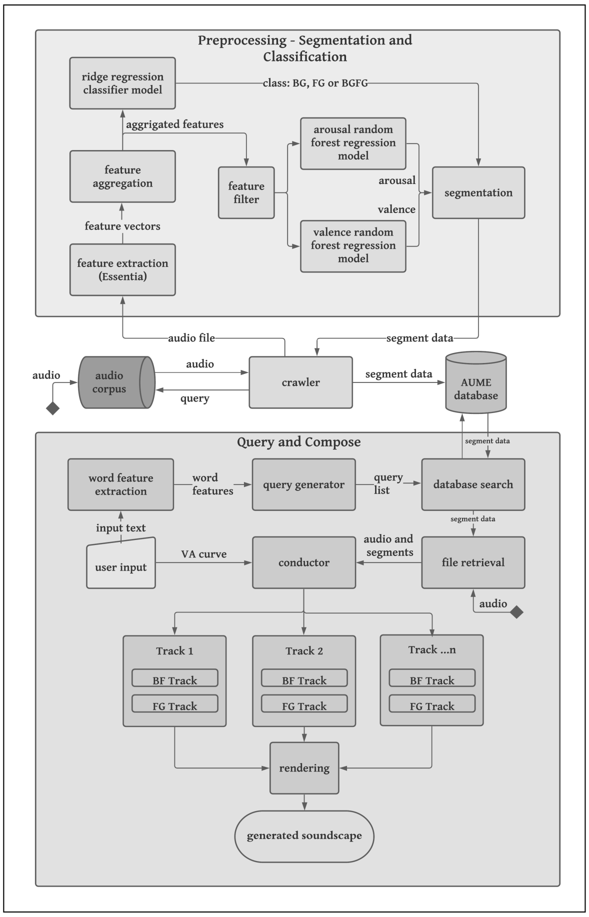

Figure 2. System diagram. Background, foreground, and background with foreground as BG, FG, and BGFG respectively.

3.3 Feature Extraction

The audio feature set we generate in feature extraction contains spectral and perceptual audio descriptors of high and low levels. These descriptors attempt to model the human auditory system. This is desirable from a soundscape studies perspective, where the perception of the human listener is an important consideration. These features lay the foundation for the new classifier and emotion prediction models in Sections 7 and 8 respectively.

3.4 Feature Aggregation

We aggregate features using a bag of frames approach, where a signal is represented as a statistical distribution. The extractor provides feature data based on the input signal. We then divide that feature data into frames from which we can calculate the mean, standard deviation, skewness, variance, and their respective derivatives, and second derivatives. Frames do not sit end to end but overlap 50% with previous and following frames. To represent windows in an audio file, we group the output statistical data into segments. The segments can then be individually analyzed by our classification and emotion prediction algorithms.

3.5 Audio File Classification

An audio recording can be divided into the general classes of background, foreground, and background with foreground sounds. Sound designers and soundscape composers manually segment audio files into building blocks for use in a composition. We use machine learning to classify segments in an audio file automatically. As shown in Figure 2, the classifier uses extracted features to classify each segment in the audio file as foreground, background, or background with foreground. In previous work, our support vector machine (SVM) classifier achieved a true positive rate of 87.77%, a false positive rate of 12.22%, and a Kappa interrater reliability statistic of 0.8167. Audio file classifications are saved in the database to be used as the backbone in composition by the conductor. Next, we improve our composition further by adding emotion metrics for each segment.

3.6 Emotion Prediction

With the ability to interpolate Russel's circumplex model shown in figure 3, AUME retrieves audio segments evaluated on a scale of valence and arousal. Russel's model suggests all emotions are distributed in a circular space. High levels of valence correspond to pleasant sounds while low valence levels correspond to unpleasant sounds. Further, high levels of arousal correspond to exciting sounds while low levels correspond to calming sounds. Sound designers evoke emotion from a listener by dynamically controlling valence and arousal levels throughout a composition. We quantify levels of valence and arousal using machine learning for emotion prediction. The emotion prediction models use a subset of extracted features to predict valence and arousal for each segment in an audio file. We store the results in the AUME database for use in composition, a process we further explain in section 3.9. Previously, our emotion prediction models accounted for 62.9% and 85.5% for valence and arousal, respectively.

3.7 Segmentation

In segmentation, we aim to group background-foreground classified segments perceived as belonging to the same class. In the pipeline 2, segmentation is the final step before adding our segment data to the AUME database. We use segment information and background, foreground labels to filter, then conjoin like classes. We use a filtering algorithm to remove the noise of misclassified segments and negligible class appearances between two segments of the same class. The consequence of filtering is a resolution loss where detail can be reduced. In contrast, filtering can provide greater continuity from increased segment length, and therefore, more natural-sounding compositions. After we use the filtering algorithm, we finalize the segments by grouping adjacent segments belonging to the same class. The crawler finishes the audio prediction side of AUME by saving segment information into the database.

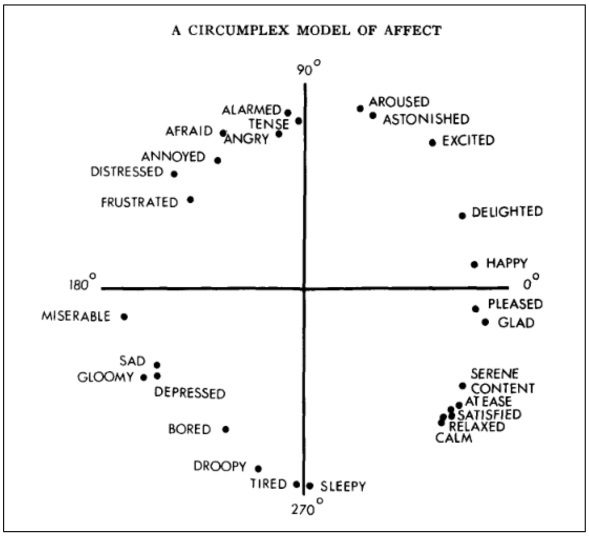

Figure 3. Common emotions displayed on the circumplex model of affect presented by Russel [2]. Arousal (eventfulness) occupies the vertical axis while valence (pleasantness) occupies the x axis.

3.8 Audio File Retrieval Using Natural Language Processing

The Audio Metaphor project uses an algorithm called SLiCE to process user input query text [8]. In the Natural Language Processing pipeline, common words listed in the Oxford English Dictionary Corpus are removed from the query, leaving only nouns, verbs, and adjectives. Words are kept in order and treated as a list. For example, with the word feature list from the natural language query: ”The angry dog bit the crying man,” ”angry dog bit crying man,” is more valid than ”angry man bit crying dog.” For a user input text with *n* words, the SLiCE algorithm constructs sublists of words of size *1* to *n* from the user input text to maximize the ability to match the appropriate sound segments. All unique sublists are put in a queue and used as search queries, starting with the longest first. When a search query returns a satisfactory result, all remaining queries that contain any of the successful word features are removed from the queue. The aim of the algorithm is to minimize the number of audio files returned and still represent all the word features in the list.

3.9 Conductor

The conductor is responsible for composing soundscapes using the audio supplied by the user query and associated class and emotion data. When a user queries the system, they can specify curves for valence and arousal to achieve a soundscape with their desired emotional characteristics. The conductor arranges segments that result from the search according to the specified curves. Next, the conductor mixes the tracks and renders the soundscape. Finally, AUME presents the resulting computationally generated soundscape to the user.

4. BUILDING A MODERN PIPELINE

To further research and develop Audio Metaphor, we move towards a state-of-the-art system to replace the YAAFE [9] audio feature extractor with Essentia [10]. Actively maintained by The Music Technology Group (MTG) of the University Pompeu Fabra in Barcelona, Essentia offers a more significant number of audio features. We expect Essentia to represent audio files with a higher degree of explanatory power and offer an opportunity to improve our predictive models - a process we explore in sections 7 and 8. Further, we investigate segmentation algorithms and their effects to tune the system for the best practical results.

5. CORPUS

The database for this project is a corpus of 395,541 audio files, a 96.8 fold increase from the original database of 4085 files. We extract files from the Freesound.org project using the Freesound API. The audio files from the Freesound project are uploaded in various formats (MP3, WAV, AIFF) and variable sample rates (130kbps to 196kbps) by individual users. They are accompanied by a collection of textual tags that describe the audio content's nature. The Freesound project processes the uploaded audio content to create normalized MP3 versions and metadata of sonic analysis. We extract audio files of duration ranging from two seconds up to 10 minutes. We do not use Freesound's sonic analysis as we tailor our own feature extraction to meet our specific needs.

6. FEATURE EXTRACTION

Essentia is an open-source C++/Python library for audio and music analysis. We use Essentia to sample corpus audio at full 22050Hz AIF format. In our bag of frames approach, we apply a frame size of 2048 samples and an analysis step of 1024 samples. We use the Blackman-Harris windowing function for spectral features. We use various statistics mentioned in section 3.4 to aggregate the features. To see all specific features we extract, see the Essentia documentation under the categories low-level stats and tonal stats [10]. This windowing configuration and subsequent analysis step result in high descriptive power for representing the texture and overall dynamics of the sound. Since we achieved good results with this method, we did not explore other window configurations.

7. SUPERVISED CLASSIFIER FOR SEGMENTATION

7.1 Corpus

We use the corpus created by Thorogood et al. [11], a curated collection of BF labeled sound files from the World Soundscape Project Tape Library database (WSPTL) [12]. The WSPTL contains five unique collections of soundscape recordings, with a total of 2545 individual sound files amounting to over 223 hours of high-quality, carefully selected recordings. The collections gathered between 1972 and 2010 are comprised of recordings from across Canada and Europe. Recording equipment included a Nagra IV-S field recorder and a pair of AKG condenser microphones. Collections are digitized and held online at Simon Fraser University [13].

The corpus is composed of 200 4-second samples from the WSPTL. Independent listeners confirmed 4-seconds was sufficient length for identifying the context of the sound. Further, the corpus is compact, so participants finished the study with minimum listening fatigue. Additionally, samples are short to preserve their class homogeneity for machine learning. The types of sounds cover the six soundscape categories defined by Schafer: natural sounds, human sounds, sounds and society, mechanical sounds, quiet and silence, and sounds as indicators. [14]. We mix the audio down to mono in favor of a higher degree of generality of the system. This compensates for recordings not obtained with similar high precision equipment or those recorded in mono.

Figure 4 shows the study's category agreement for the top 30 most agreed upon sounds by the study participants for each category, a data subset we call the BF90. A quantitative analysis of responses against the final corpus shows that participants agree on the category assigned to 92.5% (SD=3.6%) of the background samples, 80.8% (SD=9.5%) of foreground samples, and 75.3% (SD= 11.3%) of the background with foreground.

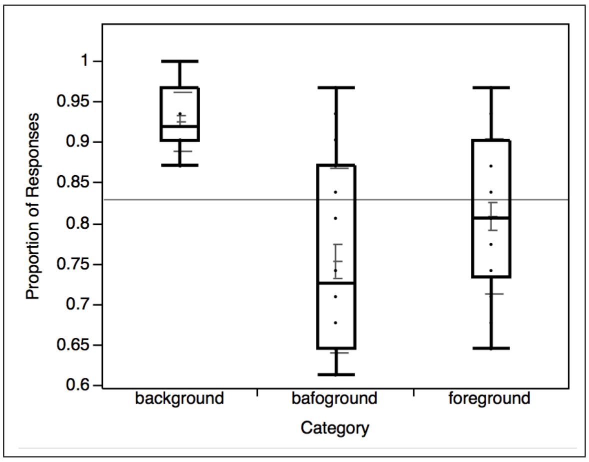

Figure 4. Box plots and mean lines for the agreement of labels for the corpus of background, foreground, and background with foreground recordings. The light grey line represents the overall mean agreement for the three classes. Figure from Thorogood et al. [11].

7.2 Model

Ridge regression is a type of regularized linear regression that incorporates techniques to reduce model complexity and prevent overfitting. In ridge regression, L2 regularization augments the cost function by adding a penalty equivalent to the square of the magnitude of the regression coefficients. This method performs well in cases where variables are highly correlated or the number of features exceeds the number of data points. We use the Scikit Learn [15] implementation of the ridge regression classifier which converts binary targets to -1, +1, then performs multi-output regression. The predicted class is the output with the highest value. We use the default regularization strength of one.

7.3 Method

We perform an evaluation of the BF-Classifier using a repeated 10-fold cross-validation strategy on the BF90 dataset. This method randomly partitions the validation set into k = 10 equally sized sub-samples before iteratively testing the remaining sub-samples against each k-partition. We repeat the evaluation 10 times to reduce noise in the 10-fold evaluation.

7.4 Evaluation

Model prediction results are shown in Table 1 and a summary is shown in Table 2. The classifier achieves an overall true positive rate of 83.0%. An inter-rater reliability analysis using the kappa statistic determines the consistency of the classification. In this case, the kappa statistic of 0.743 shows good reliability of the classification results over the 10-fold validation.

	BG	FG	BGFG
BG	**23**	2	5
FG	0	**29**	1
BGFG	5	2	**23**

Table 1. Confusion matrix of SVM classifier for the categories background (BG), foreground (FG), and background with foreground (BGFG).

True Positive	83.0%
False Positive	9.30%
Kappa statistic	0.743

Table 2. Average true positive, false positive, and Kappa statistics of the ridge regression classifier.

8. REGRESSION FOR EMOTION PREDICTION

8.1 Corpus

We use the Emo-Soundscapes dataset for emotion recognition [16]. The dataset consists of 1213, 6-second long monophonic audio clips. Fan et al. curated 600 sounds from Freesound.org and mixed 613 audio clips from a combination of these. Additionally, the dataset contains a ranking for the perceived emotion in the 2D valence arousal space. They source rankings from 1182 trusted annotators from 74 different countries. The 1182 trusted annotators had a gold standard of 92.18% accuracy and provided a total of 69477 pairwise comparisons. We use the provided ratings, a translation of the rankings from 1 to 1213 mapped to a linear space between 1 to -1 inclusive.

8.2 Model

We use the Scikit Learn [15] implementation of random forest regression in prediction for both Valence and Arousal. It is a supervised learning technique that uses an ensemble method for classification and regression. Ensemble learning methods combine multiple algorithms to produce superior robustness than any of the component algorithms alone. In the case of the random forest, bootstrap aggregation (bagging) on each tree reduces the high variance of the decision tree. The result is a flexible model with reduced susceptibility to overfitting compared to a decision tree alone. It shows superior results compared to the support vector regression model provided by the emo-soundscapes data set and comparable results to deep learning methods [17].

8.3 Method

We randomly split the dataset into training and holdout sets; 80% for the training set and the remaining 20% for the holdout set. We perform feature selection to improve computational costs and improve model performance. We use 10-fold cross-validated recursive feature elimination on the training set. Most features were eliminated, with a 604 to 72 and 604 to 67 feature reduction in arousal and valence, respectively. Next, we tune hyper-parameters using five-fold cross-validation on the training set utilizing a range of values for maximum depth, minimum leaf samples, and minimum samples split. Finally, we perform the final evaluation on the holdout set.

8.4 Evaluation

We use Mean Squared Error (MSE) and R^2 to evaluate the performance of the models. Results expressed in Table 3 prove superior models when compared to previous research. The most recent previous model results using support vector machines (SVM) demonstrated 85.5% and 62.9% for arousal and valence, respectively [16], a significant improvement on previous research [18]. Our new model accounts for 92.0% and 72.2% of the variance for arousal and valence, respectively. Compared to deep learning methods, our random forest regression model achieves similar performance with a 2.8% improvement in arousal prediction and a 3.75% performance reduction in valence prediction. MSE values have also improved when compared to the most recent SVM models and are comparable to deep learning models. MSE of the arousal model has improved from 0.035 to 0.03 while the MSE of the valence model has waned from 0.078 to 0.09. We have achieved an increase in both valence and arousal model accuracy, but there is still a clear distinction in performance between the models. We estimate that while features like loudness can loosely predict arousal, valence has fewer strongly correlated descriptors, which results in a less accurate model.

Metrics	Arousal	Valence
R^2	0.920	.722
MSE	0.03	0.09

Table 3. Performance of Valence and Arousal models.

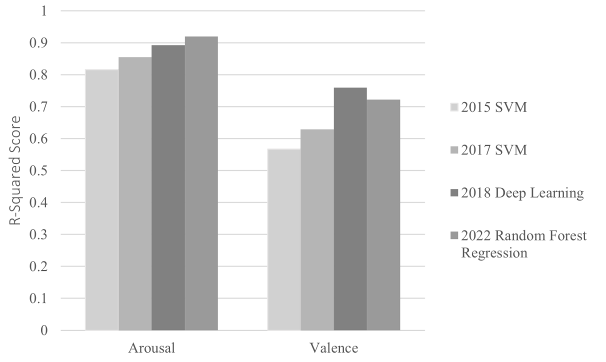

Figure 5. Previous and current predictive performance of valence and arousal models.

9. SEGMENTATION

In this section, we take the results of previous segmentation work [11] and further refine the segmentation process to yield accurately labeled audio files. Thorogood et al. (2016) explore three segmentation algorithms: median filtering, k-depth lookahead, and Maximizing Posterior Probability (MPP). We empirically explore these algorithms in practice, then implement a solution to a prevalent misclassification problem.

9.1 K-Depth Look ahead

The K-depth algorithm traverses the list of segments and searches k-depth ahead of segment A to find the furthest instance B of the same class. If it finds a match, all instances of classes between the initial segment A and matched segment B are reclassified to match segments A and B. The algorithm then continues from the segment after B. If no match is found, we continue from the segment after A.

We evaluate the median filtering, k-depth lookahead, and MPP algorithms by segmenting 15 audio files and comparing the results. The median filtering algorithm smooths nicely but loses valuable foreground segments at all k values. In Figure 7, we show the results of median filtering for k values of 0 to 7. The MPP algorithm performed well overall, but ultimately the k-depth lookahead algorithm yielded the best smoothing while maintaining resolution. Similarly, Thorogood et al. [11] find the k-depth algorithm performs best, though we note that their evaluation window is 0.25 seconds while ours is set to 1.5 seconds. We set the k-value to 2 and give preference to foreground segments.

9.2 Margin Smoothing

In audio engineering, it is common to fade samples in and out. With the abundance of features we extract, the fade must have some non-trivial effect on the classification model. From the previous classifier, we observe an unusually high presence of foreground classification for the foremost and rearmost segments. Figure 7 shows a 44-second long example audio file that contains no foreground segments other than where the fade occurs: the margins. Every fade is likely misclassified as foreground because they emulate the high standard deviation in features commonly present in a foreground sound. We implement a margin smoothing technique for every file that uses a median filter to smooth the first and last segments classified as foreground. The median filter causes a loss of resolution at the margins but justifies itself by reducing overall classification error.

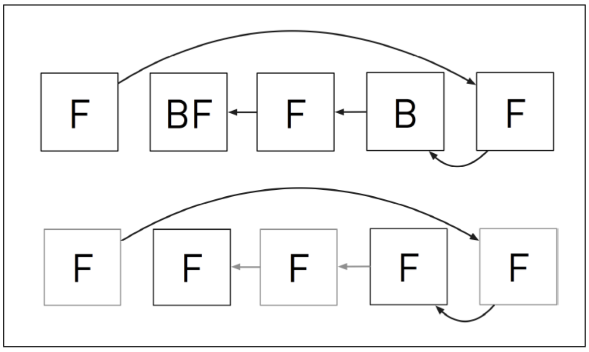

Figure 6. K-depth segmentation system jumping k = 3 then searching k - 1 and backtracking to relabel windows. Figure from Thorogood et al. [11].

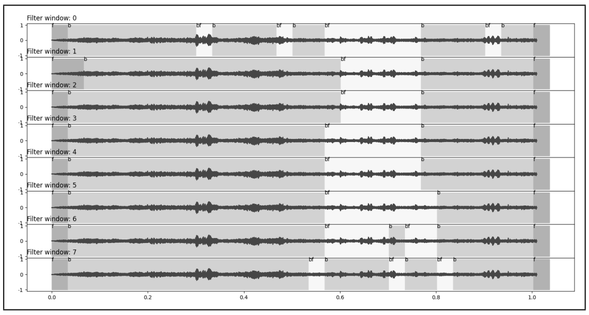

Figure 7. Median Filter Testing

10. RESULTS

To assess the performance of the new implementation of AUME, we compare segmentation between old and new systems. This comparison proves that the pipeline has been successfully reconstructed; we use it to evaluate the characteristics of each system. We extract background-foreground labeled segments from 15 audio files of various soundscape classifications using the results from the new system to compare them to the old system. The first comparison is illustrated in Figure 8, where the new system achieves identical results in the classification of this 44-second long audio clip containing sounds of the seashore. This comparison shows successful replication of the pipeline but fails to show any improvements or shortcomings on the new system. The second comparison is illustrated in Figure 9, where the new model shows slightly smoother results with only 10 segments compared to 12. This is representative of most other comparisons in that there are fewer total segments in the output of the new model. Additionally, Figure 9 demonstrates the effect of margin smoothing. The first segment produced by the old model is incorrectly classified as foreground due to the fading effect mentioned in section 9.2. The new model corrects for this and correctly classifies the segment as background.

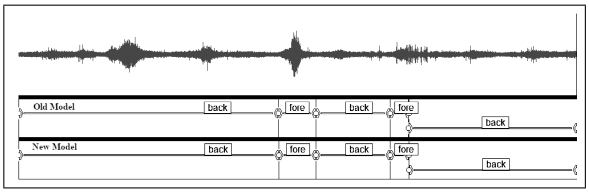

Figure 8. Labelled audio segments generated by previous (middle) and current (bottom) models for performance comparison for an audio file with identical results. The waveform (top) is displayed using the Audacity software.

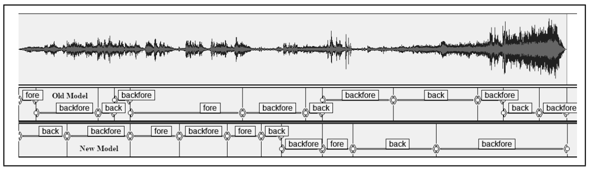

Figure 9. Labelled audio segments generated by previous (middle) and current (bottom) models for performance comparison for an audio file with divergent results. The waveform (top) is displayed using the Audacity software.

11. CONCLUSION

We describe a soundscape composition engine that chooses audio segments using natural language queries, segments and classifies the resulting files, processes them, and combines them into a soundscape composition at interactive speeds. This implementation takes user input to generate search queries and retrieves audio files that are semantically linked to audio files in the database. Sound designers can specify curves for valence and arousal to modulate the perceived emotion of the track over time.

The new AUME takes steps to improve generative soundscape composition by vastly expanding the database, implementing a robust classifier, drastically improving emotion prediction, and improving smoothness using segmentation. We expand the AUME database by 96.8 fold to increase system range and depth. System range is increased by covering a larger number of unique tags, while depth is increased by having more variations of tags already in the system. While the classifier model falls short by 3% when compared to the old classifier, it achieves similar results in practice 8. We improve prediction in arousal by 8.2% and valence by 15.6%. These improvements in emotion prediction offer the sound designer greater control in the depth and movement of emotion.

Audio Metaphor can be used to improve the productivity of sound designers by automating tedious audio database searches. Further, AUME automates the background, foreground segmentation process and composition. This level of automation affords sound designers and soundscape composers the ability to create long, complex soundscapes with a simple text query. When the length of the soundscape proves arduous for composers and designers, AUME offers a practical and efficient creative solution. Example soundscapes generated by AUME can be viewed online [1].

Acknowledgments

We would like to acknowledge the National Science and Engineering Research Council of Canada, and the Social Sciences and Humanities Research Council of Canada for their ongoing financial support. Additionally, we would like to thank Freesound.org for collecting and hosting creative commons licensed sounds.

12. REFERENCES

[1] B. Truax, "Soundscape, acoustic communication and environmental sound composition," *Contemporary Music Review*, vol. 15, no. 1-2, pp. 49–65, 1996.

[2] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[3] M. Thorogood, J. Fan, and P. Pasquier, "A framework for computer-assisted sound design systems supported by modelling affective and perceptual properties of soundscape," *Journal of New Music Research*, vol. 48, no. 3, pp. 264–280, 2019.

[4] D. Birchfield, N. Mattar, and H. Sundaram, "Design of a generative model for soundscape creation," in *in Proceedings of the International Computer Music Conference. International Computer Music Association*. Citeseer, 2005.

[5] A. Eigenfeldt and P. Pasquier, "Negotiated content: Generative soundscape composition by autonomous musical agents in coming together: Freesound." in *ICCC*, 2011, pp. 27–32.

[6] P. Teixeira, G. Bernardes, and M. Davies, "Fostering the database in audio production environments by affect soundscape retrieval."

[7] K. M. Bellisario and B. C. Pijanowski, "Contributions of mir to soundscape ecology. part i: potential methodological synergies," *Ecological Informatics*, vol. 51, pp. 96–102, 2019.

[8] M. Thorogood and P. Pasquier, "Computationally created soundscapes with audio metaphor." in *ICCC*, 2013, pp. 1–7.

[9] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software." in *ISMIR*. Citeseer, 2010, pp. 441–446.

[10] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, J. Salamon, J. R. Zapata González, X. Serra *et al.*, "Essentia: An audio analysis library for music information retrieval," in

[1] Online Sound Examples: https://digitalmedia.ok.ubc.ca/projects/aume-original/mixingExamples/

Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8. International Society for Music Information Retrieval (ISMIR), 2013.

[11] M. Thorogood, J. Fan, and P. Pasquier, “Soundscape audio signal classification and segmentation using listeners perception of background and foreground sound,” *Journal of the Audio Engineering Society*, vol. 64, no. 7/8, pp. 484–492, 2016.

[12] B. Truax, “The world soundscape project,” *WORLD SOUNDSCAPE PROJECT. Accessed May*, vol. 15, 2013.

[13] “The world soundscape project nbsp; to access the official wsp website, including its print publications, click here. to access the full wsp database with complete recordings, interviews, videos, etc. contact barry truax (truax@sfu.ca) for a guest password. the following introductory material is included here for ease of access.” [Online]. Available: https://www.sfu.ca/~truax/wsp.html

[14] R. M. Schafer, *The soundscape: Our sonic environment and the tuning of the world.* Simon and Schuster, 1993.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] J. Fan, M. Thorogood, and P. Pasquier, “Emo-soundscapes: A dataset for soundscape emotion recognition,” in *2017 Seventh international conference on affective computing and intelligent interaction (ACII)*. IEEE, 2017, pp. 196–201.

[17] J. Fan, F. Tung, W. Li, and P. Pasquier, “Soundscape emotion recognition via deep learning,” *Proceedings of the Sound and Music Computing*, 2018.

[18] J. Fan, M. Thorogood, B. E. Riecke, and P. Pasquier, “Automatic recognition of eventfulness and pleasantness of soundscape,” in *Proceedings of the Audio Mostly 2015 on Interaction With Sound*, 2015, pp. 1–6.