OpenBliSSART: design and evaluation of a research toolkit for blind source separation in audio recognition tasks

Felix Weninger, Alexander Lehmann, Björn Schuller

Angaben zur Veröffentlichung / Publication details:

Weninger, Felix, Alexander Lehmann, and Björn Schuller. 2011. “OpenBliSSART: design and evaluation of a research toolkit for blind source separation in audio recognition tasks.” In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 22-27 May 2011, Prague, Czech Republic*, edited by Petr Tichavský, Jan Černocký, Aleš Procházka, Jiří Jan, and Robert Vích, 1625–28. Piscataway, NJ: IEEE. https://doi.org/10.1109/ICASSP.2011.5946809.

OPENBLISSART: DESIGN AND EVALUATION OF A RESEARCH TOOLKIT FOR BLIND SOURCE SEPARATION IN AUDIO RECOGNITION TASKS

Felix Weninger[1], *Alexander Lehmann*[2], *and Björn Schuller*[1]

[1] Institute for Human-Machine Communication, Technische Universität München, Germany
{lastname}@tum.de
[2] Group of Applied Informatics – Cooperative Systems, Technische Universität München, Germany
lehmanna@in.tum.de

ABSTRACT

We describe and evaluate our toolkit openBliSSART (open-source Blind Source Separation for Audio Recognition Tasks), which is the C++ framework and toolbox that we have successfully used in a multiplicity of research on blind audio source separation and feature extraction. To our knowledge, it provides the first open-source implementation of a widely applicable algorithmic framework based on non-negative matrix factorization (NMF), including several preprocessing, factorization, and signal reconstruction algorithms for monaural signals. Apart from blind source separation using supervised and unsupervised NMF, we show how the framework is useful for the increasingly popular audio feature extraction methods by NMF. Furthermore, we point out a numerical optimization for NMF, and show that NMF source separation in real-time on a desktop PC is feasible with our implementation. We conclude with an evaluation of our toolkit on supervised speaker separation, demonstrating how our algorithmic framework allows to tune the real-time factors to the desired perceptual quality.

Index Terms— Blind Source Separation, Speech Enhancement, Instrument Separation, Real-Time Signal Processing

1. BACKGROUND AND OBJECTIVES

Blind Source Separation (BSS) in audio signals is a challenging field with a broad range of applications, particularly in Music Information Retrieval (MIR) and automatic speech recognition (ASR). In MIR, it can be used for polyphonic transcription (‘WAV to MIDI converter’) or recognition of lyrics in singing, and typical sources to be separated include instruments (e. g., drums), vocals, or single notes (for the transcription case). On the other hand, BSS techniques can deliver enhanced robustness of ASR by separating the wanted speech from interfering signals such as background noise, or even other speakers. The last years have seen a growing number of approaches exploiting Non-Negative Matrix Factorization (NMF) [1–5], whose most prominent advantage is that it can extract an arbitrary number of sources from monophonic signals.

In summary, NMF has delivered excellent results in blind source separation both of speech and music signals; in particular, the last years have seen considerable improvements in perceptual quality of the results [2, 3, 5]. Furthermore, applications of NMF in audio processing are not limited to BSS, as there is a growing number of studies showing the advantage of NMF-based audio feature extraction, especially in noisy conditions [6–8]. On the other hand, with the increasing amount of computational power available today even on mobile devices, we are moving towards the point where NMF-based algorithms are ready to be used in real-life applications. However, the lion’s share of the studies in the field focuses on optimizing the separation results, neglecting implementation issues. Besides, to our knowledge few demonstrators or open-source implementations for NMF-based source separation exist[1]. The openBliSSART toolkit, however, integrates leading-edge NMF algorithms into a flexible, real-time capable, and open-source[2] C++ framework that is useful for both music and speech processing and can be seamlessly integrated into other toolkits.

In the remainder of this paper, we will first describe the algorithmic framework implemented in openBliSSART in Sec. 2, pointing out a potential numerical optimization, and outline our usage of existing open-source software to improve stability and performance. We evaluate openBliSSART, and in particular the proposed NMF optimization, in Section 3, and conclude with an outlook on future research questions in Section 4. To increase clarity of the following section, we introduce the following notations: for a matrix $\mathbf{A}$, the notation $\mathbf{A}_{i,:}$ – resembling Matlab syntax – denotes the i-th row of $\mathbf{A}$ (as a row vector), and we analogously define $\mathbf{A}_{:,j}$ for the j-th column of $\mathbf{A}$ (as a column vector). We write $\mathbf{A} \otimes \mathbf{B}$ for the elementwise product of matrices $\mathbf{A}$ and $\mathbf{B}$; division of matrices is always to be understood as elementwise.

2. ALGORITHMIC FRAMEWORK

2.1. Component Separation Algorithms

openBliSSART’s design is oriented on BSS techniques realized by NMF. A basic procedure is to extract an arbitrary number of sources (*components*) from audio files by computing the non-negative factorization of a spectrogram matrix $\mathbf{V} \in \mathbb{R}_+^{M \times N}$ obtained from short-time Fourier transformation (STFT) into a spectral basis $\mathbf{W} \in \mathbb{R}_+^{M \times R}$ and activation matrix $\mathbf{H} \in \mathbb{R}_+^{R \times N}$:

$$\mathbf{V} = \mathbf{W}\mathbf{H}, \qquad (1)$$

yielding R component spectrograms $\mathbf{V}^{(j)}, j = 1, \ldots, R$ either by multiplication of each basis vector $\mathbf{w}^{(j)} := \mathbf{W}_{:,j}$ with its activations $\mathbf{h}^{(j)} := \mathbf{H}_{j,:}$, as in [2], or by a more advanced ‘Wiener filter’

This work was supported by the Federal Republic of Germany through the German Research Foundation (DFG) under the grant no. SCHU 2508/2-1 (“Non-Negative Matrix Factorization for Robust Feature Extraction in Speech Processing”).

[1] For example, on http://www.durrieu.ch/phd/software.html, a NMF algorithm for leading voice separation is available.

[2] Source code can be found at http://openblissart.github.com/openBliSSART.

approach, as described in [1, 5]:

$$\mathbf{V}^{(j)} = \mathbf{V} \otimes \frac{\mathbf{w}^{(j)}\mathbf{h}^{(j)}}{\mathbf{W}\mathbf{H}}. \tag{2}$$

Each $\mathbf{V}^{(j)}$ is then transformed back to the time domain by inverse STFT.

To obtain a factorization according to (1), a variety of NMF algorithms can be used that minimize a distance function $d(\mathbf{V}|\mathbf{WH})$ by multiplicative updates of the matrices, starting from a random initialization. $d(\mathbf{V}|\mathbf{WH})$ can be chosen as the β-divergence or one of its special instances, the Itakura-Saito (IS) [5] divergence, Kullback-Leibler (KL) divergence, or squared Euclidean distance (ED) [9]. Besides, to support overcomplete decomposition, i. e., choosing R such that $R(M + N) > MN$, sparse NMF variants [2] for either of the aforementioned distance functions, as well as the sparse Euclidean NMF variant used in [10], are implemented. Additionally, non-negative matrix deconvolution (NMD) [1, 3] is provided as a context-sensitive NMF extension where each component is characterized by a sequence of spectra, rather than by an instantaneous observation.

The aforementioned NMF and NMD algorithms can be run on magnitude, power, and Mel-scale spectra; besides, the 'sliding window' NMF from [6] is implemented that transforms the original spectrogram $\mathbf{V}$ to a matrix $\mathbf{V}'$ such that every column of $\mathbf{V}'$ is the row-wise concatenation of a sequence of short-time spectra (in the form of row vectors):

$$\mathbf{V}' := \begin{bmatrix} \mathbf{V}_{:,1} & \mathbf{V}_{:,2} & \cdots & \mathbf{V}_{:,N-T+1} \\ \vdots & \vdots & \ldots & \vdots \\ \mathbf{V}_{:,T} & \mathbf{V}_{:,T+1} & \cdots & \mathbf{V}_{:,N} \end{bmatrix}, \tag{3}$$

where T is the desired context length. That is, the columns of $\mathbf{V}'$ correspond to overlapping sequences of spectra in $\mathbf{V}$. Note that we implemented inverse operations to the aforementioned transformations of the spectrogram, including Mel filtering and transformation according to (3), to allow proper signal reconstruction.

Finally, and for reference purpose only, openBliSSART can also apply the FastICA algorithm [11] in the time domain.

2.2. Supervised Component Classification

To cope with scenarios such as instrument separation – as in [4] – it was necessary to extend the basic source separation capabilities: here, typically 20–40 NMF components are needed for appropriate signal modeling, thus the 'tracks' corresponding to one instrument, or an instrument class such as drums, generally comprise more than one component. Consequently, a classification process is necessary to overlay individual components into C class spectrograms, yielding the procedure depicted in Figure 1: first, a selection of training signals is separated by means of NMF. Subsequently, the resulting components are annotated (e. g., as drum or harmonic sounds), and features are extracted from them to train a classifier, e. g., a Support Vector Machine (SVM). Then, the actual separation process performs NMF and uses the previously trained classifier on the separated components to overlay them into class spectrograms $\mathbf{V}_c, c = 1, \ldots, C$: defining

$$J_c = \{j : (\mathbf{w}^{(j)}, \mathbf{h}^{(j)}) \text{ classified as class } c\},$$

$$\mathbf{V}_c = \sum_{j \in J_c} \mathbf{V}^{(j)}. \tag{4}$$

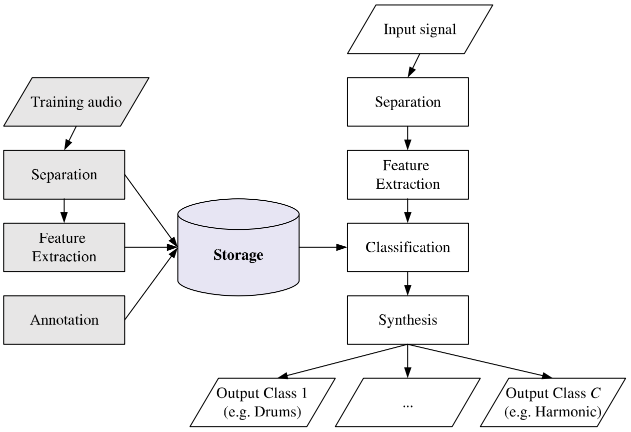


Fig. 1: Supervised component classification, as in instrument separation: the openBliSSART storage module manages the components from which a classifier is built for usage in the separation process. The steps required to train the classifier are depicted in gray shade.

preprocessing	algorithms	$d(\mathbf{V}\|\mathbf{WH})$	reconstruction
Mel filter	NMF	IS div.	default
Power spec.	NMD	KL div.	Wiener filtering
Sliding window		Eucl. dist.	comp. classif.
		+sparsity	

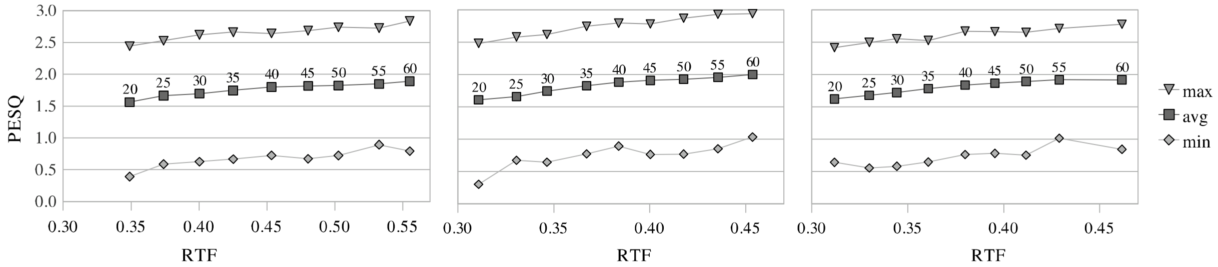
Table 1: Spectrogram preprocessing, factorization (according to a cost function $d(\mathbf{V}|\mathbf{WH})$), and signal reconstruction algorithms for monaural source separation implemented in openBliSSART.

Acoustic features which are used for classification include Mel-frequency Cepstral Coefficients or features specially suited to instrument separation. The available NMF algorithms are shown in Tab. 1, yielding a flexible framework for NMF-based source separation where preprocessing, factorization, and component reconstruction algorithms can be chosen independently.

2.3. Supervised NMF and Acoustic Feature Extraction

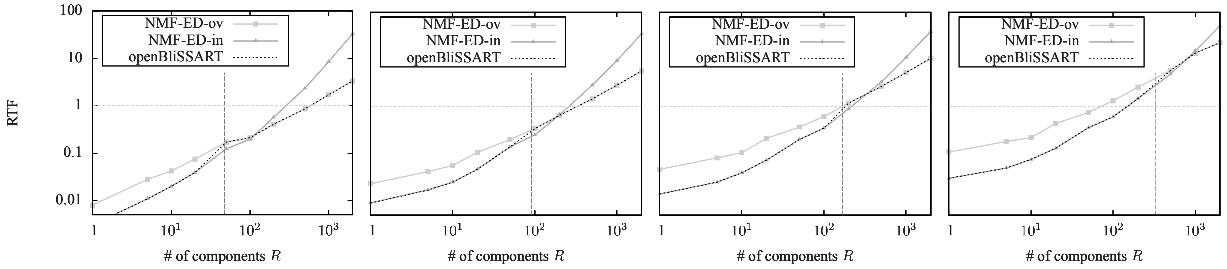
Apart from component classification, another method to integrate a-priori knowledge into the NMF source separation process is to perform *supervised NMF*, i. e., to predefine the first NMF factor as a set of spectra that are characteristic for the sources to be separated, as opposed to a random initialization of both factors, and to compute only the second NMF factor by multiplicative updates. A typical application is speech enhancement, where the sources comprise different persons speaking simultaneously [1, 10], or speech and noise [1]. The spectra used for initialization are themselves often estimated by NMF decomposition of training material [1, 7, 10]. Using any of the methods shown in Tab. 1, time signals corresponding, e. g., to different speakers can be synthesized by overlaying component spectrograms.

Finally, openBliSSART was the toolkit used in our research on supervised NMF feature extraction, which has delivered excellent results in robust speech processing including detection of non-linguistic vocalizations [7, 8]. Consequently, it receives increasing attention at the moment [6]. Thereby different sets of components are selected as an in- or overcomplete feature basis, for which the time-varying activation matrix (i. e., the second NMF factor) is computed by supervised NMF. Note that openBliSSART allows this matrix to be



(a) IS divergence (b) KL divergence (c) Euclidean distance

Fig. 2: Tuning of the trade-off between RTF and perceptual quality for supervised NMF speech separation by adjusting the number of Mel filters (20–60) and the NMF cost function. The average, minimum, and maximum PESQ score is shown for the separation of mixed signals spoken by pairs of male / female speakers (24 speakers total) from the TIMIT database.



(a) $M = 50$ (b) $M = 100$ (c) $M = 200$ (d) $M = 500$

Fig. 3: Euclidean NMF real-time factors (RTF) depending on the number of components (1 – 2 000), for matrices with $N = 1\,000$ columns and $M \in \{50, 100, 200, 500\}$ rows (Figs. 3a through 3d). 'NMF-ED-ov' and 'NMF-ED-in' denote the algorithms optimized for overcomplete and incomplete factorization, according to Sec. 2.4. 'openBliSSART' refers to the proposed automatic selection between 'NMF-ED-ov' and 'NMF-ED-in' based on the criterion $R(M + N) > MN$ to distinguish over- from incomplete factorization. The limit case $R(M + N) = MN$ is shown by the vertical bars, and real-time capability (RTF < 1) is indicated by the horizontal lines.

exported for further processing, e. g., by popular research toolkits such as Weka or HTK (Hidden Markov Model Toolkit), enabling the usage of NMF features for a huge variety of research. As a side note, hybrid supervised / unsupervised NMF such as in [7] is supported by openBliSSART, too, by allowing to specify which columns of the first NMF factor should be kept constant throughout the NMF iterations.

2.4. Optimization of Euclidean NMF

It is worth examining the multiplicative update rules for Euclidean distance NMF, as proposed in [9], more closely, to derive an interesting possibility for numerical optimization. In matrix formulation, they read

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{W}\mathbf{H}} \quad \text{and} \tag{5}$$

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \frac{\mathbf{V}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T}. \tag{6}$$

In these rules, we can rearrange the matrix products by using their associativity. First, we consider the denominator of the $\mathbf{H}$ update rule (Eq. 5), which contains the product $\mathbf{W}^T\mathbf{W}\mathbf{H}$. When executed in the order $\mathbf{W}^T(\mathbf{W}\mathbf{H})$, the computational complexity is $O(MNR)$; in contrast, it is $O(R^2(M + N))$ for the order $(\mathbf{W}^T\mathbf{W})\mathbf{H}$, assuming the standard matrix multiplication algorithm. Thus, the effort for the first case is expected to be lower if and only if $MN < R(M + N)$, that is, in case of overcomplete factorization. The nested matrix product $\mathbf{W}\mathbf{H}\mathbf{H}^T$ in the denominator of rule (6) can be treated analogously. Naturally, these are only asymptotic considerations, which are however supported by our experimental results using efficient linear algebra routines (Sec. 3). Hence, the openBliSSART routine for ED-NMF uses the computation order $(\mathbf{W}\mathbf{H})\mathbf{H}^T$ resp. $\mathbf{W}^T(\mathbf{W}\mathbf{H})$ for $MN < R(M + N)$, and $\mathbf{W}(\mathbf{H}\mathbf{H}^T)$ resp. $(\mathbf{W}^T\mathbf{W})\mathbf{H}$ otherwise. For evaluation of this strategy in the next section, the algorithms that always use the former or latter computation order will subsequently be denoted by 'NMF-ED-ov' resp. 'NMF-ED-in', as they are arguably optimized to either over- or incomplete factorization.

2.5. Optimization of FFT and linear algebra

For further performance enhancement, openBliSSART utilizes several leading-edge open-source libraries: first, FFTW [12] realizes Fast Fourier Transformation (FFT) for arbitrary window sizes, disposing of the need for zero padding. Furthermore, as all of the NMF algorithms integrated in openBliSSART perform a high number of matrix multiplications, we use the open-source BLAS implementation provided by the ATLAS project [13]. From our experience, the BLAS routines decrease the real-time-factor (RTF) by an order of magnitude for typical NMF applications, compared to a 'naive' matrix multiplication routine implemented in C++. Note that other efficient implementations of Fourier transformations and Basic Linear Algebra Subroutines (BLAS) could be effortlessly integrated, as C++

abstractions are provided in the signal processing and linear algebra libraries.

3. PERFORMANCE BENCHMARKS

To provide some performance benchmarks of our toolkit, we first visualize in Fig. 2 the trade-off between RTF and and separation quality in supervised speaker separation according to Sec. 2.3. Since in [1], no significant gain in perceptual quality could be obtained by using NMD instead of NMF bases, we restrict the evaluation to NMF. Thereby we compared the effect of using different numbers of Mel filters for scaling of the spectrogram: as the size of the matrix $\mathbf{V}$ increases linearly with the number of filters, so does the computational effort for factorization, while in contrast a gain in separation quality is expected by using more filters. Besides the number of Mel filters, we varied the cost function (IS, KL, and ED), as the algorithms minimizing these cost functions considerably differ in the number of required matrix operations. To our knowledge, the importance of both these parameters has not been addressed in previous studies.

We defined our evaluation methodology in accordance with [1]: we randomly selected 12 pairs of male and female speakers from the TIMIT database. For each pair, we mixed together two randomly selected sentences of roughly equal length, and computed a NMF basis $\mathbf{W}$ from the spectra in the other sentences spoken by each speaker. Using supervised NMF with $\mathbf{W}$, separated signals for both speakers were obtained, whose similarity to the originals was measured using the PESQ score (Perceptual Evaluation of Speech Quality, ITU-T recommendation P.862). 100 iterations were performed on a 2.4 GHz desktop PC with 4 GB of RAM, using a single computation thread. The RTFs are computed by taking the elapsed CPU time over the length of the mixed signals. From Fig. 2, it can be seen that best average results are obtained by taking the KL divergence as distance function. Notably, the Euclidean distance is on par with the KL divergence concerning the RTF, while the IS divergence takes considerably more computation time. As expected, the overall picture is that both computational effort and separation quality increase with the number of Mel filters. It must be taken into account, however, that the results were obtained on mixture signals of 2–3 seconds length, such that the overhead for initializing the separation application considerably influences the RTF.

Second, we show in Fig. 3 the RTFs for in- and overcomplete factorization by NMF minimizing the ED function. By comparing the RTFs for the ‘openBliSSART’ strategy to either of the algorithms optimized to in- or overcomplete factorization (‘NMF-ED-in’, ‘NMF-ED-ov’) according to Sec. 2.4, it can be seen that the proposed implementation that determines the optimal algorithm by the factorization dimensionality leads to considerable improvements in the RTF, especially for very low or very high numbers of components. For this experiment, we simply used random matrices, as separation quality is not affected by the choice of algorithm, and for RTF calculation we measured only the CPU time for the actual factorization and assumed that the matrix columns correspond to signal frames shifted by 10 ms.

4. CONCLUSION

We introduced openBliSSART, a toolkit for blind source separation and NMF-based audio processing, which has led us to great success in a variety of research, including instrument separation [4], noise-robust speech recognition [7], and detection of non-linguistic vocalizations in speech [8]. In contrast to previously available open-source implementations of specialized NMF algorithms, it provides a comprehensive, modular, and easily extensible framework; thus, we strongly believe that it will be of high value to the research community. On the other hand, it features some specific optimizations of NMF whose benefit could be demonstrated through our benchmark results.

In future development, besides keeping the toolkit on the leading edge by including source / filter models and probabilistic modeling, foremost we will enhance its applicability in real-life scenarios by allowing on-line and incremental audio processing, paving the way for interesting new fields of research on blind source separation.

5. REFERENCES

[1] P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 1–14, 2007.

[2] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1066–1074, March 2007.

[3] W. Wang, A. Cichocki, and J. A. Chambers, “A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2858–2864, July 2009.

[4] B. Schuller, A. Lehmann, F. Weninger, F. Eyben, and G. Rigoll, “Blind enhancement of the rhythmic and harmonic sections by NMF: Does it help?,” in *Proc. of the International Conference on Acoustics (NAG/DAGA 2009)*, Rotterdam, Netherlands, 2009, pp. 361–364, DEGA.

[5] C. Févotte, N. Bertin, and J.-L. Durrieu, “Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis,” *Neural Computation*, vol. 21, no. 3, pp. 793–830, March 2009.

[6] J. F. Gemmeke and T. Virtanen, “Noise robust exemplar-based connected digit recognition,” in *Proc. of ICASSP*, Dallas, TX, March 2010, pp. 4546–4549.

[7] B. Schuller, F. Weninger, M. Wöllmer, Y. Sun, and G. Rigoll, “Non-negative matrix factorization as noise-robust feature extractor for speech recognition,” in *Proc. of ICASSP*, Dallas, TX, March 2010, pp. 4562–4565.

[8] B. Schuller and F. Weninger, “Discrimination of speech and non-linguistic vocalizations by non-negative matrix factorization,” in *Proc. of ICASSP*, Dallas, TX, March 2010, pp. 5054–5057.

[9] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proc. of NIPS*, Vancouver, Canada, 2001, pp. 556–562.

[10] M. N. Schmidt and R. K. Olsson, “Single-channel speech separation using sparse non-negative matrix factorization,” in *Proc. of Interspeech*, Pittsburgh, PA, 2006.

[11] A. Hyvärinen, “Fast and robust fixed-point algorithms for Independent Component Analysis,” *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[12] M. Frigo and S. G. Johnson, “The design and implementation of FFTW3,” *Proc. of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[13] R. C. Whaley, A. Petitet, and J. Dongarra, “Automated Empirical Optimization of Software and the ATLAS project,” *Parallel Computing*, vol. 27, no. 1-2, pp. 3–35, 2001.