Application for the Automatic Pitch Detection and Correction of Detuned Singing

Małgorzata Michalska
École Polytechnique Fédérale de Lausanne, Switzerland

Abstract: This paper describes an application for automatic detection and correction of detuning in singing. It presents the observations that became the core of the work, application principles, limitations, perspectives and used algorithms. It explains in detail the experiments performed and the results obtained. Finally, it discusses some opportunities that have been revealed during the research and points to improvements and extensions possible in the future work.

Keywords: pitch detection, pitch correction, pitch shifting, detuning, melody alignment, singing evaluation

1. Introduction

It is not very surprising that technology influences more and more aspects of human lives and social activities. It impacts the marketing and improves the quality of any product to be sold. Pitch correction is not an exception at all. With current techniques is it possible to find and remove even slight detuning in singing and apply various effects to improve the quality of the recording [1, 6]. The author's application described in this paper makes also an effort to automatically detect and correct detuning in singing, however its concept and destination are different comparing to the software currently available for various platforms. It is in fact a personal singing trainer, where the emphasis is put on educational and entertainment aspect.

The problem that has to be solved in both cases is not trivial at all. First, in order to narrow the framework of exploration and maximize the usability of the application, it is necessary to make some assumptions. With that being done, appropriate signal processing algorithms must be chosen and implemented. Their obvious limitation is the complexity that most likely disables real-time execution. Finally, all previous steps, even the strictly technical ones have to be performed with regards to musical aspects. Good knowledge and analysis of basic properties of singing, such as common mistakes (tempo changes, fluctuation, skipping the tone, immediate changes of key), features of human vocal tract or some elements of music theory (key and its transformation into frequencies) are necessary to handle the stated problem [14, 16, 22]. These aspects all together form an approach that has been rarely explored and documented so far.

The paper is structured in the following way: first, in Section 2. the challenges related to pitch detection as well as currently existing tools for pitch detection and correction are shortly described. They are compared with the assumptions applied to the application. Section 3. handles the application principles in a more detailed way and describes the algorithms that have been used at different stages of analysis. Next, the experiments, tests, results and author's observations are described in Section 4. In Section 5. the observations are presented from another perspective, namely they focus on problems that might have arisen from necessary simplifications and the opportunities of improvement. Finally, Section 6. summarizes the experiments performed and the attention is driven to more interdisciplinary research in the future.

Autor korespondujący:
Małgorzata Michalska, malgorzata.michalska@epfl.ch

Artykuł recenzowany
nadesłany 6.01.2016 r., przyjęty do druku 1.02.2016 r.

2. Related Work

2.1. Pitch Detection Challenges

The analysis of audio content as such can cover multiple aspects [11]. One of the problems broadly investigated in literature is an (automatic) singing quality evaluation with the most common context related to Query-by-Humming systems [20] or vocal training [17, 25]. Systems of that kind must rely on the pitch detection which is a well-established problem in the audio-processing field with several effective methods proposed in literature, operating in both: time- and frequency domain [7, 8]. An important challenge for such applications is to efficiently model the common errors made by people at singing [13] and also make a clear distinction, what is an error and what acts as a grace, especially for the purpose of automatic evaluation. There are several operations that can be troublesome to handle by automatic systems, for example glissando (a continuous slide between two sounds), vibrato effect (fluctuation of a sound frequency) or octave errors by frequency detection. These issues contribute to the complexity of note segmentation problem which is an essential element of any pitch detection system [3].

2.2. Currently Available Software

2.2.1. Auto-Tune

According to the producer, Auto-Tune is used daily by thousands of audio professionals around the world. It is often referred as a holy grail of recording and has been adopted

worldwide as the largest-selling audio plug-in of all time. Its features include correction of intonation and timing in vocals or solo instruments, preservation of the natural properties of the input, advanced features like formant modelling and many others. The program can work in automatic and graphical modes [1, 2].

2.2.2. Celemony Melodyne

In 2008 the German magazine Der Spiegel described Celemony Melodyne as a *photoshop for sound*. The innovative approach to visualise music in Celemony involves representing single sounds as graphical objects. Parameters of an object (length, width, position) reflect some properties of sound, such as duration, volume and pitch. Unlike Auto-Tune, Celemony is able to process polyphonic sound [4, 5].

2.2.3. Serato Pitch'N'Time Pro

Serato Pitch'N'Time Pro is a pitch-shifting and time-stretching plug-in for the Pro Tools platform intended to be used by professionals. Pitch'N'Time provides a good quality of pitch shifting and tempo correction which do not affect the timbre of the recording. It was achieved by using some properties of human perception in the sound processing instead of mathematical operations only. The program is currently widely used in the UK to correct pitch in the films before they are to be transferred into a video version [10, 19].

2.3. Application Proposed

Since the concept behind the approach described in this paper is not market-oriented, the purpose of the proposed application it to make the user aware of errors and not to mask them. Moreover it targets at possibly simple algorithm implementations to provide efficiency. However, instead of making an attempt of real-time processing, which is a highly complicated task, it provides visualization in real-time to achieve the desired educational effect [21]. For training purposes, the level of detection has been set to the so-called "merciless". Correction is used here as another means of learning. Due to possible changes of key (or tuning other than the MIDI standard of 440 Hz), the analysis of singing within the application is being performed by intervals (frequency ratio of two consecutive notes) instead of absolute values. Application has been designed as a desktop one, written in C++ using Qt 4.8.4 platform and Windows 7 as an operating system. However, thanks to the environment it could be easily transferred into other systems.

3. Used Techniques

3.1. Detailed Principles

The analysis of singing is performed according to the pattern provided in the MIDI format. An appropriate database of sample melodies to be tested has been prepared. Without the pattern file, the sung sequence can be evaluated only in terms of keeping up with the tones of a scale. The application enables uploading a WAVE PCM file containing a singing or recording it directly using a microphone. Detuning of each note is calculated in cents. Cent is a conventional unit expressing the distance between two sounds and the size of a relevant interval. It corresponds to 1/100 of a semitone and 1/1200 of an octave. The frequency ratio between the sounds 1 cent away is equal to $\sqrt[1200]{2} \approx 1.00058$ [15]. Such a measure provides extremely precise evaluation and puts an emphasis on tones sung *in between* compared to the scale what very often results from the incorrect intonation of a sound.

The main priority is given to the visual aspect of the application. Instead of the standard musical representation which is not readable for the majority of society, the notes are represented as blocks whose vertical alignment denotes the frequency. The notion of time is represented by the horizontal alignment and for the case of playing back the pattern melody is also represented with colours. It is assumed that the melody is sung using syllable *na* instead of actual lyrics or any other vocalization. It is mainly due to the fact that some sounds do not carry any frequency and will be simply classified as noise. Such sounds might be also very difficult to transpose without a significant distortion. Finally, the *n* at the beginning of each note helps the user split the consecutive notes correctly, what is of highest importance for transforming the sequence of sung frames into a sequence of notes.

3.2. Processing Algorithms

The very first operation that has to be performed on each frame of signal is to determine whether it contains a sound or silence. It is achieved by calculating the variance of the signal inside each frame. The frames are classified based on this value: if variance is too small, the frame is detected as silence and omitted in the analysis. The discrimination level has been tuned during the experiments with users and took into consideration the sources of noise, such as: microphone, environment or improper voice emission. It has been observed that the tuning of variance level is critical for the purpose of further processing.

Pitch detection is performed using the autocorrelation method which can be explained in the simplest way as a cross-correlation of a signal with itself [23]. The first maximum for a non-zero argument is the length of the period given in samples which can be easily transformed into frequency. The operation is performed on frames consisting of 2048 samples (rectangular window) that overlap in 50%. This length has been established experimentally in order to conform with the frequency range it has to deal with and with the sampling rate equal to 44.1 kHz used for all test recordings. Processed frames are then grouped into notes according to the averaged values and possible shifts of frequency in consecutive frames.

The alignment of a sung sequence and a MIDI pattern is performed using Levenshtein distance, also referred as L-distance, edit-distance or minimum edit-distance, which is widely used in bioinformatics to match two sequences of DNA locally or globally. The algorithm aims at estimating the minimal cost that is necessary to transform one sequence into another one, where the transformation can be performed using three different operations: changing the symbol for another one, skipping or adding the symbol. This method completely disregards the time dependencies in the sequence. Since the time dependencies (i.e., rhythm) are likely to be incorrect or distorted in a singing of an average person, it can be considered as a huge advantage.

Pitch correction is performed using the PSOLA algorithm (Pitch Synchronous Overlap And Add). This algorithm consists of particular phases, which are: splitting the signal into segments, pitch marks determination, centring the segments at the pitch marks, stretching or squeezing (according to the transposition up or down) and finally overlapping and adding [14]. The factor of operation depends on the detected detuning expressed in cents, as defined previously.

4. Experiments

Tests of the application with users of different levels of singing or musical proficiency were the essential part of the research. Performed regularly during the development process, they enabled a design feedback, where constant modifications

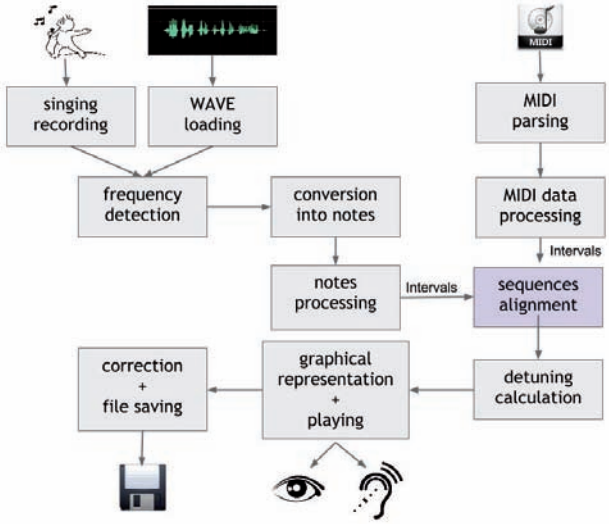

Fig. 1. Use case and stages of analysis
Rys. 1. Przykład użycia i etapy analizy

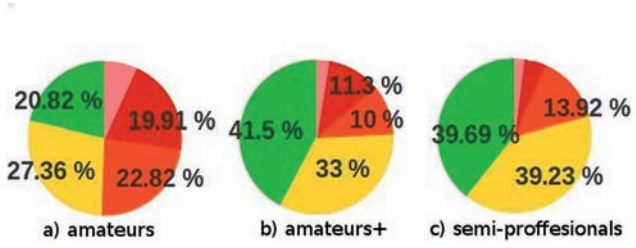

Fig. 2. Percentages of sung notes for different groups of testers
Rys. 2. Procentowy udział nut dla różnych grup użytkowników

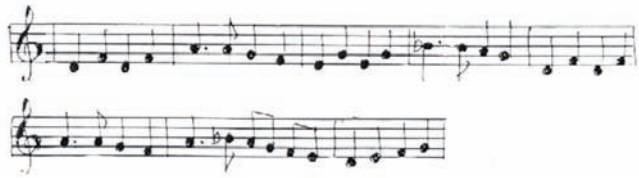

Fig. 3. Musical representation of a popular folk melody "Hej sokoły"
Rys. 3. Muzyczna reprezentacja popularnej melodii ludowej "Hej sokoły"

and tuning of some application parameters were applied. The whole group of testers consisted of 34 persons, males and females. They could be split into 3 groups: amateurs, amateurs that enjoy singing and take part in some musical activities and the so-called semi-professionals, members of a university choir. Such a spectrum of users enabled validation, evaluation and brought many thoughts and observations as well as suggestions coming directly from the testers.

Each test was performed according to the same pattern. A detailed use case and stages of analysis are summarized in Figure 1. Each user chose 2 songs from the provided database, however, one song (considered as an *easy* one) was sung by all persons, while the second one was optional and could be chosen freely. Each song was played back along with the visualization and the user could train as long as one wanted before the actual attempt of recording a singing.

Once the recording was made, it was played back and its graphical representation was displayed on the screen. Detuning was indicated with colours, starting from green (correct note) ending at red (detuning bigger than one semitone). Such a representation pointed out extremely precisely a lot of mistakes that were made very often by users at different levels of proficiency. Surprisingly, even in the most proficient group only around 40% of notes were *green* (correct). Figure 2 shows the percentages of each note colour within 3 groups. The ranges of particular colours are the following: green – detuning less than 20 cents, yellow – between 20 and 50 cents, orange – between 50 and 100 cents, red – more than 100 cents. Pink notes are those which could not be aligned by the algorithm.

At this point, the application was also analysed for the occurrence of octave errors during the pitch detection stage. This kind of errors are a very common side effect of simple pitch detection algorithms, such as, used here, autocorrelation. Contrary to the literature [7], not a single octave error has been observed in the singing of the tested group of users. It was not the case, when the application was tested, for example, using the melody played by the violin.

Although it was possible to sing with a metronome, very often the tempo, as well as rhythmic dependencies were affected. Thus, it confirmed the legitimacy of disregarding the time dependencies and focusing on the alignment of relative intervals. When the subjects were asked to point the mistakes in singing that the application showed and they agreed with, most common ones were singing slightly below/above the desired pitch, floating voice that changes the pitch when it is supposed to sing one note only or changing the key in the middle. The author's observation is that sometimes these mistakes resulted from stress caused by producing a recording. Nevertheless, over 95% of tested subjects confirmed, that they agree with the mistakes found by the program. Some of them emphasized that they had been aware of these imperfections even before and the program simply confirmed their vocal problems. On the other hand, some commented that the evaluation by a program is *merciless* and that the accuracy of detuning detection is much higher than the sensitivity of a human ear. Around 5% did not agree with the errors marked

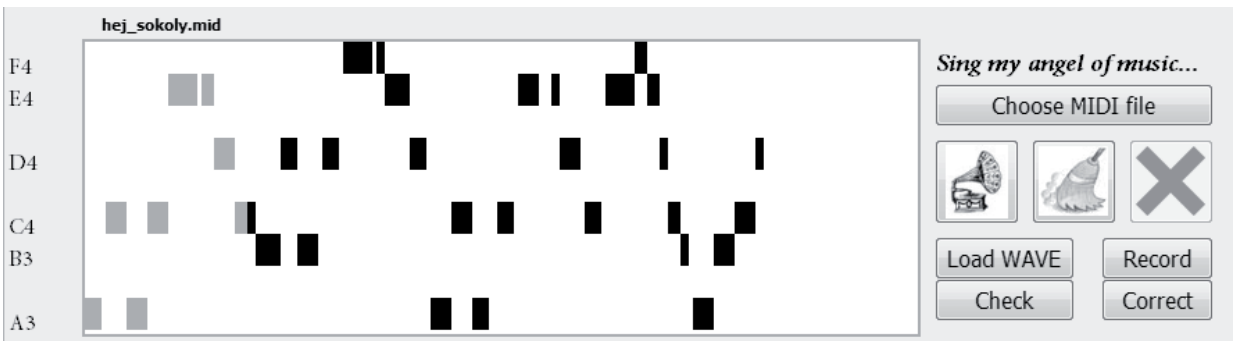

Fig. 4. Representation of the same melody stored in a MIDI file (current position of the playing back denoted with a colour)
Rys. 4. Reprezentacja tej samej melodii zapisanej w pliku MIDI (obecna pozycja odtwarzania oznaczona kolorem)

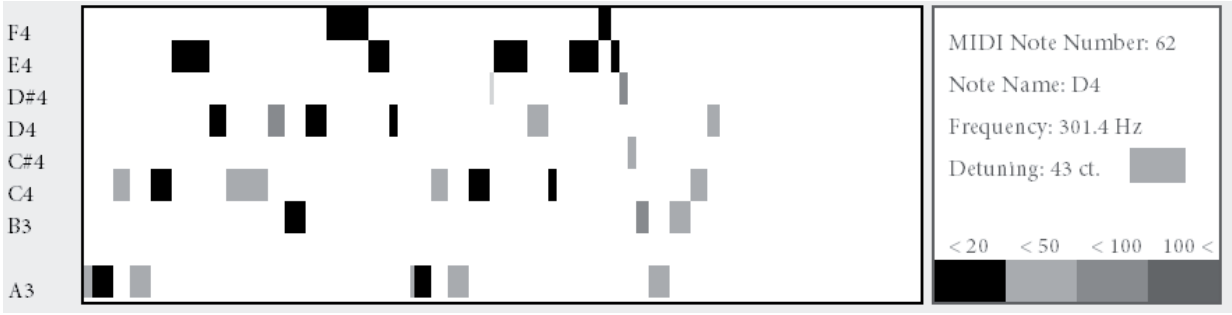

Fig. 5. Visualisation of the melody sung by a user
Rys. 5. Wizualizacja melodii zaśpiewanej przez użytkownika

Fig. 6. Example of a sung sequence before correction
Rys. 6. Przykład zaśpiewanej sekwencji przed korekcją

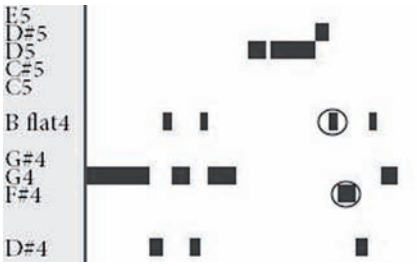

Fig. 7. Example of a sung sequence after correction. Remarkable shifts applied to: B4 (transformed to Bflat 4) and G4 (transformed to Fsharp 4)
Rys. 7. Przykład zaśpiewanej sekwencji po korekcji. Wyraźna korekcja dla dźwięków B4/H4 (obniżony do Bflat4/B4) oraz G4 (obniżony do Fsharp4/Fis4)

by the application, but an interesting point is, that all of them were amateurs and had problems to define what *off tune* actually means. Probably an appropriate conclusion to this would be to define a strong correlation between the vocal experience and awareness of ones detuning [11]. In terms of usability, over 90% of users claimed that the application is fully usable and intuitive. Among the other 10% there were some suggestions for improvement like integrating the block representation with musical notes or preparing the mobile version of the application to be used portably during, for example, a choir rehearsal.

Figures 3–5 show different representations of the same melody. Figure 3 presents the scores that were not part of an interface, however, they let compare the musical and non-musical way of drawing a melody. Figure 4 depicts the block representation of the MIDI pattern in the middle of playing back along with the elements of the GUI. Finally, Figure 5 is an example of the analysis performed. In the window on the right, a detailed information about the note the mouse is currently over is shown.

Pitch correction was performed according to the detuning established in the analysis phase and marked with an appropriate colour. Figure 6 shows a sample input melody processed by the detuning detector and for a comparison Figure 7 presents the same sequence after correction. The melody used here is a fragment of *Imperial March* from *Star Wars* soundtrack. It must be emphasised that the correctness of each note represented with its colour is evaluated first of all with regards to the interval it produces with the preceding note and not to its expected absolute value. Therefore the correction may apply two types of modification. The first type is shifting the note by one or more semitone to obtain the correct interval, what happened for sounds B4 and G4 (although G4 was originally marked as *correct* it must have been shifted due to the change of its predecessor). The second type is smoothing the less significantly detuned notes (most likely resulting from incorrect intonation), as applied to those with detuning above 20 cents.

5. Discussion

Working with people at different levels of musical proficiency leads to some interesting thoughts on the target user group of this application and its usability. For sure, there must be a limit of detuning that can be automatically analysed and corrected. When the number of completely wrong sung notes exceeds the number of more or less correct ones, there is no chance that all modules (detection, alignment, correction) would go smoothly and produce a reasonable output. To use a personal singing trainer the user should have at least *basic ear* for music. Without feeling at least a little bit of control over one's voice, no educational effect in this field can be achieved. Comparing the results from different subjects, it can be concluded that persons with at least some musical awareness produced the best material for analysis.

Another problem related to musical proficiency of the users is the level of voice in the input signal. According to Love [12] the power of human vocal tract has been barely explored by people. Remarkably many of us do not use it correctly and at its full power. In effect the emission of the voice is affected and distorted what introduces much more noise in the produced sound. In this application, such noise very often eliminated a frame of sound from further processing during the very first stage of analysis. To get rid of this limitation a more advanced method for voice activity detection should be applied [18].

It can be also debated if the alignment method that relies on the sequence of intervals and disregards the time dependencies is the optimal solution to the problem. On contrary to its advantage which is the immunity to unintentional changes of key by the user, it can be questioned whether the equal-tempered scale is an appropriate choice for the analysis of relative intervals. An alternative approach could be to use Dynamic Time Warping [20], so that the rhythm and timing errors are also covered.

The biggest disadvantage of the program is the quality of sound that gets worse after the transposition. Considering the fact that most home-made recordings are being performed with a simple equipment, an effort has been made to filter the files using one of some popular audio processing software tools. However, it did not contribute a lot. The only difference was in the slight noise reduction, but some clicks or other undefined sounds could be heard even though. It is possible that they are artefacts produced by the transposition itself and some other more advanced signal processing algorithms are necessary for their removal. On the other hand, some other pitch shifting algorithms, such as phase vocoder might be worth considering [9].

Applying pitch correction raises also the issue of *naturalness*. Even a non-musical ear would probably experience that if multiple sounds in a row are transposed exactly according to the given pitch, the timbre becomes a little bit *robot-like*. For this reason the quality of transposition cannot be measured only in terms of precise pitch shifts and exact frequency alignment. Keeping and preserving the natural features of the human voice is another critical aspect. However, this property is extremely hard to measure based on the technical approach only. Exploring which sounds human ear perceives as natural and which not belongs, however, rather to the psychoacoustics area.

Further improvement that apart from the technical approach would require knowledge in other fields is related to the non-pattern correction case which in this application was handled very simply. Tuning sounds according to the scale is an extremely simplified approach, especially for the sounds sung right in between. Determining the key according to which the sequence should be corrected would be, however, a technical and musicological challenge at the same time.

6. Conclusion

The application for automatic pitch detection and correction presented in this paper differs from the currently available software. Within the application framework a combination of some simple processing algorithms has been delivered in order to provide a more complex solution. An attempt has been made also to extend the technical approach to voice processing by non-technical aspects, such as properties of human voice and singing as a whole. The initial expectations and assumptions have been verified and well tested with people at different levels of singing proficiency. Although the application has its imperfections, it constitutes a very good starting point for further exploration at the borders of Music Informatics.

Acknowledgment

The author would like to thank Prof. Bożena Kostek for her supervision, help and precious feedback.

References

1. *Antares Audio Technologies Auto-Tune* – official website, http://www.antarestech.com/ (accessed 20 Oct. 12).
2. Bellis M., *Who Invented AutoTune*?, http://inventors.about.com/od/astartinventions/a/Who-Invented-Auto-Tune.htm (accessed 18 Feb. 13).
3. Boersma P., *Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound*, Institute of Phonetic Sciences, University of Amsterdam, Proceedings 17, 1993, 97–110.
4. *Celemony Melodyne* – official website, http://www.celemony.com (accessed 20 Oct. 12).
5. *Celemony Melodyne promoting videos*, http://www.celemony.com/cms/index.php?id=videos (accessed 20 Feb. 13).
6. Daley D., *Vocal Fixes: Modern Vocal Processing in Practise,* "Sound on Sound", 2003–2010.
7. Dziubiński M., Kostek B., *Octave error immune an instantaneous pitch detection algorithm*, "Journal of New Music Research", Vol. 34, No. 3, 2005, 273-292, DOI: 10.1080/09298210500235301.
8. Gerhard D., *Pitch extraction and fundamental frequency: History and current techniques*, Tech. Report, Dept. of Computer Science, Univ. of Regina, 2003.
9. Laroche J., Dolson M., New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects, IEEE Conf. on Applications of Sign. Proc. for Audio and Acoustics, New York, 1999, DOI: 10.1109/ASPAA.1999.810857.
10. Lech M., *Application for Automatic Detection and Correction of Detuned Singing*, Master Thesis at Multimedia Department, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 2007 (in Polish).
11. Lerch A., *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, Wiley-IEEE Press, 2012.
12. Love R., *Set Your Voice Free: How To Get The Singing Or Speaking Voice You Want*, Little, Brown and Company/ Hachette Book Group, New York, 1999.
13. Meek C., Birmingham W., *Johnny can't sing: a comprehensive error model for sung music queries*, Proc. of International Symposium on Music Information Retrieval, 2002, 124–132.
14. Pardo B., *Finding Structure in Audio for Music Information Retrieval*, "IEEE Signal Processing Magazine", Vol. 23, No. 3, 2006, 126–132, DOI: 10.1109/MSP.2006.1628889.
15. Pilch M., Toporowski M., *Dawne temperacje. Podstawy akustyczne i praktyczne wykorzystanie.* Akademia Muzyczna im. Karola Szymanowskiego w Katowicach, Katowice, 2014 (in Polish).
16. Pechelt L., Typke R., *An interface for melody input*, "ACM Transactions on Computer-Human Interaction", Vol. 8, No. 2, 2001, 133–149, DOI: 10.1145/376929.376978.
17. Półrolniczak E., Łazoryszczak M., *Quality assessment of intonation of choir singers using F0 and trend lines for singing sequence*, "Metody Informatyki Stosowanej", PAN, Nr 4, 2011, 259–268.
18. Ramirez J., Segura J., Benitez C., de la Torre A., Rubio A., *Efficient voice activity detection algorithms using long-term speech information*, Speech Communication, Vol. 42, No. 3–4, 2004, 271–287, DOI: 10.1016/j.specom.2003.10.002.
19. *Serato Pitch 'N' Time Pro* – official website, http://www.serato.com/products/pnt/ (accessed 20 Oct. 12).
20. Stasiak B., *Follow That Tune – Adaptive Approach to DTW-based Query-by-Humming System*, "Archives of Acoustics", Vol. 39, No. 4, 2014, 467–476, DOI: 10.2478/aoa-2014-0050.
21. Wiszniewska M., *Realization of a computer application automatically correcting detuned singing*, Master Thesis at Multimedia Department, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 2013.
22. Yu H.-M., Tsai W.-H., Wang H.-M., *A Query-By-Singing System for Retrieving Karaoke Music*, "IEEE Transactions on Multimedia", Vol. 10, No. 8, 2008, 1626–1637, DOI: 10.1109/TMM.2008.2007345.
23. Zieliński T., *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, WKiŁ, Warszawa 2009 (in Polish).
24. Zölzer U., *DAFX. Digital Audio Effects*, Wiley, New York 2005.
25. Żwan P., *Automatic singing quality recognition employing artificial neural networks*, "Archives of Acoustics", Vol. 33, No. 1, 2008, 65–71.

Aplikacja do automatycznej detekcji i korekcji fałszu w śpiewie

Streszczenie: Artykuł opisuje aplikację służącą do automatycznej detekcji i korekcji fałszu w śpiewie. W kolejnych krokach przedstawione są obserwacje stanowiące podstawę pracy, założenia, ograniczenia, perspektywy oraz wykorzystane algorytmy. Szczegółowy opis dotyczy przeprowadzonych eksperymentów i uzyskanych wyników. Ostatnia część poświęcona jest dyskusji na temat nowych możliwości odkrytych podczas badań oraz kierunków dalszych prac.

Słowa kluczowes: detekcja fałszu, korekcja fałszu, transpozycja częstotliwości dźwięku, rozstrojenie, dopasowanie melodii, ocena śpiewu

Małgorzata Michalska, MSc
malgorzata.michalska@epfl.ch

She obtained the BSc degree in Electronics and Telecommunications (2012) and the MSc in Computer Science (2013) at Gdańsk University of Technology, both with honors. Currently she is a PhD student in Electrical Engineering School of École Polytechnique Fédérale de Lausanne, Switzerland. Her research topics involve optimization and modelling of multimedia applications based on dataflow programming networks. Her special interest, combined with personal activities, focuses on audio processing with an emphasis on various analysis of singing.