A NOVEL KLT ALGORITHM OPTIMIZED FOR SMALL SIGNAL SETS

Francesco Gianfelici, Giorgio Biagetti, Paolo Crippa, and Claudio Turchetti

Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni,
Università Politecnica delle Marche,
Via Brecce Bianche 12, I-60131 Ancona, Italy.

ABSTRACT

Karhunen-Loève Transform, being able to represent stochastic processes under appropriate conditions, is a powerful signal processing tool. But the high computational cost incurred in the modeling of long signals has limited its use in the recognition of speech segmented at the word level.

In this paper we present a novel algorithm that significantly reduces the computational cost when the number of signals to be treated is small in comparison to their samples.

1. INTRODUCTION

The Karhunen-Loève Transform (KLT) is a representation of stochastic processes (SPs) defined over a bounded interval, and is therefore suitable to characterize speech signals segmented at the word level. This transformation ensures the best approximations for SPs under the fixed-rank condition, thus establishing a second-order statistical characterization of the SP in terms of uncorrelated random variables (RVs). In general, this approach produces a canonical representation of a given zero-mean SP that can be written as a linear combination of orthonormal functions, whose coefficients are thus zero-mean RVs.

The KLT properties described so far are the roots for its widespread use amongst different application fields, such as information coding, image compression and restoration, model order reduction in distributed parameter systems, and so on. In the field of speech signal processing, such operator is generally used as an effective means for speech enhancement, but apart from this, it only plays a marginal role in the modeling or recognition of speech, mostly because of its computational complexity. Some approximated approaches to overcome this problem have been developed [1], but the need for establishing a rigorous formulation of the signal representation, without relying on approximations, may discourage the use of such techniques in many applications.

In this paper we propose an efficient algorithm to compute the KLT, suitable for the development of a rigorous model of speech signals. Different utterances of the same word have been considered as realizations of an SP to which it is possible to apply the KLT operator. Using a theoretical formulation, based on an analytical approach, a procedure to compute the KLT operator has been obtained. The proposed formulation exploits the (integral operator) kernel separability property that arises when the correlation of the SP is estimated from a set of known sample realizations. The resulting algorithm in the discrete-time domain can be viewed as a sort of Singular Value Decomposition (SVD) [2] optimized to address the problem of computing the KLT in the common case when the number of realizations is small with respect to their length (expressed in samples). Experimental evidence showed a significant amount of reduction in computation time with respect to both SVD-based and traditional KLT algorithms.

2. TRADITIONAL KLT

Let $\xi(t)$ be a real-valued zero-mean SP, defined over the interval $[0,\, T]$, whose autocorrelation function is $R_{\xi\xi}(t,s) = \langle \xi(t)\, \xi(s) \rangle$. The KLT [3] is a canonical representation of such SP, in the mean square sense, given by:

$$\xi(t) = \sum\nolimits_{k=1}^{\infty} a_k\, \varphi_k(t) \tag{1}$$

where $\{\varphi_k(t)\}_{k\in\mathbb{N}}$ is a set of orthonormal eigenfunctions of the autocorrelation function $R_{\xi\xi}$ and a_k are uncorrelated zero-mean RVs, defined as:

$$a_k = \int_0^T \xi(t)\, \varphi_k(t)\, dt \quad , \quad \langle a_i\, a_j \rangle = \lambda_i\, \delta_{ij}\ , \tag{2}$$

and λ_k happens to be the eigenvalue corresponding to the eigenfunction $\varphi_k(t)$ of the operatorial equation:

$$\int_0^T R_{\xi\xi}(t,s)\, \varphi_k(s)\, ds = \lambda_k\, \varphi_k(t) \tag{3}$$

whose solutions are such that:

$$R_{\xi\xi}(t,s) = \sum\nolimits_{n=1}^{\infty} \lambda_n\, \varphi_n(t)\, \varphi_n(s)\ . \tag{4}$$

From (2) one of the most important properties of this representation follows: If the series (1) is truncated in such a

way as to maintain only the terms corresponding to the bigger eigenvalues, the resulting sum will be the best approximation, in the mean square sense, of the stochastic process being represented. This property may be easily verified using the fact the the coefficients a_k are uncorrelated.

In the case of finite discrete-time SPs each realization can be seen as a vector of random variables. Let $\mathbf{x}$ be a vector of this kind, whose j-th element corresponds to the random variable extracted from the process $\xi(t_j)$ in correspondence of the j-th point where the process is defined. It is thus possible to replace the correlation function $R_{\xi\xi}(t,s)$ with a matrix:

$$\mathbf{R_{xx}} = \langle \mathbf{x}\,\mathbf{x}^T \rangle \tag{5}$$

which is obviously symmetric, and the eigenfunctions simply with the eigenvectors of the (5).

Let $\mathbf{U}$ be a matrix whose columns constitute a set of orthonormal eigenvectors of $\mathbf{R_{xx}}$, so that $\mathbf{U}^T\,\mathbf{U} = \mathbf{I}$ and:

$$\mathbf{R_{xx}} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T \quad , \qquad \mathbf{\Lambda} = \operatorname*{diag}_{i=1,\ldots,M} \lambda_i \tag{6}$$

where M is the number of non-null eigenvalues. Thus, for every realization $\mathbf{x}$, it is possible to write the vector $\mathbf{a}$ of the coefficients a_k as $\mathbf{U}^T\,\mathbf{x}$ and *vice-versa*, through the matrix $\mathbf{U}$, we can obtain the realizations from the vector of the coefficients as:

$$\mathbf{x} = \mathbf{U}\,\mathbf{a}\,. \tag{7}$$

Eqn. (7) is the discrete-time equivalent of the series representation (1). These representations are called Karhunen-Loève transforms, and they constitute a generalization of the Fourier transform.

In order to exactly compute $\mathbf{U}$, the correlation matrix of the stochastic process to be represented should be known in advance. This is often not possible, and it is common practice to rely on approximations computed using an estimate:

$$\mathbf{R_{xx}} \approx (1/N) \sum\nolimits_{k=1}^{N} \mathbf{x}_k\,\mathbf{x}_k^T = (1/N)\,\mathbf{D}\,\mathbf{D}^T \tag{8}$$

made with a set of N sample realizations $\mathbf{x}_k$, contained in the columns of the $L \times N$ matrix $\mathbf{D}$.

The matrix so obtained may be used straightforwardly to compute the decomposition (6), but it is necessary to be aware that observations of the stochastic process must be made so that the sample realizations are all of the same length L, and that this length determines the dimensions of the resulting matrix $\mathbf{R}$, of which it is necessary to compute eigenvalues and eigenvectors. The algorithms commonly employed to solve this symmetric eigenproblem have indeed a computational complexity of order $O(L^3)$ (see Section 4), but moreover they require the whole $L \times L$ matrix to be stored in memory. Alternatively, it may be possible to directly compute the SVD of the data matrix $\mathbf{D}$, but this approach is unsuited to continuous-time signals and proved to be from almost twice to three times slower than the proposed approach.

3. THE FAST KLT ALGORITHM

Considering the continuous-time problem first, we maintain that when the correlation function can be approximated as:

$$R_{\xi\xi}(t,s) \approx \frac{1}{N} \sum\nolimits_{i=1}^{N} x_i(t)\,x_i(s) \tag{9}$$

where $x_i(t)$, $i = 1,\ldots,N$ are sample realizations of the stochastic process $\xi(t)$, it is possible to exploit the separability property of the kernel to get a closed-form solution to the eigenproblem, and to obtain a substantial gain in the discrete-time computation of non-null eigenvalues and corresponding eigenvectors. Rewriting (3) as follows:

$$\frac{1}{N} \int_0^T \sum\nolimits_{i=1}^{N} x_i(t)\,x_i(s)\,\phi_k(s)\,ds = \lambda_k \phi_k(t) \tag{10}$$

and letting:

$$a_{ik} = \int_0^T x_i(s)\,\phi_k(s)\,ds \tag{11}$$

we have:

$$\frac{1}{N} \sum\nolimits_{i=1}^{N} a_{ik}\,x_i(t) = \lambda_k \phi_k(t)\,. \tag{12}$$

Multiplying both sides of (11) by the eigenvalue λ_k and applying (12), we obtain:

$$\lambda_k\,a_{ik} = \frac{1}{N} \sum\nolimits_{j=1}^{N} a_{jk} \int_0^T x_i(s)\,x_j(s)\,ds \tag{13}$$

which is the eigenproblem about the matrix $\mathbf{S}$ of the normalized scalar products s_{ij} between pairs of realizations:

$$s_{ij} = \frac{1}{N} \int_0^T x_i(s)\,x_j(s)\,ds\,. \tag{14}$$

Let the diagonal matrix $\mathbf{\Lambda}$ of the non-null eigenvalues λ_k and the $N \times M$ eigenvector matrix $\mathbf{A}$ be the solution to this problem, so that $[\mathbf{A}]_{ik} = a_{ik}$ and:

$$\mathbf{S} = \mathbf{A}\,\mathbf{\Lambda}\,\mathbf{A}^T \qquad \mathbf{A}^T\,\mathbf{A} = \mathbf{I}\,. \tag{15}$$

Having found the a_{ik} coefficients it is easy to compute all the eigenfunctions corresponding to non-null eigenvalues using (12). After normalization the eigenfunctions are thus:

$$\varphi_k(t) = \frac{1}{\sqrt{\lambda_k\,N}} \sum\nolimits_{i=1}^{N} a_{ik}\,x_i(t)\,. \tag{16}$$

Similarly, the discrete-time equivalent of (16) will be:

$$\mathbf{U} = N^{-1/2}\,\mathbf{D}\,\mathbf{A}\,\mathbf{\Lambda}^{-1/2} \tag{17}$$

with the columns of $\mathbf{U}$ containinig the eigenvectors of the corresponding discrete-time eigenproblem, problem that can simply be stated by redefining the matrix $\mathbf{S}$ as being:

$$\mathbf{S} = (1/N)\,\mathbf{D}^T\,\mathbf{D}\,. \tag{18}$$

Eqs. (18), (15), and (17) thus represent, in this order, the operational steps the Fast KLT algorithm is composed of.

"Traditional KLT" complexity		
i)	Computation of $\mathbf{R}$	L^2N
ii)	Computation of $\mathbf{U}$ (eigenvectors)	$\frac{4}{3}L^3$
"Fast KLT" complexity		
i)	Computation of $\mathbf{S}$	LN^2
ii)	Computation of $\mathbf{A}$ and $\mathbf{\Lambda}$	$\frac{4}{3}N^3$
iii)	Computation of $\mathbf{U}$ (eigenvectors)	$(L+1)NM$

Table 1. Fast & Traditional KLT complexity

4. FAST VS. TRADITIONAL KLT

This section will give a detailed analysis of the computational savings that can be achieved using the Fast KLT algorithm, instead of the traditional one.

As depicted in Table 1, the traditional KLT algorithm is composed essentially of two steps: the computation of the matrix, and the solution of the corresponding eigenproblem. The Fast KLT algorithm performs the same operations, but on a smaller matrix, and adds a third step needed to compute the linear combination (17). In this context, the computational cost has been expressed in terms of the (approximate) number of operations, in this case simple multiplications (MULs), required to perform each step.

As it is easy to note, if $N \ll L$ the efficiency of the fast algorithm is far higher than that of the traditional algorithm: the computation of $\mathbf{S}$ only requires LN^2 MULs instead of the L^2N required for $\mathbf{R}$. The eigenproblem to be solved is only of order N instead of order L, accounting for [4] about $4/3\,N^3$ MULs, instead of $4/3\,L^3$. The Fast version has indeed an additional step, needed to compute (17), but its complexity is just similar to that of the first step. It accounts for about $(L+1)NM$ MULs, and being M (number of non-null eigenvalues) not greater than, and usually very close to, N, this cost can be approximated to LN^2, which is the same cost that has been incurred in the computation of the matrix $\mathbf{S}$ itself.

From this data, it is possible to state that, in the condition we consider, that is, when $L \gg N$, the overall cost of the two algorithms may be appoximated with $4/3\,L^3$ for the traditional version and with $2LN^2$ for the Fast version.

An experimental measure of the computation times required to obtain the eigenfunctions has been carried out on a system equipped with a 3.2 GHz Pentium™ IV processor, using the `dsyevr` and `dgesdd` routines [5] of the LAPACK libraries, for eigen- and singular value decomposition, respectively, combined with the ATLAS [6] package for basic linear algebra calculations. Some results have been reported in Table 2, from which a two order of magnitude gain with respect to the tradition KLT and a factor between two and three with respect to SVD appears. Furthermore, the traditional algorithm uses larger matrices, and was unable to execute on the test system for longer L.

L	Step *i)*	Step *ii)*	Step *iii)*	Total
"Fast KLT" $N = 500$				
500	0.06 s	0.37 s	0.07 s	0.50 s
5000	0.54 s	0.37 s	0.64 s	1.55 s
20000	5.02 s	0.37 s	2.58 s	7.97 s
"SVD-based KLT" $N = 500$				
500	—	—	0.83 s	0.83 s
5000	—	—	4.31 s	4.31 s
20000	—	—	17.16 s	17.16 s
"Traditional KLT" $N = 500$				
500	0.06 s	0.38 s	—	0.44 s
5000	6.04 s	154.08 s	—	160.12 s

Table 2. Computation time comparison of the fast versus the SVD-based and traditional KLT algorithm.

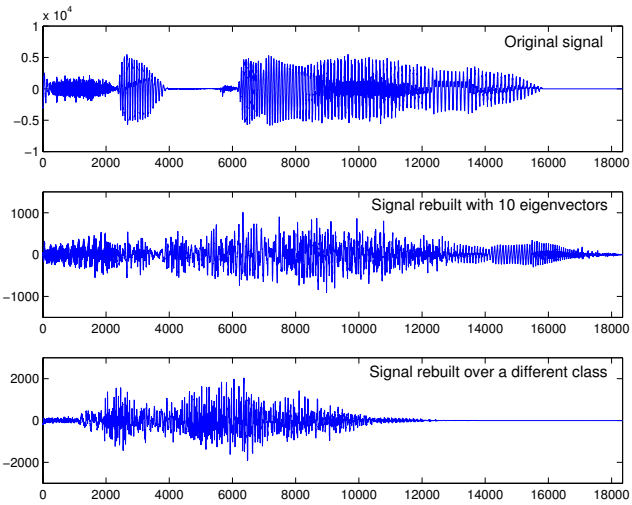

Fig. 1. A speech signal and its reconstructions.

5. APPLICATION EXAMPLES

In this section some examples of the application of the KLT to speech signals are presented. The knowledge base used to perform these applications was the Italian section of the Multext Prosodic Database [7], which is an extract from the EUROM.1 speech corpus. Ten Italian speakers, having different sex, age, and geographical origin, recorded 15 sentences in an anechoic room, amounting to nearly 7000 words, transcribed and segmented at the word level.

The computational complexity reduction obtained by using the Fast KLT algorithm permits to accurately model speech signals segmented at the word level. Thus, different utterances of the same word have been considered as different realizations of a single zero-mean SP, to which the Fast KLT operator has been applied. For each word it is thus possible to obtain the canonical decomposition (4). With this decomposition it is then possible to characterize a whole

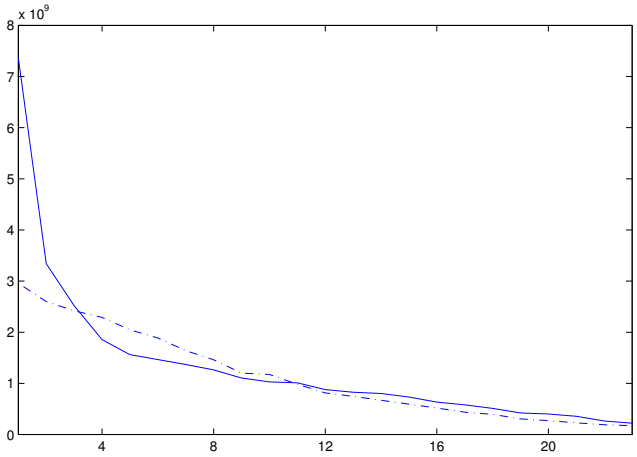

Fig. 2. Dynamic of the eigenvalues associated to the Italian words "settimana" and "favore".

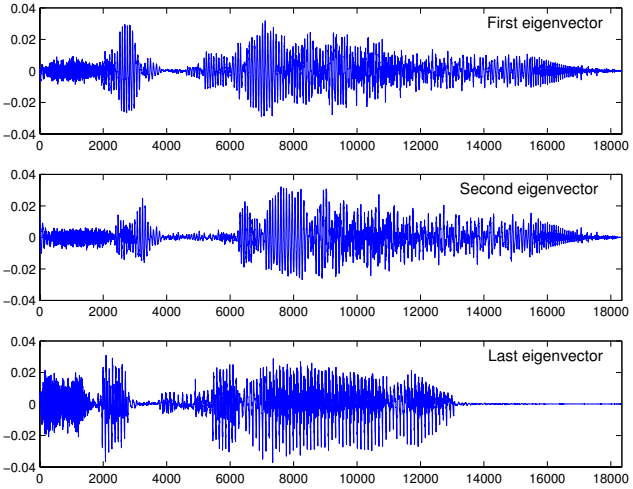

Fig. 3. Some eigenvectors of the Italian word "settimana".

class of signals through a set of eigenfunctions, which are dependent on the word spoken but not on the particular utterance, and are thus common to all the signals in the class. The particular utterance is further characterized by its projections onto the eigenfunction basis, i.e. by a realization of the RVs a_k in (1).

The signals to which the KLT is to be applied must all have the same length. To accomplish this, zero-padding has been applied at the end of each signal, and despite of the huge variability of signal lengths, no transformations nor resampling have been applied to the signals themselves, in order not to alter what we believe is a characteristic phenomenon of human speech.

As an example, the Italian word "settimana" (that means "week"), has been considered. One of its realizations is shown in Fig. 1, together with its reconstruction using a limited set of eigenvectors (the reconstruction over the whole set is obviously identical to the original signal), and over the eigenvectors obtained through the characterization of a different word (Italian "favore," which means "favor"). It is apparent that in this case the rebuilt word is totally different from the original one, as it is nevertheless a linear combination of words belonging to the class "favore." This phenomenon has some interesting aspects for its foreseeable applications in speech recognition techniques [8].

In order to complete the characterization, the dynamic of the eigenvalues is shown in Fig. 2, and some of the eigenvectors are shown in Fig. 3. It is clear that such eigenfunctions are a linear combination of the original signals, as it should be being them obtained through (16).

6. CONCLUSIONS

In this paper an efficient algorithm for the fast computation of the KLT operator, named Fast KLT, has been presented. It derives from a rigorous theoretical analysis based on properties of separable kernels. The Fast KLT efficiently computes eigenvectors corresponding to non-zero eigenvalues of the correlation matrix when this is estimated from a small number of sample realizations.

Some application examples of the Fast KLT have shown the effectiveness of this technique to model speech signals segmented at the word level, for which the application of the traditional algorithm would have been unpractical.

7. REFERENCES

[1] C.E. Davila, "Blind adaptive estimation of KLT basis vectors," *IEEE Trans. on Signal Processing*, vol. 49, no. 7, pp. 1364–1369, July 2001.

[2] S. Haykin, *Adaptive Filter Theory*, Prentice Hall Int. Ed., New Jersey, 1996.

[3] Edward R. Dougherty, *Random Processes for Image and Signal Processing*, SPIE—IEEE Series on Imaging Science and Engineering, 1998.

[4] Inderjit Singh Dhillon, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, Univ. of California, Berkeley, CA, 1997.

[5] E. Anderson *et al.*, *LAPACK Users' Guide*, SIAM, Philadelphia, PA, 3rd edition, 1999.

[6] R. Clint Whaley, Antoine Petitet, and Jack J. Dongarra, "Automated empirical optimization of software and the ATLAS project," *Parallel Computing*, vol. 27, no. 1–2, pp. 3–35, 2001.

[7] Estelle Campione and Jean Véronis, "A multilingual prosodic database," in *Proc. of the 5th Int. Conf. on Spoken Language Processing (ICSLP'98)*, Sydney, Australia, 1998, vol. 7, pp. 3163–3166.

[8] F. Gianfelici and C. Turchetti, *A Stochastic Process Recognizer*, Italian Patent Office - Italian Government Department of Productive Activities, 2004, Italian Patent (in Internationalization Phase), Dep. Num. AN2004A000050.