Elastic-Transform Based Multiclass Gaussianization

Alexandru Paul Condurache, Alfred Mertins, *Senior Member, IEEE*

Abstract

The concept of "Gaussianization" implies a transformation aimed at changing the distribution of the input random variable to Gaussian. It has been used until now as a means to achieve independence among components in multivariate distributions that in turn was used as a tool for various purposes ranging from density estimation to normalization. In this contribution we propose Gaussianization for pattern recognition applications in support of Gaussianity assumptions made by various classifiers. Previous approaches completely ignore separability considerations, the Gaussianization being conducted over the entire data, irrespective of class affiliation and are not useful for recognition purposes. We instead propose a transform such that the output random variable is distributed according to a Gaussian mixture, where each class accounts for one mixture component. We successfully test our method on both synthetic and real data.

I. Introduction

The Gaussian assumption refers to the use of a Gaussian model to describe a random variable. This assumption is often made in practice, being justified by the central limit theorem (CLT) [6]. In its classical form, the CLT states that the mean of n independent identically distributed random variables, with finite mean and variance, has a limiting distribution for $n \rightarrow \infty$ that is Gaussian. The random variables of practical interest are measurements of real-world processes being thus available only as the result of a combination of many unobserved random influences. In the framework generated by the CLT, this combination is considered a summation, and therefore the target random variable is assumed Gaussian.

Corresponding author's e-mail: condurache@isip.uni-luebeck.de

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

A. P. Condurache and A. Mertins are with the Institute for Signal Processing, University of Lübeck, D-23538 Lübeck, Germany

The Gaussian distribution has a set of appealing characteristics related to its mathematical tractability, like, e.g., the equivalence of decorrelation and independence or the fact that all cumulants of an order larger than two are zero. These properties make inference and reasoning more easy in Gaussian environments. Therefore, the Gaussian assumption is made actually more often than needed, and a lot of effort has been put into developing methods optimally suited to such environments.

The main problem is, however, that the Gaussian assumption does not always hold. Rather than ignoring this or rethinking everything, we propose here a means of adapting to the Gaussian setup. Here we concentrate on Gaussianization for pattern recognition applications. We propose a nonlinear transform such that the class-conditional densities are Gaussian in the transformed space, thus the pdf of the data in the transformed space is a Gaussian mixture model (GMM).

Gaussianization has already been discussed for purposes such as density estimation [2], independent component analysis [9], blind source separation [3], speaker adaptation [12], adaptive filtering [7], and system identification [10]. However, all these methods are ”holistic“ in the sense that they gaussianize the entire input data irrespective of the class label, thus destroying separability and being unsuited for our purposes.

Linear transforms have the advantage of mathematical tractability, but by their inherent constraint they cannot achieve the desired Gaussianization. Nevertheless linear transforms are used to project the data onto a space where additional constraints besides Gaussianity are more likely to hold. There is a strong relationship between the Gaussian assumption and Linear Discriminant Analysis (LDA). It can be shown that when modeling the data with Gaussian distributions in a multiclass scenario, the search for a linear transform under the constraints that the covariances are equal and the means have reduced rank leads to LDA [5]. Any invertible and differentiable transformation $\boldsymbol{y} = T(\boldsymbol{x})$ modifies the statistical properties of the input data according to the well-known formula $p(\boldsymbol{y}) = p(\boldsymbol{x})\frac{1}{|T'(\boldsymbol{x})|}$, where $|T'(\boldsymbol{x})|$ is the determinant of the Jacobian matrix of the transform. It is thus clear that only a nonlinear transform can achieve the desired Gaussianization.

The probability density function of a multidimensional random variable can be either factorial or nonfactorial. A factorial multivariate pdf has the property that it can be factorized into independent components, thus $p(\boldsymbol{x}) = p(x_1, x_2, \ldots, x_N) = p(x_1)p(x_2)\cdots p(x_N)$. Under such circumstances, multivariate Gaussianization is actually a set of univariate Gaussanizations, but only if we can obtain the independent components. Conversely, if the distribution is not factorial, or if the independent components cannot be found, we need to resort to other methods. A more widely used example in this direction is the iterative Gaussianization [8] that is strongly linked to projection pursuit density estimation [4]. However, iterative

Gaussianization is a holistic method, being discussed (and shown to converge) only for $N(\mathbf{0}, \mathbf{1})$.

In this contribution we describe a Gaussianization method that works also for the case when no independent components are available and that is not holistic. At the core of our approach is an elastic transform between nonparametric and parametric pdf estimates of a training sample. We test our approach on real and synthetic data. The paper is organized as follows: in Section II we describe how multiclass Gaussianization works. In Section III we describe what experiments we have conducted to demonstrate the validity of our approach. Finally, in Section IV we present our conclusions.

II. Methods

In the supervised multiclass Gaussianization proposed here, the pdf of the available labeled training data is first estimated nonparametrically, then parametrically as a GMM with one component per class. Then, an elastic transform is computed such that the nonparametric estimate is "morphed" on the GMM, minimizing the sum of squared differences (ssd) between the two functions. In contrast to other methods, we use pdf-distance measures inspired from the performance analysis of kernel density estimators [13]. The displacement field corresponding to the elastic transform defines the way the input data should be modified such that its distribution is a GMM. Clearly the displacement field is properly defined only over a region Ω close to the support of the training sample. We extend this to $\mathbb{R}^N$ by means of the identity transform, such that data points outside this support remain unchanged. The computation of the displacement field of the elastic transform relies on the pdf estimation. The pdf is estimated at the knots of a grid laid over the support of the training sample in the feature space. In practice the grid is in the form of a hyperrectangle whose sides are double the standard deviations of the training sample in the corresponding directions. The grid is centered on the training sample and it thus spreads over the margins of the sample.

A. Pdf estimation

For our purposes we estimate the pdf from the available sample both nonparametrically and under the Gaussian parametric assumption. This procedure works irrespective of the number of classes. An example is shown in Fig. 1. In a multiclass scenario, the nonparametric estimate is carried out for the entire training set, ignoring the class labels, whereas the parametric estimate considers the class information.

1) Nonparametric estimation: For nonparametric estimation we use the Parzen estimation procedure. The estimate $\tilde{p}_{\mathfrak{D}}(\boldsymbol{x})$ is computed using the N vectors of the training sample as $\tilde{p}_{\mathfrak{D}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \gamma_h(\boldsymbol{x} -$

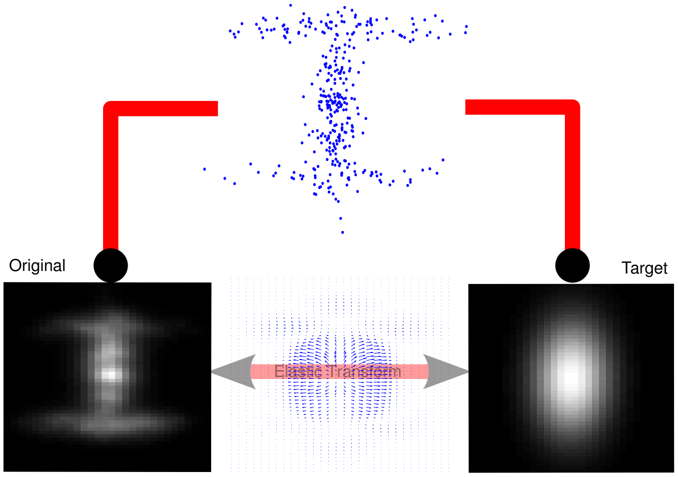

Fig. 1. Example of non-Gaussian 2D data. The Original (nonparametric pdf estimation) and the Target (Gaussian, parametric pdf estimation) are computed. The displacement field of the elastic transform (defined over the transform grid) is also shown.

$\boldsymbol{x}_i)$ where we use the Gaussian isotropic kernel $\gamma_h(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{m}{2}}\sigma^m} \exp\left(-\frac{\|\boldsymbol{z}\|^2}{2\sigma^2}\right)$ with m the size of the feature space.

Under these circumstances, $\tilde{p}_{\mathfrak{D}}(\boldsymbol{x})$ can be rewritten as $\tilde{p}_{\mathfrak{D}}(\boldsymbol{x}) = \frac{1}{N \cdot h^m} \sum_{i=1}^{N} \gamma(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h})$ with $\gamma(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{m}{2}}} \exp\left(-\frac{1}{2}\|\boldsymbol{z}\|^2\right)$ and $h = \sigma$.

The bandwidth parameter h is computed as $h = \left[\frac{8 \cdot \sqrt{\pi} \cdot R(\gamma)}{3 \cdot \mu_2^2(\gamma) \cdot N}\right]^{\frac{1}{5}} \hat{\sigma}_m$, which is related to Silverman's rule of the thumb [13]. For 1D random variables, $\hat{\sigma}_1$ is defined as $\hat{\sigma}_1 = \frac{SIQR}{\Phi^{-1}\left(\frac{3}{4}\right) - \Phi^{-1}\left(\frac{1}{4}\right)}$, with $SIQR$ the sample interquartile range, and $\Phi^{-1}(p) = \sqrt{2} \cdot \mathrm{erf}^{-1}(2 \cdot p - 1)$ for $p \in (0,1)$ the quantile function of the univariate Gaussian distribution. Here, we approximate $\hat{\sigma}_m$ with $\hat{\sigma}_1$ computed for the component of $\boldsymbol{z}$ with the largest variance. $R(\gamma)$ is defined as $R(\gamma) = \int \gamma^2(\boldsymbol{z}) d\boldsymbol{z}$ and $\mu_2(\gamma) = \int \|\boldsymbol{z}\|^2 \cdot \gamma(\boldsymbol{z}) d\boldsymbol{z}$. For 1D random variables, h minimizes the estimator error for kernels with fixed bandwidth [13].

2) Parametric estimation: The parametric estimate for the multiclass case is computed as $\tilde{p}_{\mathfrak{T}}(\boldsymbol{x}) = \sum_{l=1}^{L} P_l \cdot p(\boldsymbol{x}|\omega_l)$, where L is the number of classes. The values $P_l = N_l/N$, with N_l being the number of training vectors in class ω_l, are the a priori probabilities of the classes, and the class-conditional likelihoods $p(\boldsymbol{x}|\omega_l)$ are multivariate Gaussians. We use the maximum-likelihood estimate $\mu_l = \frac{1}{N_l} \sum_{k=1}^{N_l} \boldsymbol{x}_k^l$ and the unbiased estimate $\boldsymbol{\Sigma_l} = \frac{1}{N_l - 1} \sum_{k=1}^{N_l} (\boldsymbol{x}_k^l - \boldsymbol{\mu}_l)(\boldsymbol{x}_k^l - \boldsymbol{\mu}_l)^T$ for the class means and covariance matrices.

B. Elastic transform

The nonlinear Gaussianization transform is computed in a variational approach from the elastic transform [11] that modifies the nonparametric pdf estimate such that it becomes as similar as possible to the Gaussian parametric one (see Fig. 1).

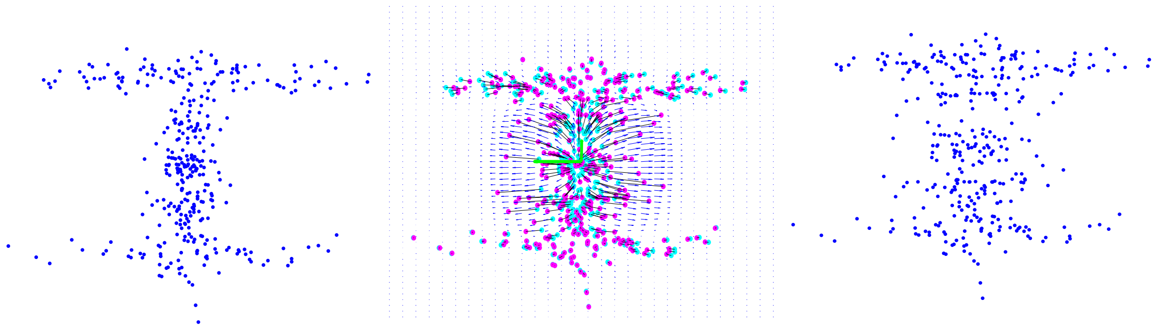

Fig. 2. Shown here are first the original non-Gaussian data, then the transform grid defined over the target region, with initial positions of the data points (cyan) and final position after applying the transform (magenta). The displacement vectors are also marked (black) as well as the principal components (light green). Finally the transformed data is shown.

Next we denote the nonparametric estimate $p_{\mathfrak{O}}(\boldsymbol{x})$ with $\mathfrak{O}(\boldsymbol{x})$ (for original) and parametric pdf estimate $p_{\mathfrak{T}}(\boldsymbol{x})$ with $\mathfrak{T}(\boldsymbol{x})$ (for target). The sought transformation is $\phi : \mathbb{R}^d \to \mathbb{R}^d$, with d the dimension of the input. The transformed original $\mathfrak{O}(\phi(\boldsymbol{x}))$ is as similar as possible to the target $\mathfrak{T}(\boldsymbol{x})$. The transformation $\phi = \boldsymbol{x} - u(\boldsymbol{x})$ has two parts: the identity $\boldsymbol{x}$ and the displacement $u(\boldsymbol{x})$.

Given $\mathfrak{T}$ and $\mathfrak{O}$ we look for the displacement u such that

$$\mathcal{I}[u] = \mathcal{D}[\mathfrak{T}, \mathfrak{O}, u] + \alpha \mathcal{S}[u] \to \min . \tag{1}$$

where $\mathcal{D}[\mathfrak{T}, \mathfrak{O}; u]$ is the distance between $\mathfrak{T}$ and $\mathfrak{O}$ with respect to u, $\mathcal{S}[u]$ is a regularizing term and α is a positive real constant. The distance measure we use here is the sum of squared differences $\mathcal{D}[\mathfrak{T}, \mathfrak{O}; u] = \frac{1}{2}\|\mathfrak{O}(\phi(\boldsymbol{x})) - \mathfrak{T}\|^2_{L_2(\Omega)}$, with Ω being the region under consideration, as described in the beginning of this section. As regularizing term we use the linearized elastic potential $\mathcal{S}[u] = \int_\Omega \frac{\mu}{4} \sum_{j,k=1}^{d} (\partial_{x_j} u_k + \partial_{x_k} u_j)^2 + \frac{\lambda}{2}(\text{div } u)^2 d\boldsymbol{x}$ with λ and μ being two constants [11].

The solution of the optimization problem (1) is obtained by numerically solving the corresponding Euler-Lagrange equations $f = \mu \triangle u + (\lambda + \mu)\nabla \text{div } u$, with f the force related to the distance measure $\mathcal{D}$. For this purpose a fixed-point iteration scheme is used. The displacement u is made time dependent and the sought minimizer is obtained as the steady-state solution of the corresponding time-dependent partial-differential equation (see [11] for a more detailed analysis). The transform thus obtained is diffeomorphic.

C. *Transform grid*

To ensure numerical tractability, the elastic transform is computed over a discretization of the target region Ω. We call this discretization the *transform grid* (see Fig. 2). The displacement u is computed at noninteger positions with the help of interpolation (we have used bilinear interpolation in our experiments).

The size δ of the grid is an important parameter of our transform. It is related to the total variance of the training data. Empirically we found the following relationship $\delta = \log_{10}(\mathrm{tr}(\Sigma))$, where Σ is the covariance matrix of the training data.

III. Experiments and discussion

We test our multiclass Gaussianization on synthetic and real data. The tests on synthetic data are meant to demonstrate exemplary the benefits and limitations of our approach. These tests are conducted on separable data, differentiating between linearly separable data and nonlinearly separable data. We have chosen the separable case such as to have a reference in the sense that error-free classification is possible there. The real data is the Fischer's iris dataset [1] and is not separable.

During testing we have randomly divided our data into a training and a test set, each containing 50% of the initial data points. Using the training set we compute the Gaussianization transform, then apply it to the test set.

We work with binary classifiers. We have classified the data both before (Org.) and after Gaussianization (Gauss.) with five types of classifiers: (i) a white Gaussian Bayesian classifier computed under the assumption of equal, unit class covariance matrices (W), (ii) a linear Gaussian Bayesian classifier computed under the assumption of equal class covariance matrices (L), (iii) a nonlinear Gaussian Bayesian classifier (nL), (iv) a SVM with a Radial Basis Function (rbf) kernel (SVM) and (v) a linear perceptron (P).

Some parameters of the elastic transform are set a-priori. All our experiments were satisfactorily conducted with the same parameter choice ($\alpha = 1$, $\lambda = 4 \cdot 10^{-5}$ and $\mu = 16 \cdot 10^{-5}$), established by six-fold cross validation on the linearly separable 2D data set. Other parameters (δ and h), are computed from the training data.

A. *Experiments on synthetic data*

We have conducted tests on data sets with dimensions of up to 10 and with various numbers of classes between two and five. Here we summarize our findings on 2D examples, a dimension that is most convenient for visualization. There are two classes present. For each experiment we have used 800 manually generated data points, 400 per class.

1) Linearly separable data: The linearly-separable data and the displacement field of the Gaussianization transform are shown in Fig. 3. The result of Gaussianization on the test set is shown in Fig. 4. The elastic transform parameters were $\delta = 2.1$ and $h = 1.6$. The classification results are shown in Table I. Only after Gaussianization does the linear Gaussian classifier find the separating surface.

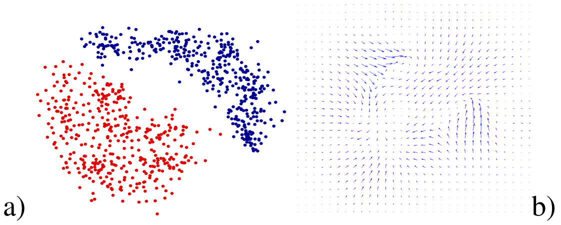

Fig. 3. Linearly-separable data (a) and corresponding displacement field (b). The class affiliation is color coded.

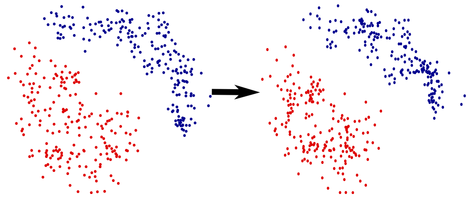

Fig. 4. Linearly-separable test dataset before and after Gaussianization.

2) Nonlinearly separable data: We have also generated nonlinearly separable data, shown in Fig. 5. The elastic transform parameters were $\delta = 1.8$ and $h = 1.4$. The classification results for this case are shown in Table II. The quadratic and linear Gaussian classifiers provide identical results, because the class-covariance matrices are very similar, as it can be seen in Fig. 6.

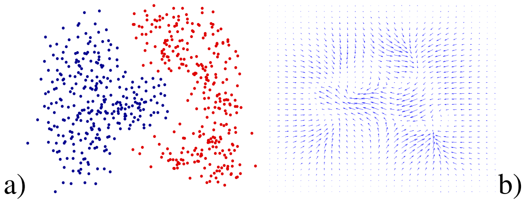

Fig. 5. Nonlinearly separable data (a) and corresponding displacement field (b). The class affiliation is color coded.

TABLE I

RESULTS ON THE LINEARLY SEPARABLE DATASET.

error (%)	W	L	nL	SVM (SVs)	P
Org.	1.75	0.75	0.75	0 (15)	0.25
Gauss.	0.25	0	0	0 (11)	0

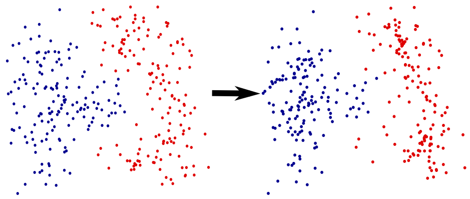

Fig. 6. Nonlinearly separable test dataset before and after Gaussianization.

TABLE II
RESULTS ON THE NONLINEARLY SEPARABLE DATASET.

error (%)	W	L	nL	SVM (SVs)	P
Org.	5.25	4.25	4.25	1.25 (17)	3.75
Gauss.	2.25	1.75	1.75	0.25 (12)	1.75

B. *Experiments on real data*

We have also tested our algorithm on the Iris dataset. It contains 150 feature vectors from three classes, with 50 vectors each. There are four features per vector. To adapt the binary classifiers to a multiclass scenario, we have used a majority voting rule. The results are shown in Table III. The elastic transform parameters were $\delta = 0.7$ and $h = 0.6$.

C. *Discussion*

The experiments on synthetic data show that after Gaussianization, the tested classifiers work better, in the sense that they make less errors. Also in comparison to other classifiers, the improvement of Gaussian classifiers—in terms of reduction of the error rate—is larger. For rbf-SVMs the number of support vectors diminishes. In the case of real data, the perceptron and the white Gaussian classifier missclassify three, respectively two more vectors after Gaussianization, but the nonlinear Gaussian classifier perfectly separates the data. Thus, the Gaussianization transform brings improvements only for the more powerful classifiers. Our method has five parameters. Two of them can be computed from the training data, the rest

TABLE III
RESULTS ON THE IRIS DATASET.

error (%)	W	L	nL	SVM (SVs)	P
Org.	8	2.66	2.67	1.33 (20)	10.66
Gauss.	10.66	1.33	0	0 (17)	14.66

should be established by cross-validation. The experiments show that the method is largely insensitive to the latter.

IV. Summary, conclusions and outlook

We proposed a nonlinear multiclass Gaussianization transform for the purpose of supporting the Gaussian assumption often made in many classification problems. The multiclass Gaussianization is computed as the displacement field of an elastic transform that makes the nonparametric pdf estimate of the training data as similar as possible, with respect to the ssd between the two functions, to a GMM with one component per class. The Gaussianization transform is defined at discrete positions over a region of interest centered on the training sample. Data points outside this region are left unmodified. We restrict our analysis to the support of the available training set. We use interpolation to compute the displacement at positions between the knots of the grid where the transform is defined.

δ and h are computed from the training data. The size δ of the grid is a parameter to which our method is particularly sensitive. The formula we have proposed for this parameter was established empirically, therefore better choices are possible. Theoretically we could impose new means and variances for the transformed data with the aim to increase separability, but practically this destroys the cohesion of the data with negative influences on the classification performance. At the same time, our transform is diffeomorphic and the elastic constraint we use supports data cohesion in the transformed space. Hence, we will hardly achieve perfect Gaussianization, but rather increase the gaussianity in the training sample. Our Gaussianization method is susceptible to the curse of dimensionality, as it relies on pdf estimation. It is also computationally demanding having to determine the displacement field at each grid point. We are currently investigating methods to overcome such limitations. There are optimization schemes—developed in the context of image registration—that decrease the computation time [11]. Nevertheless, we believe that there is a practical upper limit for the size of the displacement field allowing about 15 dimensions for the feature space and ten grid points per dimension. Larger displacement fields lead, with the current hardware, to unacceptably long computation time.

References

[1] J.C. Bezdek, J.M. Keller, R. Krishnapuram, L.I. Kuncheva, and N.R. Pal. Will the real iris data please stand up? *IEEE Trans. on Fuzzy Systems*, 7(3):368–369, 1999.

[2] S. Chen and R. Gopinath. Gaussianization. In *Proc. of NIPS*, Denver, USA, 2000.

[3] T. M. Dias, R. Attux, J. M. Romano, and R. Suyama. Blind source separation of post-nonlinear mixtures using evolutionary computation and gaussianization. In *Proc. of ICA*, pages 235–242. Springer, 2009.

[4] J. H. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *J. Amer. Statistical Assoc.*, 79:599–608, 1984.

[5] R. A. Gopinath. Maximum likelihood modeling with gaussian distributions for classification. In *Proc. of ICASSP*, pages 661–664, Seattle, U.S.A., 1998.

[6] R. Hogg, A. Craig, and J. McKean. *Introduction to mathematical statistics.* Prentice Hall, 6 edition, 2004.

[7] P. Kidmose. Adaptive filtering for non-gaussian processes. *Proc. of ICASSP*, pages 424–427, 2000.

[8] V. Laparra, G. Camps-Valls, and J. Malo. Iterative gaussianization: From ICA to random rotations. *IEEE Trans. on Neur. Net.*, 22:534–549, 2011.

[9] S. Lyu and E. P. Simoncelli. Nonlinear extraction of independent components of natural images using radial gaussianization. *Neur. Comput.*, 21:1485–1519, June 2009.

[10] Imen Mezghani-Marrakchi, G. Mah, M. Jadane-Sadane, S. Djaziri-Larbi, and M. Turki-Hadj-Allouane. ”gaussianization” method for identification of memoryless nonlinear audio systems. In *Proc. of EUSIPCO*, pages 2316 – 2320, 2007.

[11] Jan Modersitzki. *Numerical methods for image registration*. Oxford university press, 2004.

[12] G. Saon, S. Dharanipragada, and D. Povey. Feature space gaussianization. In *Proc. of ICASSP*, pages I – 329–332, 2004.

[13] M.P. Wand and M.C. Jones. *Kernel Smoothing.* Chapman and Hall, 1995.