DYSFLUENT SPEECH DETECTION BY IMAGE FORENSICS TECHNIQUES

Juraj Pálfy[1,2], *Sakhia Darjaa*[1*]

Jiří Pospíchal[2]

[1]Slovak Academy of Sciences
Institute of Informatics
Bratislava, Slovakia

[2]Slovak University of Technology
Faculty of Informatics and Information Technologies
Bratislava, Slovakia

ABSTRACT

As speech recognition has become popular, the importance of dysfluency detection increased considerably. Once a dysfluent event in spontaneous speech is identified, the speech recognition performance could be enhanced by eliminating its negative effect. Most existing techniques to detect such dysfluent events are based on statistical models. Sparse regularity of dysfluent events and complexity to describe such events in a speech recognition system makes its recognition rigorous. These problems are addressed by our algorithm inspired by image forensics. This paper suggests our algorithm developed to extract novel features of complex dysfluencies. The common steps of classifier design were used to statistically evaluate the proposed features of complex dysfluencies in spectral and cepstral domains. Support vector machines perform objective assessment of MFCC features, MFCC based derived features, PCA based derived features and kernel PCA based derived features of complex dysfluencies, where our derived features increased the performance by 46% opposite to MFCC.

Index Terms— Dysfluency detection, image forensics, speech

1. INTRODUCTION

The communication disorder known as stuttering is characterized by dysfluencies, which are disruptions in smooth flow of speech[1]. Speech Language Pathology (SLP) divides dysfluent events into categories. Well-known dysfluency categories include hesitations (e.g. pauses), prolongations (e.g. lllike) and repetitions. Repetitions have specific categories: syllable repetitions (e.g. re re research), word repetitions (e.g. my my love) and phrase repetitions (e.g. I do my, I do my work). From the other side, unlike read speech, spontaneous speech contains high rates of disfluencies (e.g. hesitations, phrase repetitions)[2]. Dysfluencies such as prolongations, repetitions and hesitations add needless lexical information to talkers' conversation. Information redundancy caused by common dysfluencies influence the Automatic Speech Recognition (ASR) system performance negatively. Therefore one of many aspects of dysfluency detection in speech technology is to augment the ASR system to decrease the recognition errors.

Dysfluent speech recognition attracts considerable attention in Speech Language Pathology (SLP), where objective evaluation of stuttered speech is still under development. Researchers with technical background invest their effort to develop a framework to resolve this problem for SLP. In [3], a parameter for global evolution of dysfluent speech is defined on the basis of spectral changes and with Bayesian detector defines a parameter for globally evaluation of dysfluent speech, to evaluate the efficiency of SLP therapy. Phoneme repetitions were classified by Hidden Markov Models on the basis of Mel-Frequency Cepstral Coefficients (MFCC) [4]. MFCC with Dynamic Time Warping (DTW) was used by [5] to study syllable repetition detection accuracy comparing various dimensional MFCC feature vectors (12, 13, 26 and 39 dimensional). In [6] the performance of Least Square Support Vector Machine is examined with Sample Entropy derived from Bark scale, Erb scale and Mel scale for distinguishing the prolongation and repetitions events in speech. MFCC and Linear Predictive Cepstral Coefficients (LPCC) feature extractions with k-Nearest Neighbor (kNN) and Linear Discriminant Analysis (LDA) classifiers were compared to recognise repetitions and prolongations in stuttered speech. In [7] hierarchical Artificial Neural Networks (ANN) were introduced to support stuttered speech recognition process.

Another aspect for disfluent speech recognition in the field of speech processing is to moderate its effect in ASR. In this case researchers try to improve the accuracy in current ASR by introducing information on the unknown phenomenon for ASR. Disfluencies are observed in spontaneous speech with non-clinical speakers generally; this type of events is tagged with letter 'i' in word disfluency (for people with stuttering, letter 'y' is used). Early detection of disfluencies studied in [8] shows that the juncture phenomena which occur between words in fluent speech are usually absent at the interruption point in disfluent utterances. Text and prosody information

*This contribution was supported by Research & Development Operational Programme funded by the ERDF project RPKOM, ITMS 26240220064. We thank Erika Pálfy for her help concerning communication disorders.

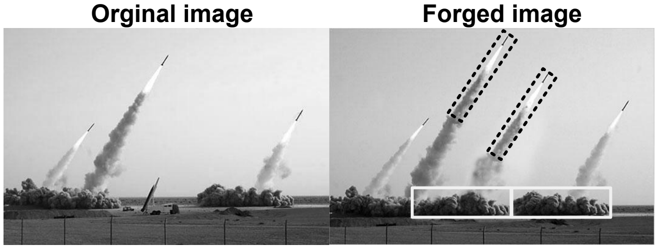

Fig. 1. Specimen for copy-move forgery (prepared in accordance with [11])

(duration, F0, energy and pause) were used in [9] with Conditional Random Fields to locate interruption point (i.e. IP), the point in time at which the speaker breaks off from the original utterance. Feature extraction technique [9], but on the basis of Weighted Finite State Transducers (WFST), was implemented to detect filled pauses and reparandum region of repeats and repairs [10].

The 'complex' dysfluencies specified as a chaotic mixture of dysfluent events (e.g. prolongation combined with various types of repetition) are frequent in stutterers' speech. In dysfluent speech recognition, the common methodology is to fix a window (e.g. 200 ms, 500 ms, 800 ms) and build a dysfluency recognition system (e.g. kNN, ANN) which can recognize the 'simple' dysfluent events in a fixed interval of speech. Annotations of dysfluent speech show that dysfluent events frequently do not fit the fixed window, but are dynamically distributed throughout much longer 2-4 s intervals. The above-mentioned problems are addressed by our following approach, inspired by image forensics techniques. There are several types of image tampering, but hiding with cloning some objects from natural images is a common form of digital image tampering, known as copy-move forgery [12]. An example of copy-move forgery is shown in Fig. 1, where the original image shows three missiles and the forged image contains four missiles. On the basis of SLP and our observation during transcription of English and Slovak stutterers, the following two facts have been achieved. Prolongation is a recursion of one atomic structure (e.g. phoneme) in a signal space. Repetition is a recursion of many grouped atomic structures (e.g. syllables, words) with random edits (e.g. pauses, filled pauses) in a signal space. Analogously to copy-move forgery, in Fig. 2, in signal space, interval *A* is repeated ('cloned') in interval *B*. Along with developing our algorithms, we borrow the copy-move forgery detection techniques to find application of this technique in dysfluent speech analysis.

2. METHODOLOGY

2.1. Database

The authors [13] used 12 selected audio recordings *working set* from University College London Archive of Stuttered Speech (UCLASS). We used a subset of this working set with 22.05 KHz sampling rate and the total of 19:32 min playing time for our experiments. The subset of recordings was annotated by English SLP; we made manual time alignment of lexical content and inserted the appropriate SLP labels to the dysfluent events (additional information about data and presented algorithm: `sites.google.com/site/georgepalfy/`).

2.2. Feature Extraction

Works dealing with dysfluency detection frequently use Fourier transformation to analyze spectrum and to compute derived homomorphic features, for example MFCC, LPCC, PLP [14].

Principal Component Analysis (PCA) is a well established technique for extracting structures from high-dimension data sets. This is performed by extracting *eigenvalues*. PCA is an orthogonal transformation used to describe our data. The new coordinate values by which we represent our data are called *principal components*. It is often the case that a small number of principal components is sufficient to account for most of the structure in the data [15].

Let a data set M consist of centered observations $x_k \in R^n, k = 1, \ldots, M$ and $\sum_{k=1}^{M} x_k = 0$; the covariance matrix corresponding to this data set is given by

$$C = \frac{1}{M} \sum_{j=1}^{M} x_j x_j^T. \tag{1}$$

Diagonalizing C, we obtain the principal components, which are the orthogonal projections onto the eigenvectors, obtained by solving the eigenvalue equation

$$\lambda v = Cv \tag{2}$$

where $\lambda \geq 0$ and $v \in R^n \setminus \{0\}$. As

$$Cv = \frac{1}{M} \sum_{j=1}^{M} (x_j \cdot v) x_j, \tag{3}$$

all solutions for v must be a linear combination of $x_1, \ldots, x_M$, which can be expressed as

$$v = \sum_{j=1}^{M} \alpha_j x_j \tag{4}$$

where α_j, for $j = 1, \ldots, M$, are coefficients [16]. Kernel PCA performs a non-linear transformation of the sample x in input space, $x \in R^N$, to the high-dimensional feature (dot product) space F expressed by the map

$$\Phi : R^N \to F \tag{5}$$

$$x \to X. \tag{6}$$

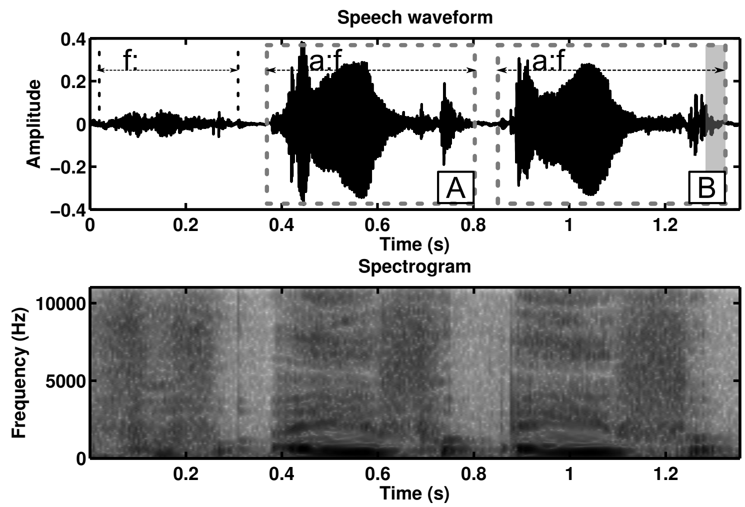

Fig. 2. Syllable repetition combined with prolongation

PCA is then performed in this high-dimensional space. The covariance matrix in feature space can be expressed as Equation (1) with the help of mapping $\Phi(x_j)$ samples into the feature space.

$$\bar{C} = \frac{1}{M}\sum_{j=1}^{M}\Phi(x_j)\Phi(x_j)^T \tag{7}$$

The goal is to perform the eigendecomposition of V, that is,

$$\lambda V = \bar{C}V. \tag{8}$$

By the definition of V, it can be shown that v lies in the span of $\Phi(x_1), \ldots, \Phi(x_M)$. We can consider the equivalent equations

$$\lambda(\Phi(x_k)\cdot V) = (\Phi(x_k)\cdot \bar{x_k}\cdot \bar{C}V) \tag{9}$$

$$V = \sum_{j=1}^{M}\alpha_j\Phi(x_j) \tag{10}$$

$$K_{i,j} = k(x_i, x_j). \tag{11}$$

Combining Equation (9), Equation (10) and defining $M \times M$ kernel matrix $K = (\Phi(x_i)\cdots\Phi(x_j)$ leads to the formulation

$$M\lambda K\alpha = K^2\alpha \tag{12}$$

$$M\lambda\alpha = K\alpha \tag{13}$$

where α is a column vector with the entries $\alpha_1, \ldots, _M$. Equation (13) is solved by the eigenvectors α^k. Requiring V^k to be normalized leads to the normalization condition $\lambda_k(\alpha^k)\alpha^k = 1$. Let x be a test point, with an image $\Phi(x)$ in F; then

$$(V^k\cdot\Phi(x)) = \sum_{i=1}^{M}\alpha_i^k(\Phi(x_i)\dot{\Phi}(x)) \tag{14}$$

may be called its nonlinear principal component corresponding to Φ [15], Schölkopf et al. in [17, 18, 16]. In this paper for non-linear transformation the Gaussian kernel function

$$K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{15}$$

is applied. To calculate the speech features (MFCC), we maintain the standard method used in Hidden Markov Model Toolkit: Hamming window length (0.023 s), overlapping adjacent frames (0.01 s) and number of bandpass filters (20). Each frame was processed with attributes to conserve MFCC vector with 13 coefficients.

Prior to PCA and kernel PCA transformation of speech, the magnitude spectrum (with 256 elements) was calculated over frames and windows identical to MFCC computation. Every magnitude spectrum computed in this way was then divided into sixteen equal intervals. These fixed intervals of magnitude spectrum were then represented as square matrix A ($n \times n$ elements).

$$A_{16,16} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,16} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,16} \\ \vdots & \vdots & \ddots & \vdots \\ a_{16,1} & a_{16,2} & \cdots & a_{16,16} \end{pmatrix} \tag{16}$$

Elements ($a_{n,n}$) of matrix A correspond to the elements of magnitude spectrum (e.g. $a_{1,1}$ is the 1st element of magnitude spectrum, $a_{2,16}$ is the 32nd element of magnitude spectrum). Matrix A built from elements of magnitude, in next step undergo PCA and kernel PCA transformation. In transformation of A to feature vector x the 1st principal component was used. After these transformations matrix $A_{16,16}$ was represented by feature vector x containing its 16 dominant elements. Before further processing, feature vectors x were collected to $m \times n$ dimensional feature matrix X, where rows

(m) of X represent n-dimensional feature vectors x. MFCC vectors also conserved this arrangement, but their dimension was $m \ldots 13$.

The matrix sorting and distance metrics are widely used techniques in copy-move detection [19], [12]. We expect that after sorting matrix X, in the new ordered matrix S the most similar feature vectors become neighbors. This condition allows to effectively reduce the total number of examined feature vector pairs in sorted matrix S. The search space from the X is then reduced into subregions of S, where the search is executed on i-th selected feature vector and its n neighboring vectors (subregion) in S.

The matrix S originated from X after sorting its rows in increasing order. D_i and D_e denote two different distance metrics. D_i is the minimal index distance (e.g. index offset) between neighboring vectors and D_e corresponds to the maximal Euclidean distance between feature vectors. N_n gives the number of searched neighbors and N_r denotes the number of rows in matrix X. The algorithm (*redDet*) output R contains

Algorithm 1 Redundancy detection (redDet)

```
Sort rows of X to obtain S.
for k ← 1, N_r − N_n do
    Select k-th vector s_k from matrix S.
    Select N_n neighbors for vector s_k.
    Compute D_i and D_e for s_k and its neighbors.
    Save vector pair positions, D_i and D_e to R_k.
end for
```

redundant vector pair positions and their distance metrics. For the purpose of detecting dysfluencies in continuous speech, R is postprocessed to become R''.

$$R'_{i,5} = \begin{pmatrix} r'_{1,1} & r'_{1,2} & \cdots & r'_{1,5} \\ r'_{2,1} & r'_{2,2} & \cdots & r'_{2,5} \\ \vdots & \vdots & \ddots & \vdots \\ r'_{i,1} & r'_{i,2} & \cdots & r'_{i,5} \end{pmatrix} \tag{17}$$

Postprocessing begins with arranging the vector positions between vector pairs in increasing order. After this step the first column ($r'_{i,1}$) of R' contains vector positions lower than those in the second column ($r'_{i,2}$) and $r'_{i,1} < r'_{i,2}$ holds. This ordering of vector pairs position gives us the time order in the 1st column ($r'_{i,1}$). In the next postprocessing step, all the unique vectors of R' were found and their vector frequencies (R' reduced to matrix R'') were computed. For each unique vector of R'' a minimal Euclidean distance and maximal frequency of every its occurrence were chosen. After the final step, after ordering its columns the primeval R becomes R'. Posterior R' was then reduced to obtain R''. R'' holds vector pairs position (unique vectors in the 1st column $r''_{i,1}$ and its pairs position in the 2nd column $r''_{i,2}$), D_i, D_e and frequencies. Our algorithm was evaluated by experimentally determined parameters $D_i > 100$ ms, $N_n = 60$ and $D_e < 8$. According to this setup, the algorithm output (Fig. 3) shows an example of outcome on the interval, where dysfluent event appears (repetition of phrases). Vectors denoted by *A* ($r''_{i,1}$) are repeated in position *B* ($r''_{i,2}$).

The algorithm processed 5 s intervals with 2.5 s overlaps. For every 5 s long speech interval the following three features were computed between feature vectors: (1) *index distance*, (2) *Euclidean distance* (*euc.*), and (3) *frequency* (*freq.*). The last feature was calculated in the postprocessing step, where for every unique vector in R' its *frequency* was determined. In R'', repeated regions in the speech are then observed as vectors with high occurrence.

2.3. Evaluation

To be able to compare the performance of studied features (algorithm output *Mfccrep*, *Pcarep* and *Kpcarep*) to commonly used MFCC coefficients, objective assessment was given by using feature vectors as inputs for SVM. SVM and their variants and extensions, often called kernel methods, have been studied extensively and applied to various pattern recognition and function approximation issues [20]. According to [21], in our experiment, we used sigmoid kernel function

$$k(x, y) = \tanh\left(\gamma x^T y + r\right). \tag{18}$$

During the classifier design process, one of the intermediate steps is the measurement of data class separability, which we prove with correlation between two classes (fluent and dysfluent speech). In the next stage we studied the data characteristics for two classes with Mann-Whitney U-test. Nonparametric Mann-Whitney U-tests examine the equality of class medians of random variables X, Y. We use confusion matrix to measure SVM models' performance. In addition to *accuracy*, we compute *sensitivity* and *specificity* of confusion matrix [22].

3. RESULTS

Table 1 and Table 2 compare the features computed from MFCC with Dynamic Time Warping (*MFCC DTW*) and the features computed using our algorithms (*Pcarep* - PCA based derived features, *Kpcarep* - Kernel PCA based derived features and *Mfccrep* - MFCC based derived features). Low correlation coefficients r in Table 1 refer to the fact that there is a low linear dependence between features computed from fluent and dysfluent intervals of speech. There is negative Spearman rank correlation between our proposed features *Pcarep freq.*, *Kpcarep freq.* and *Mfccrep freq.* which is related to observations that frequency of similar feature vectors in fluent intervals is lower than that of similar feature vectors in dysfluent intervals of speech. In case of class separability it is considered that *p-values* except *MFCC DTW* and *Pcarep euc.* are below 5% significance level. According to significance

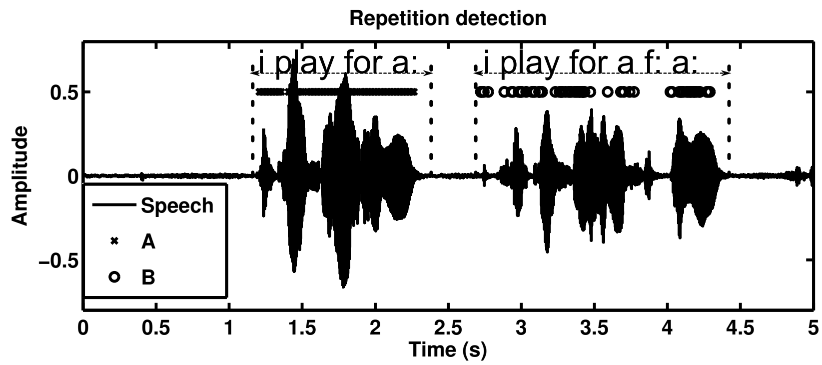

Fig. 3. Algorithm output in case of dysfluent event (phrase repetition)

Table 1. Spearman rank correlations between fluent and dysfluent events groups

Feature	r	p-value
MFCC DTW	0.0356	0.3654
Pcarep euc.	0.0366	0.0606
Pcarep freq.	-0.5678	0.0
Kpcarep euc.	0.0627	0.0013
Kpcarep freq.	-0.5689	0.0
Mfccrep euc.	-0.0409	0.0362
Mfccrep freq.	-0.5738	0.0

Table 2. Mann-Whitney U-test between fluent and dysfluent events groups

Feature	h	p-value
MFCC DTW	1	0.0
Pcarep euc.	0	0.0720
Pcarep freq.	1	0.0
Kpcarep euc.	0	0.5008
Kpcarep freq.	1	0.0
Mfccrep euc.	1	0.0
Mfccrep freq.	1	0.0

of Spearman correlation values, the features clearly separate fluent and dysfluent events groups.

Data characteristics study results of the proposed features are shown in Table 2. Nonparametric Mann-Whitney U-tests in Table 2 are significant, when their *p-values* are below 5% level. h values specify accepted hypotheses. We fail to reject hypotheses $h = 0$, only for features *Pcarep euc.* and *Kpcarep euc.* Rejected hypotheses $h = 0$ in other features describe that features with $h = 1$ do not have equal medians. According to the test results, features with $h = 1$ are characterized by unequal data distribution for fluent and dysfluent features groups.

Feature	Sensitivity	Specificity	Accuracy (%)
MFCC	0.508	0.496	50.2
MFCC DTW	0.739	1	85.4
Pcarep euc.	0.55	0.469	50.9
Pcarep freq.	1	0.897	94.8
Kpcarep euc.	0.679	0.328	50.1
Kpcarep freq.	1	0.904	95.1
Mfccrep euc.	0.623	0.338	47.8
Mfccrep freq.	1	0.926	96.2

Table 3. Support vector machines testing results

We divided the data into training (80 %) and testing (20 %) sets. In the next step, we trained eight individual SVMs with sigmoid kernel function. In Table 3, the *MFCC*, *MFCC DTW*, *Pcarep*, *Kpcarep* and *Mfccrep* features are evaluated. For MFCC feature, we get 50.2 % accuracy. *MFCC DTW* achieved 85.4 % accuracy. SVM model for *Mfccrep freq. sensitivity* did not make any *false negative* prediction and produced only 8 % *false positive* predictions. According to the represented classification results shown in Table 3, the *Mfccrep freq.* maintains the upper limit with 97.6 % accuracy.

4. CONCLUSION

Using data class separability power and examining the data characteristics, we compared MFCC features to our derived speech features computed from spectral and cepstral domain.

We developed the algorithm created specially for complex repetition detection. Our paper shows that the technique used in image forensics to detect copy-move forgeries offers an alternative way of dysfluency detection.

New features of complex repetitions computed from our algorithms outputs were statistically analyzed. Objective assessment of new features, DTW algorithm and MFCC were compared by Support Vector Machines (SVM). SVM trained by MFCC features to recognise repetitions show 50.2 % accuracy. In case of our features based on MFCC, SVM accomplished 96.2 % accuracy on the equivalent speech data sets.

In the future, the algorithm may be further studied by optimizing its parameters (e.g. with evolutionary computation) on larger stuttering database to obtain more robust and precise performance.

5. REFERENCES

[1] K.G. Shipley and J.G. McAfee, *Assessment in Speech-Language Pathology: A Resource Manual*, Delmar Cengage Learning, USA, fourth edition, 2008.

[2] E. Shriberg, "To errrr is human: ecology and acoustics of speech disfluencies," *Journal of the International Phonetic Association*, vol. 31, pp. 153–169, 6 2001.

[3] T. Lustyk, P. Bergl, R. Čmejla, and J. Vokřál, "Change evaluation of Bayesian detector for dysfluent speech assessment," in *International Conference on Applied Electronics 2011*, Pilsen, Czech Republic, 2011, pp. 231–234, University of West Bohemia.

[4] M. Wiśniewski and W. Kuniszyk-Jóźkowiak, "Automatic detection and classification of phoneme repetitions using HTK toolkit," *Journal of Medical Information and Technologies*, vol. 17, 2011.

[5] K. M. Ravi Kumar and S. Ganesan, "Comparison of Multidimensional MFCC Feature Vectors for Objective Assessment of Stuttered Disfluencies," *Int. J. Advanced Networking and Applications*, vol. 2, no. 5, pp. 854–860, 2011.

[6] M. Hariharan, V. Vijean, C. Y. Fook, and S. Yaacob, "Speech stuttering assessment using sample entropy and Least Square Support Vector Machine," *12 IEEE 8th International Colloquium on Signal Processing and its Applications*, pp. 240–245, mar 2012.

[7] I. Świetlicka, W. Kuniszyk-Jóźkowiak, and E. Smołka, "Hierarchical ANN system for stuttering identification," *Computer Speech & Language*, vol. 27, no. 1, pp. 228–242, jan 2013.

[8] R. J. Lickley, "Juncture cues to disfluency," *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on Spoken Language Processing*, 1996.

[9] Y. Liu, E. Shriberg, A. Stolcke, and M. Harper, "Comparing HMM,MaximumEntropy, and Conditional Random Fields for Disfluency Detection," in *Proceeedings of the European Conference on Speech Communication and Technology*, 2005.

[10] S. Maskey, B. Zhou, and Y. Gao, "A Phrase-Level Machine Translation Approach For Disfluency Detection Using Weighted Finite State Transducers," in *Interspeech 06*, 2006, pp. 749–752.

[11] M. Nizza and P. J. Lyions, "In an Iranian Image, a Missile Too Many," Available at: http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/, July 10, 2008, 9:16 am, [Online; Cited July 2, 2013].

[12] Y. Cao, T. Gao, L. Fan, and Q. Yang, "A robust detection algorithm for copy-move forgery," *Forensic Science International*, vol. 214, no. 1-3, pp. 33–43, jan 2012.

[13] P. Howell, S. Davis, and J. Bartrip, "The University College London Archive of Stuttered Speech (UCLASS)," *Journal of speech, language, and hearing research : JSLHR*, vol. 52, no. 2, pp. 556–69, apr 2009.

[14] L. R. Rabiner and R. W. Schafer, "Introduction to Digital Speech Processing," *Foundations and Trends in Signal Processing*, vol. 1, no. 12, pp. 1–194, 2007.

[15] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, July 1998.

[16] A. Lima, H. Zen, Y. Nankaku, Ch. Miyajima, K. Tokuda, and T. Kitamura, "On the Use of Kernel PCA for Feature Extraction in Speech Recognition," *IEICE Trans Inf Syst*, pp. 2802–2811, 2004.

[17] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, USA, fourth edition, 2009.

[18] Ch. Leitner, F. Pernkopf, and G. Kubin, "Kernel PCA for Speech Enhancement," *INTERSPEECH*, vol. 1, no. 2, pp. 2–5, 2011.

[19] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Comput. Sci., Dartmouth College, Tech. Rep.*, pp. 1–11, 2004.

[20] S. Abe, *Support Vector Machines for Pattern Classification*, Advances in Pattern Recognition. Springer London, London, 2010.

[21] J. Pálfy and J. Pospíchal, "Recognition of Repetitions Using Support Vector Machines," in *SPA 2011 : Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings Poznan*, Poznan, 2011, pp. 79–84, IEEE Poland Section.

[22] L. Hamel, *Knowledge Discovery with Support Vector Machines*, John Wiley & Sons, Inc., Hoboken, NJ, USA, jul 2009.