Proceedings of the Federated Conference on Computer Science and Information Systems pp. 353–356 DOI: 10.15439/2017F182 ISSN 2300-5963 ACSIS, Vol. 11

Extraction of specific data from a sound sample by removing additional distortion

Dawid Połap
Institute of Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: Dawid.Polap@polsl.pl

***Abstract*—Correct identity recognition based on a voice sample must deal with many problems such as too big or small distance from the microphone, noise or abnormal voice. Hoarseness, coughing or even stuttering can also be encountered as disturbance of the voice. Research on new aspects of intelligent processing for voice brings possibilities to use intelligent methods to increase efficiency in processing and quality of record. In this paper, a spectrogram analysis for the detection of specific data and remove these distortions in the sample is presented. The proposed solution has been tested and discussed for real use in identity verification systems.**

I. Introduction

VOICE and sound are present in digital form which is used in multimedia systems and various aspects of computer processing. Sound signal analysis finds many uses in practical applications such as identity verification or even analysis of specific commands used for controlling different hardware. This is the main motor for signal research to allow quick and accurate signal analysis and good data extraction. For this purpose, different tools are used like statistical artificial intelligence techniques. Sound analysis is very often associated with the processing of sound files or even images. In [1], the idea of using heuristic algorithms in conjunction with the neural classifier has been introduced as a tool for identity verification process. Again in [2], the authors presented the use of modified mellin transform for detection of selected voice disorders.

Processing sound samples is not only a distortion analysis or identity verification, but also analysis of information, for example in the form of singing. It is important to distinguish the singing from other forms of speaking [3]. Moreover, there is an algorithm to check if a recorded sound can be classified as a spoken form [4]. All these techniques and applications find their place in different systems where voice is an important element. In [5], smartphone user identity was presented where gait characterization was used and the same idea can be used with voice analysis. Again in [6], voice-based authentication for the purpose of application in mobile phones was discussed. Large systems need huge databases where samples and data will be kept. During using the data contained in the database, it is often necessary to select a variety of them by using search and sort algorithms [7], [8]. Of course, algorithm is needed in the construction of large systems but also programming language are very important. In [9], [10], the development of human-friendly notation was shown.

In this paper, the algorithm for simple detection of sound samples distortion is presented. The main use of this algorithm is based on the analysis of samples in identity verification systems.

II. Voice defect detection

The ideal voice-based user verification system should almost always handle verification. The work of the system shall be possibly independent, that means we can expect the system to work despite distortion, noise, and sample size. Unfortunately, the number of different problems from detection methods, analysis to classification that this type of tool must handle is large therefore it must be continuously developed and improved for practical use. Let us think about practical implementation in a company to grant access to workers. In case of a large company, an employee i.e. coming to work must declare the name to the voice receiver. In order not to cause queues and unnecessary problems, the software should verify the person in spite of any possible voice transformations such as hoarseness or cough.

A. Algorithm for detecting selected distortion of the voice sample

The proposed solution is based on creating a collection of voice samples with the same information from one person. For these, implemented system is trained to evaluate input signals. In the process we can distinguish two stages - pattern preparation and pattern analysis. In the methodology SURF method described in [11] is used to extract key-points from recorded spectrograms. The process is presented in Algorithm 1.

B. Pattern preparation

At the beginning of processing, a spectrogram is created for each sample. The spectrogram is a graph of the amplitude spectrum, which is formed by the use of a short-time Fourier transform (STFT). The process of creating spectrograms is based on the principle of calculating the short-time fast

Algorithm 1 Pattern creation process

```
Start,
Define the radius $r$ and limit value $\phi$,
Load all audio samples,
for each audio sample do
    Create spectrogram,
    Find key-points using SURF algorithm,
    Create white bitmap called pattern,
    for each key-point do
        Set key-point,
        Draw and fill the circle with radius $r$ with key-point
        as center,
    end for
    Save pattern,
end for
Create an array filled with 0,
for each pattern do
    for each pixel on the pattern do
        if pixel is black then
            Increase the corresponding value in the array by 1,
        end if
    end for
end for
Create white bitmap which will be called a general pattern,
for each $value$ in the array do
    if $value \geq \phi$ then
        Set black pixel,
    else
        Set white pixel,
    end if
end for
Return a general pattern,
Stop.
```

Fourier transform. According to [12] these are most usefuly represented by the following equation

$$\begin{aligned} STFT\{x[n]\}(m,\omega) &\equiv X(m,\omega) \\ &= \sum_{n=-\infty}^{\infty} x[n]w(n-m)\exp(-j\omega n), \end{aligned} \tag{1}$$

where $x[n]$ is the discrete signal and function w is the Hann window defined as

$$w(n) = 0.5\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right). \tag{2}$$

In this way, the defined transformation allows the spectrogram to be calculated by

$$spectogram\{x(t)\}(\theta,\omega) \equiv |X(\theta,\omega)|^2. \tag{3}$$

In the next step, we find key points in these images using SURF (Speeded Up Robust Features) algorithm [11]. It uses a Hessian to find points of interest and indicates local changes around the area. It is defined as

$$H(x,\omega) = \begin{bmatrix} L_{xx}(x,\omega) & L_{xy}(x,\omega) \\ L_{xy}(x,\omega) & L_{yy}(x,\omega) \end{bmatrix}, \tag{4}$$

where $L_{xx}(x,\omega)$ is the convolution of the image with the second derivative of the Gaussian and can be calculated as

$$L_{xx}(x,\omega) = I(x)\frac{\partial^2}{\partial x^2}g(\omega), \tag{5}$$

$$L_{yy}(x,\omega) = I(x)\frac{\partial^2}{\partial y^2}g(\omega), \tag{6}$$

$$L_{xy}(x,\omega) = I(x)\frac{\partial^2}{\partial xy}g(\omega), \tag{7}$$

where $g(\omega)$ is the Gaussian kernel. $I(x)$ is an integral image where x is the point that stores the sum value of all pixels in the neighborhood calculated by

$$I(x) = \sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} I(x,y). \tag{8}$$

The whole detection algorithm is based on non-maximal-suppression of determinant of Hessian matrix defined in (4) Then, the extremes are found, which can be considered as key-points. The next step of SURF is the description which is based on Haar wavelet. Mentioned determinant can be calculated as

$$det(H_{approximate}) = D_{xx}D_{xy} - (wD_{xy})^2, \tag{9}$$

where w is the weight, and D_{xx} refers to $L_{xx}(x,\omega)$.

Having key-points of all the samples, a pattern can be created for each of them. For each spectrogram, we create a new image, where we transfer the key-points. Further, for each point, a neighborhood is created in the form of circle - key-point is a center and r is a radius.

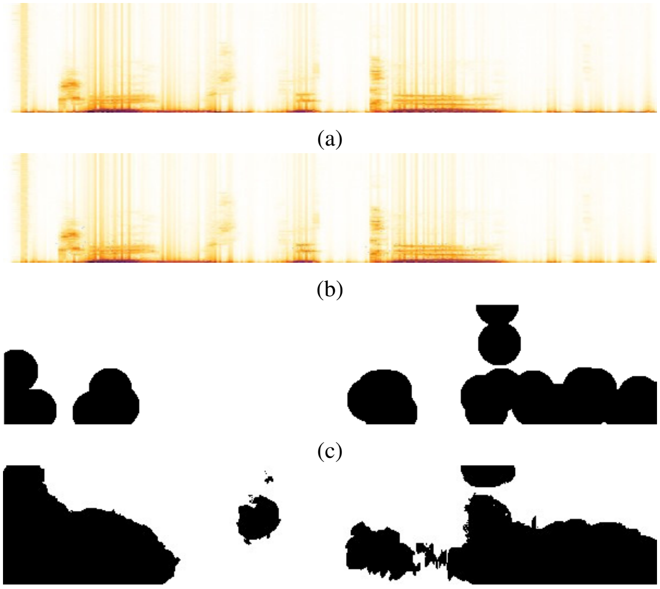
C. Pattern analysis

The next step in proposed methodology is creation of a general pattern and then comparison of this pattern to extract only interesting information. Such extraction will allow us to remove unnecessary areas for instance distortions such as coughing.

In the first step, we create an array corresponding to the image size $n \times m$. Each cell is equal to 0. For each pattern, the pixel value is checked - if it is black, the corresponding cell in the array is incremented by 1. After checking all the images, we define the minimum value ϕ which will represent the limit value in the general pattern creation process.

New bitmap of size $n \times m$ is created. If the value of the cell in the array is greater or equal ϕ, a black pixel is set. Otherwise, the pixel is white. The effect of this algorithm is shown in the Fig. 1, and the whole idea is described in Algorithm 1.

Analysis is understood as fitting a new sample to the general pattern. Once the pattern is matched, the rest of the samples can be removed because it contains noises or other misleading information for the verification process. If the sample is larger



(a)

(b)

(c)

(d)

Fig. 1: In the figures we can: (a) original spectrogram, (b) spectrogram with key-points found by SURF algorithm, (c) pattern created on the basis of key points, (d) pattern created on the basis of all the spectrograms.

than the pattern, it is shifted from left to right to find at least 80% of the key-points within the general pattern. If it is smaller, the sample is resized to the pattern size and the position of key-points are verified. The example of the action is shown in Fig. 3.

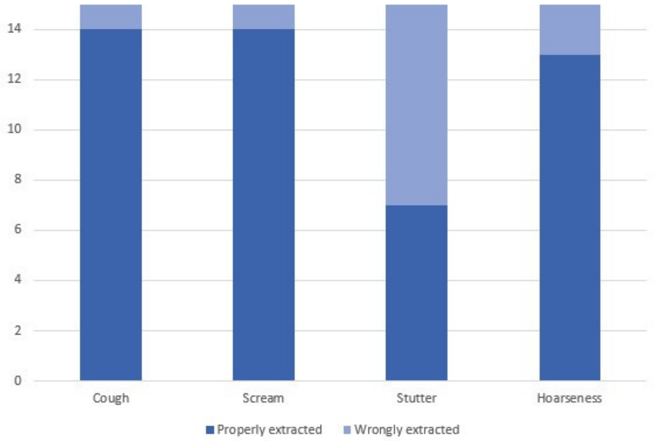


Fig. 2: Data extraction results due to selected voice imperfections.

III. Experiments

A small database of 100 different samples of *Han Solo* sentences was created. 40 samples were recorded without any voice defect to create the original pattern matching the person. The remaining samples, for test purposes, were divided into four groups of 15 samples, namely: cough, scream, stutter and hoarseness.

Tests were performed because of the different parameter values $r \in \langle 1, 40 \rangle$ and $\phi \in \langle 5, k \rangle$ where k is the number of samples without any defect. In the case of a radius, too little value caused no pattern matching. On the other hand, too big value caused the pattern to cover the sample. It turns out that the value of the radius should be matched by the size of the sample – for samples of 940x300. The best results were obtained with a radius of 20 pixels. Adjusting ϕ value is much simpler. The greater the value, the less points will be transferred to the general pattern. The tests showed that the number of points should not be too high, because the voice may have different distortions, such as the distance from the microphone. The best results were obtained for $\phi = 10$.

For such selected parameters, the proposed technique was tested for each voice sample with a defect. The results are shown in Fig. 2. Extraction of the name in most samples labeled as stuttering failed what can be caused by wrong choice of parameters. In other cases, extraction was successful for almost every sample. For such data, the efficiency of the method is 80%, which is not the best result. However, a greater number of samples as well as a better selection of values could improve the performance index.

IV. Conclusions

In the paper, the application of graphic processing for removing unnecessary data from the verification sample of a voice spectrogram has been demonstrated. The proposed solution is intended for identity verification systems based on a short audio sample containing only the name of the person. Technique was tested on a small database of sound samples, resulting in 80% efficiency in extraction of these specific information. Of course, such a solution would reduce the amount of calculations for classifiers because the sample will not only be smaller but contain only needed information. Unfortunately, there are also some imperfections that affect the percentage efficiency of the method, that is, manually adjusting the value of the parameters that significantly affect the process of matching a sample to the pattern.

In future research, adjustment of parameter values will be analyzed for the best adjustment for many people in the database and others datasets. The method will be subjected to more analysis in order to increase efficiency. Furthermore, the removed imperfections in the samples may be subjected to a certain classification for analysis.

V. Acknowledgment

Authors acknowledge contribution to this project to the Diamond Grant No. 0080/DIA/2016/45 funded by the Polish Ministry of Science and Higher Education.

References

[1] D. Połap, "Neuro-heuristic voice recognition," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 487–490.

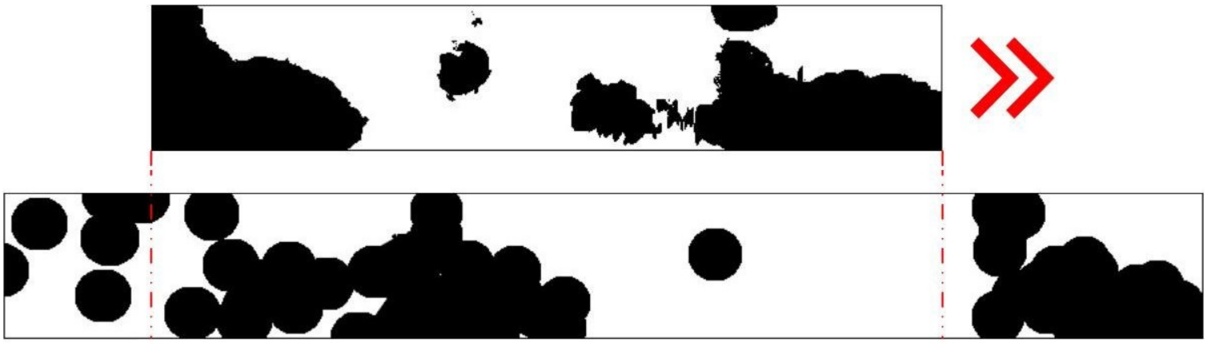


Fig. 3: The process of matching a sample to a pattern, where of the samples is fitted into a general pattern used for key-points matching to remove coughing, distortion and other noises.

[2] C. R. Francis, V. V. Nair, and S. Radhika, “A scale invariant technique for detection of voice disorders using modified mellin transform,” in *Emerging Technological Trends (ICETT), International Conference on*. IEEE, 2016, pp. 1–6.

[3] S. D. You, Y.-C. Wu, and S.-H. Peng, “Comparative study of singing voice detection methods,” *Multimedia Tools and Applications*, vol. 75, no. 23, pp. 15 509–15 524, 2016.

[4] S. S. Kumar and K. S. Rao, “Voice/non-voice detection using phase of zero frequency filtered speech signal,” *Speech Communication*, vol. 81, pp. 90–103, 2016.

[5] R. Damaševičius, R. Maskeliūnas, A. Venčkauskas, and M. Woźniak, “Smartphone user identity verification using gait characteristics,” *Symmetry*, vol. 8, no. 10, p. 100, 2016.

[6] R. Johnson, W. J. Scheirer, and T. E. Boult, “Secure voice-based authentication for mobile devices: vaulted voice verification,” in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 87 120P–87 120P.

[7] Z. Marszałek, “Performance test on triple heap sort algorithm,” *PUBLISHER UWM OLSZTYN 2017*, vol. 20, no. 1, pp. 49–61, 2017.

[8] ——, “Novel recursive fast sort algorithm,” in *International Conference on Information and Software Technologies*. Springer, 2016, pp. 344–355.

[9] M. Nosál’, J. Porubän, and M. Sulír, “Customizing host ide for non-programming users of pure embedded dsls: A case study,” *Computer Languages, Systems & Structures*, 2017.

[10] S. Chodarev, “Development of human-friendly notation for xml-based languages,” in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 1565–1571.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[12] P. Welch, “The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.