Systematic Database Creation for Expressive Singing Voice Synthesis Control

Martí Umbert, Jordi Bonada, Merlijn Blaauw

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

marti.umbert@upf.edu, jordi.bonada@upf.edu, merlijn.blaauw@upf.edu

Abstract

In the context of singing voice synthesis, the generation of the synthesizer controls is a key aspect to obtain expressive performances. In our case, we use a system that selects, transforms and concatenates units of short melodic contours from a recorded database. This paper proposes a systematic procedure for the creation of such database. The aim is to cover relevant style-dependent combinations of features such as note duration, pitch interval and note strength. The higher the percentage of covered combinations is, the less transformed the units will be in order to match a target score. At the same time, it is also important that units are musically meaningful according to the target style. In order to create a style-dependent database, the melodic combinations of features to cover are identified, statistically modeled and grouped by similarity. Then, short melodic exercises of four measures are created following a dynamic programming algorithm. The Viterbi cost functions deal with the statistically observed context transitions, harmony, position within the measure and readability. The final systematic score database is formed by the sequence of the obtained melodic exercises.

Index Terms: expressive singing voice synthesis, unit selection, database creation

1. Introduction

Expressive performances have attracted the interest of researchers for the last years. Providing expression and emotions has been a goal for many types of synthesizers, from instruments to speech and singing voice synthesis. An important issue is to provide control data to the synthesizer which represent a given emotion, expression or style. Several strategies have been proposed to cover the target expressive space.

In the case of the reconstructive phrase modeling system [1], the Synful orchestra achieves musical expressivity by deriving parameters from database of real songs according to a target song. In their work, working with a systematic database is discarded because it would reduce the expressive power of the system since the amount of expressive articulations is too large. In the current work we propose a possible solution to overcome this problem with a double criteria: to cover the frequent articulations with a musical criteria at the same time.

Other approaches have designed scores by taking musical phrases from real repertoires, as in [2] for a violin synthesizer. A first manual step is done to select a repertoire that covers the target features and that is musically relevant. Then, an algorithm is used to select and transform trajectories to cover different note transition features (e.g. articulations, note intervals) and intro note features (accents, duration and dynamics).

In concatenative speech synthesis, articulations are captured from real recordings of sounds, which span from single or groups of phonemes (diphones, triphones), to words or sentences. In [3], unit selection of the recorded sounds has been studied. However, when preparing scripts to achieve phoneme coverage, other aspects can be taken into account, such as how difficult words are to read or grammatical correctness of a formed sentence, which has been addressed in [4] and also in [5].

In emotional speech synthesis, statistical modeling techniques have been proposed to model speaking styles as in [6]. In this case, similar to recording real musical scores, no specific constraints are given to the recording scripts. Emotion related scripts are recorded and then modeled with HMMs.

Statistical modeling of singing style has also been used in [7] with focus to relative pitch, vibrato and dynamics using context dependent HMMs. In this case, real recordings are used and therefore no previous study is performed with respect to which scripts given the target style.

This paper studies the generation of a set of exercises that represent to some extent a given style properties and melodies. These exercises will then be recorded by one singer in one style. Pitch, dynamics and note durations will be extracted and used within a unit selection-based approach to generate expressive controls of a singing voice synthesis system.

In our case we did not choose the option of generating melodic exercises directly from real repertoires. These typically have the disadvantage of being redundant, so only a portion of an entire song introduces new note sequences. Also, in order to select which part of a song to include as an exercise, it should be carefully studied. Therefore, we obtain melodic exercises by concatenating short melodic units generated in a systematic way, also including musical knowledge as explained in section 2. First, a set of scores are statistically analyzed in order to know which feature values (note strengths and figures and pitch intervals in semitones) should be covered, their relevance and how these are connected. Then, dynamic programming is applied in order to generate melodic exercises as sequences of concatenated units. Finally, in section 3 conclusions and future work are presented.

2. Database creation

2.1. Units versus contexts

The basic elements of our systematic process of melodic exercises creation are units made up as sequences from one to three notes surrounded by a previous and following note or silence. An example is shown in Figure 1. In this paper a note is defined mainly by the following properties: note strength, note duration (seconds), and the figure and pitch interval with the next one. Note strength (NS) is a measure for the accentuation of a note beat within a bar. Figure interval (FI) refers to the relationship between two consecutive note durations and the same applies to pitch interval (PI) with respect to the note frequencies. This data is shown in Figure 2.

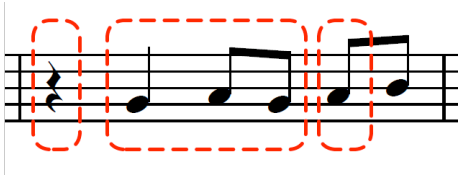

Figure 1: *Unit of three notes with preceding silence and following note.*

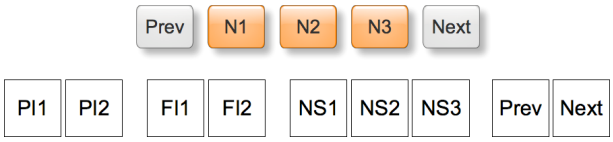

Figure 2: *Unit and context features.*

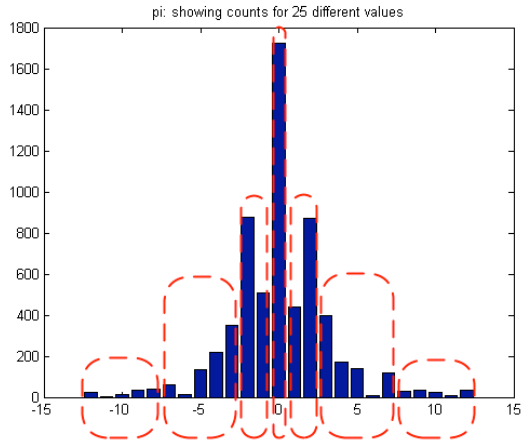

Figure 3: *Pitch interval distribution (in semitones) and clusters.*

For each note property there are many possible combinations, which imply a great amount of units, specially in the case of sequences of three notes. This relates to the goal of the systematic database, which is to cover a high amount of relevant note combinations. Therefore, the coverage criteria is not defined with respect to the units but related to a higher abstract unit or context. Each context comprises several possible units.

Thus, the relationship between units and contexts has to be defined by grouping the set of values of each note property into clusters. Once the clusters are set, it is possible to statistically analyze the transition probabilities between contexts. Both steps are explained in the following section.

2.2. Statistical analysis and clustering

In order to study the set of note properties that need to be covered, a set of songs belonging to the same style have been processed using Music21 [8], a Python toolkit to process music in symbolic form.

Since most of the processed units are three notes long, and each note is defined in terms of its strength, duration, and figure and pitch intervals, the possible number of units is enormous. As explained in the previous section, to reduce the amount of units to cover, these are clustered into similar contexts.

In general, clusters have been organized so that close values are represented by the same cluster. In the case of pitch interval clusters, it has been taken into account that within the same cluster all pitch intervals correspond to only ascending or descending intervals, since we do not want to transform an ascending pitch contour to synthesize a descending one (and vice versa). Therefore, an interval of zero semitones (same consecutive notes) is grouped in a separate cluster. In the case of the figure interval, clusters do not need to follow the same constraint concerning the direction of the interval (ascending or descending). Note strength clusters have been grouped according to the note accentuation within a measure.

In Figure 3 and Table 1, the values distribution for the figure interval and their clustering is shown. The same information is presented with respect to note strength in Figure 4 and Table 2, and concerning pitch interval in Figure 5 and Table 3.

Using this cluster representation, the context frequencies have been counted and the 90% most common ones have been selected to be covered, generating a list of 993 contexts of three notes. Also, the amount of connections between these selected contexts (by overlapping two or one notes or just concatenating them) has been computed to measure the transition probabilities among contexts. These contexts are a higher level representation of 1480 units.

Table 1: *Pitch interval cluster values.*

Cluster	Range of values
1	[-12, -8]
2	[-7, -3]
3	[-2, -1]
4	[0]
5	[1, 2]
6	[3, 7]
7	[8, 12]

2.3. Melodic exercises generation

The Viterbi algorithm has been used in order to generate the sequence of melodic exercises of the systematic database. The temporal resolution, or tick, of each melodic exercise is defined by the minimum note length. In our case we have used a tick of an eighth note. The sequence of ticks defines a note strength grid which is used in order to know which units fit at each position in time. The note strength grid generation is explained in section 2.3.1.

At each (forward) step of the Viterbi algorithm, the accumulated cost of inserting a given database unit at a certain tick is computed using a set of cost functions explained in section 2.3.2. These cost functions handle the transitions between units according to the statistical information at context level computed as explained in section 2.2. The cost functions also measure whether an instance fits in the grid and reusing a context is penalized. Harmony is managed by the preset accompaniment chords (which convey the target style) of the melodic exercises and how these and the unit notes match. Inserting silences in the middle of the exercise is also favored considering readability, in order to help the singer to breath in the middle of the performance. Also, the generated note pitches are constrained to the singers tessitura in order to facilitate singing the exercises.

The following subsections explain the process followed to generate the melodic exercises as sequence of three note long units. In a similar way exercises of two and one notes were

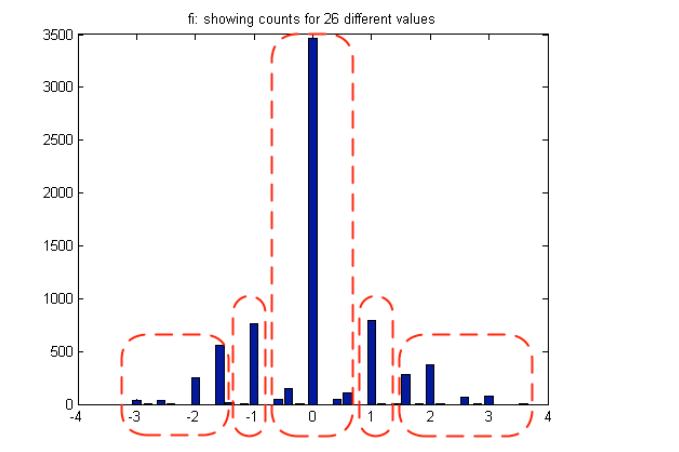

Figure 4: *Figure interval distribution (in octaves) and clusters.*

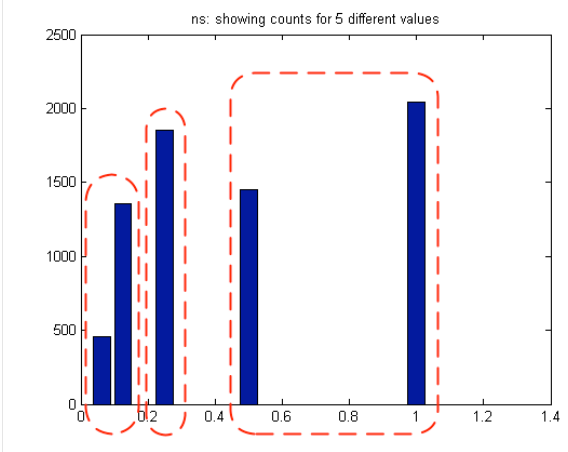

Figure 5: *Note strength distribution and clusters.*

generated. In these cases, the previous and following notes are considered to be silences, so the Viterbi algorithm was not necessary since unit overlapping does not apply. These exercises were generated in a more straightforward manner by taking one value per cluster to generate the contexts to cover.

Table 2: *Figure interval cluster values.*

Cluster	Range of values
1	[-3, -1.585]
2	[-1.41, -1]
3	[-0.585, 0.585]
4	[1, 1.415]
5	[1.585, 3.585]

2.3.1. *Note strength grid*

Given the minimum note length that will be used in the systematic score, a grid can be generated which sets where notes can be placed and which their note strengths are at those positions. The length of this grid is related to the amount of measures per exercise.

For a minimum note length of an eighth note, the note strength grid for a single measure (4 beats, 8 ticks) is musically defined as shown in the following vector:

$$[1, 0.125, 0.25, 0.125, 0.5, 0.125, 0.25, 0.125] \quad (1)$$

2.3.2. *Cost measures*

The accumulated cost for an evaluated node of the Viterbi matrix is evaluated by several cost measures.

The first computed cost checks whether the note strengths features of the unit match the note strengths related to the tick position where is is intended to be inserted. If the unit does not fit, then it is not necessary to check all the other costs, and the total cost is set to infinity. For units that do fit, the cost is set to zero.

The second computed cost relates to the transition between units. The result of the statistical analysis provides this cost for an overlapping of two, one or zero notes (concatenation). These transition is computed for the current selected unit with respect to all possible previous units.

Table 3: *Note strength cluster values.*

Cluster	Range of values
1	[0.5, 1]
2	[0.25]
3	[0.125, 0.625]

Since the aim is to have the highest coverage possible with the minimum amount of melodic exercises, context repetition is taken account for penalization. Therefore, a history of all previously selected contexts is kept, so that if in the currently evaluated node path there is a context repetition, a cost proportional to the amount of repetitions is added. Although some context repetitions may appear in the final score, this cost favors the selection different contexts.

The harmony cost takes into account the chords for the melodic exercises. The same sequence of chords has been predefined for all exercises in order to make it easy for the singer: C7 (1st bar), Am7 (2nd bar), Dm (3rd bar 1st half), G7 (3rd bar 2nd half), C7 (4th bar). Those notes with cost zero are the ones belonging to the chord. Otherwise, it is more costly to add notes which do not match with the chord note information. In Table 4 the harmony costs are shown relating which notes are favored (zero cost) per chord and which ones are more penalized (non-zero cost).

Finally, since melodic exercises are four measures long (plus one as a break between exercises), and in order to make them more easy to sing, a silence has been included in the middle, at the end of the second measure and at the beginning of the third one. Several tick candidates for inserting the pause are considered in the Viterbi paths and the least costly one is chosen.

Table 4: *Harmony costs.*

Bar	Chord	C	D	E	F	G	A	B
1	C7	0	1	0	2	0	1	0
2	Am7	0	1	0	2	0	0	1
3	Dm	1	0	1	0	2	0	2
3	G7	2	0	2	0	0	1	0
4	C7	0	1	0	2	0	1	0.5

2.3.3. Stop criteria

The algorithm stops generating melodic exercises depending on two conditions. The first one is related to the coverage. If all 993 contexts have been selected (one unit per context is enough) after the generation of a melodic exercise, the generation of exercises is stopped. This is controlled by the history of selected contexts as explained in the previous section.

The second stop criteria is related to the available recording session duration and the tempo of the generated score. If the accumulated duration of all exercises reaches the recording time, given the amount of measures per exercise and the bpm, then no more melodic exercises are generated.

2.3.4. Results

The systematic script has been generated by taking 57 jazz standard songs, setting the tessitura to one octave, a tempo of 71 bpm and a limit for the recording time of one hour. These constraints generate a recording script of 236 exercises and a coverage of 82% of contexts.

The generated melodic exercises as concatenation of three note long units can be downloaded in pdf and audio files are online at: `http://www.dtic.upf.edu/~mumbert/ssw8/`.

3. Conclusions

A system for the systematic generation of melodic exercises has been presented. The aim of such melodic exercises is to cover the statistically and musically more relevant note combinations in terms of note strength and figure and pitch intervals. The concepts of units up to three notes and their feature clustering in order to group them into high level contexts has been presented in order to define the coverage criteria.

We plan to perform an evaluation in order to prove that the generated systematic score is representative of the songs statistically analyzed. This can be proven by taking a set of target songs different from the analyzed set but belonging to the same style. Our unit selection-based approach can be used in order to retrieve units from the systematic database and to measure the degree of transformation (note duration, pitch shifting) that these require to match the target. Also, the difference of note strengths between the selected units and the target units can be computed. Finally, phrasing aspects are important. For example, the length of retrieved sequences of consecutive units from the database is a measure of representativeness. The longer these sequences are, the longer the recorded contours used by our framework will be.

These measures should differ from generating a systematic score from another style database and computing the levels of unit transformations and consecutive unit sequences lengths.

We also plan to improve the grouping process of note properties into clusters. This could also be done following a K-means algorithm. Also, the central values within a cluster should be more represented in the final score than extreme values.

Also, once the systematic score is recorded, and the expressive contours extracted, the complete framework with both symbolic and expressive trajectories will be tested to generate the expressive contours.

4. References

[1] E. Lindemann, "Music synthesis with reconstructive phrase modeling," *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 80–91, 2007.

[2] A. Pérez, "Enhancing spectral synthesis techniques with performance gestures using the violin as a case study," Ph.D. dissertation, Universitat Pompeu Fabra, 2009.

[3] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 373–376.

[4] M. Dong, L. Cen, P. Chan, and H. Li, "Readability consideration in speech synthesis recording script selection."

[5] S. Fitt, "Using real words for recording diphones," 2001.

[6] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi, "Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing," *IEICE - Trans. Inf. Syst.*, vol. E88-D, no. 11, pp. 2484–2491, Nov. 2005.

[7] K. Saino, M. Tachibana, and H. Kenmochi, "A singing style modeling system for singing voice synthesizers." in *INTERSPEECH*. ISCA, 2010, pp. 2894–2897.

[8] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *Proceedings of the International Symposium on Music Information Retrieval*, vol. 11, 2010, pp. 637–42.