APPROXIMATE CONVOLUTION USING PARTITIONED TRUNCATED SINGULAR VALUE DECOMPOSITION FILTERING

Joshua Atkins, Adam Strauss, Chen Zhang

Beats Electronics, LLC
1601 Cloverfield Blvd. Suite 5000N
Santa Monica, CA 90404

ABSTRACT

In many signal processing applications it is necessary to perform large convolutions in real-time. For systems where an exact convolution is too complex we propose an approximation using a partitioned truncated singular value decomposition (PTSVD) filter. In this method the filter is first partitioned into P segments of length N, the singular value decomposition is performed on the N $\times$ P matrix, and only the largest M singular values and associated vectors are used to reconstruct the filter. We show an efficient real-time implementation utilizing a filter bank and tapped delay line and then further simplify the structure utilizing an IIR model. Finally, we show an application of the method in a simulated reverberation engine and compare complexity and memory load to state of the art methods.

Index Terms— Convolution, SVD, Filtering

1. INTRODUCTION

Traditional audio signal processing problems both in telecommunications and multimedia often rely on FIR filter models, e.g. for the room impulse response, that can be very large and, consequently, difficult to implement in practice. State of the art techniques for implementing these filters in real-time systems use the overlap-add or overlap-save methods and partitioned frequency domain convolution to reduce complexity and delay [1, 2, 3, 4]. However, frequency domain techniques are inherently block based and introduce an amount of system latency. Alternative methods have been explored in certain application domains, such as using a perceptual model to remove certain time-frequency data from processing [5] or subband decomposition of the impulse response [6]. For short impulse responses (on the order of a few 100 FIR coefficients), IIR methods are attractive ways for reducing complexity [7, 8, 9], but these methods fail for longer filters.

In this paper we propose an alternative idea, partitioned truncated singular value decomposition (PTSVD) filtering, where the impulse response is partitioned in time, factorized using the singular value decomposition (SVD), and then reconstructed using only the M singular vectors corresponding to the M largest singular values. This filtering method was initially explored by Mitra et. al. in [10, 11] for the purpose of creating efficient versions of linear phase bandpass and lowpass FIR prototypes. The image processing community also has used the truncated SVD for 2D filter design [12, 13].

In this work we provide additional analysis of the method shown in [10, 11], extending it to systems that are not guaranteed to be linear phase and analyzing the tradeoff in complexity, memory usage, and approximation error. In Section 3 we propose a filter structure that takes advantage of the truncated SVD matrices and leads to an efficient implementation. We then show a further approximation that both reduces the memory footprint and the computational complexity using an IIR input and output filter. This filtering method not only has the benefit of reduced memory and complexity over traditional methods, it is also delay-less since it does not require a block-based processing structure.

2. FILTER APPROXIMATION

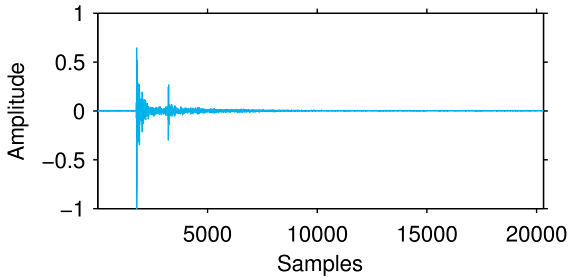
Let $\mathbf{h} = [h(0)h(1)\dots h(L-1)]$ be an impulse response of length L. We can construct a N $\times$ P matrix $\mathbf{H}$ by partitioning $\mathbf{h}$ into P partitions of length $N = \lceil L/P \rceil$ (and zero padding $\mathbf{h}$ if necessary so that it is of length P $\times$ N). $\mathbf{H}$ can then be factored using SVD [14] as

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^H,$$

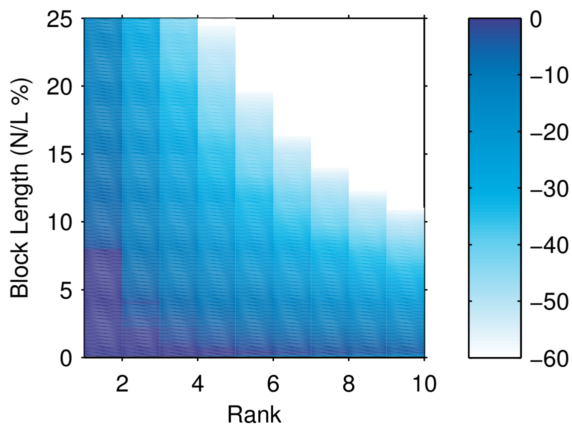
where $(\cdot)^H$ is the conjugate transpose, $\mathbf{U}$ and $\mathbf{V}$ are the N $\times$ N and P $\times$ P singular vectors which form a basis for the factorization, and $\mathbf{S}$ is a N $\times$ P matrix containing the singular values along its main diagonal. We will assume that the singular values are in descending order.

We can create a M$^{\text{th}}$ order approximate filter, $\mathbf{H}_M$, by using only the M largest singular values in its reconstruction. This is done by truncating $\mathbf{U}$ and $\mathbf{V}$ to be of size N $\times$ M and P $\times$ M, respectively, and taking the M $\times$ M portion of $\mathbf{S}$ corresponding to the largest singular values. The approximate filter is then

$$\mathbf{H}_M = \mathbf{U}_M\mathbf{S}_M\mathbf{V}_M^H. \quad (1)$$



(a) Impulse response



(b) Error (in dB)

Fig. 1: A typical room impulse response (RT60 = 400ms) and the PTSVD approximation error.

The error in our M$^{\text{th}}$ order approximation is given as

$$e(M, N) = ||\mathbf{H} - \mathbf{H}_M||_2,$$

where $||\cdot||_2$ is the entry-wise ℓ_2-norm. The use of the SVD guarantees that $\mathbf{H}_M$ is the rank-M reconstruction of $\mathbf{H}$ with the lowest error, $e(M, N)$.

At this point we have two free parameters, M and N, that determine the error in our approximation filter. Figure 1 shows the error surface for a typical room impulse response. This example shows that there exist very low rank approximations of the original filter that achieve minimal error.

3. EFFICIENT FILTER STRUCTURE

We now will show how the filter $\mathbf{H}_M$ generated in the last section can be implemented efficiently for real-time applications. First, we will rewrite Equation 1 in an expanded form as

$$\mathbf{H} = \begin{bmatrix} \sigma_0 \mathbf{u}_0 \\ \sigma_1 \mathbf{u}_1 \\ \cdots \\ \sigma_{M-1} \mathbf{u}_{M-1} \end{bmatrix}^T \begin{bmatrix} v_0^0 & v_0^1 & \cdots & v_0^{P-1} \\ v_1^0 & v_1^1 & \cdots & v_1^{P-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M-1}^0 & v_{M-1}^1 & \cdots & v_{M-1}^{P-1} \end{bmatrix}$$

where σ_m are the singular values from $\mathbf{S}_M$ and $\mathbf{u}_m$ are the length N singular vectors of $\mathbf{U}_M$. Recognizing that the P columns of $\mathbf{H}_M$ are the time-partitioned version of the filter, each delayed N samples from the last, we can write a filter implementation of $\mathbf{H}$ as

$$y(n) = \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} v_m^p \sigma_m \mathbf{u}_m^T \mathbf{x}(n - pN)$$

where $\mathbf{x}(n)$ is the vector of the last N samples of x at time step n. Figure 2 shows this filter structure, which resembles a filter-bank analysis section with M length N filters each followed by a tapped delay line of length P. Note that this filter structure achieves a lower complexity implementation of the FIR filter when a low rank approximation is used, but the memory usage is increased significantly to store the M delay lines (each the same length as the original filter). This will limit system performance in real applications where memory bandwidth is an issue.

The representation in Figure 2 can be further optimized by modeling the input and output filters using an IIR approximation [7, 8, 9]. This reduces multiply-add instructions, but also reduces memory storage significantly since the M length L delay lines do not need to be stored due to the recursive IIR structure. Figure 3 shows IIR approximation error in the first 4 $\mathbf{u}_m$ and $\mathbf{v}_m$ filters for the reverberation filter shown in Figure 1 using a partition size of $N = 53$. The IIR approximations were designed using invfreqz in MATLAB, which uses an equation-error method for an initial coefficient guess followed by an iterative scheme to minimize the solution-error.

4. COMPLEXITY ANALYSIS

The PTSVD filter structure proposed in the last section in its initial form requires $M \times (N + P)$ multiply-add instructions per input sample and memory of size $M \times (N + P + L) + N$. A conventional time-domain FIR filter requires L operations per input sample and $2L$ variables. Implementing the FIR filter with partitioned convolution (partition size = N) greatly reduces complexity: $4\alpha \log_2(2N) + 4P + 1$ instructions per sample and $4PN$ variables for overlap-add, where α is a platform specific FFT cost [3].

Figure 4 shows the complexity and memory of the partitioned convolution and PTSVD at various rank-M approximations ($N = 128$ is assumed). These are shown as a percent of the complexity and memory of the time-domain FIR implementation, so values above 100% provide no savings. For nearly all filter lengths the PTSVD approach is lower complexity than a traditional FIR, showing significant benefits when the filter length becomes large.

It is also clear from this graph that the partitioned convolution (dashed line) is more efficient than the PTSVD for filters less than 10,000 coefficients and $M > 2$.

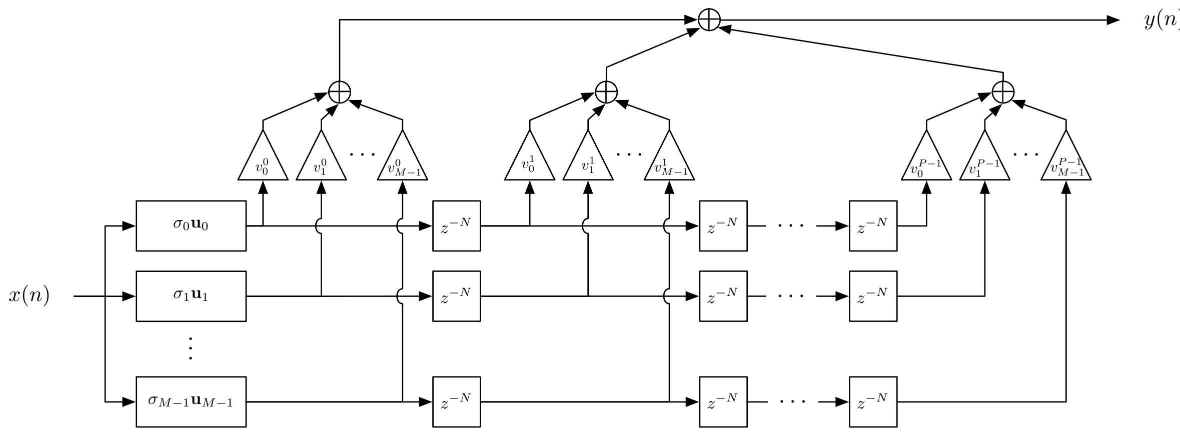


Fig. 2: The structure of the PTSVD filter.

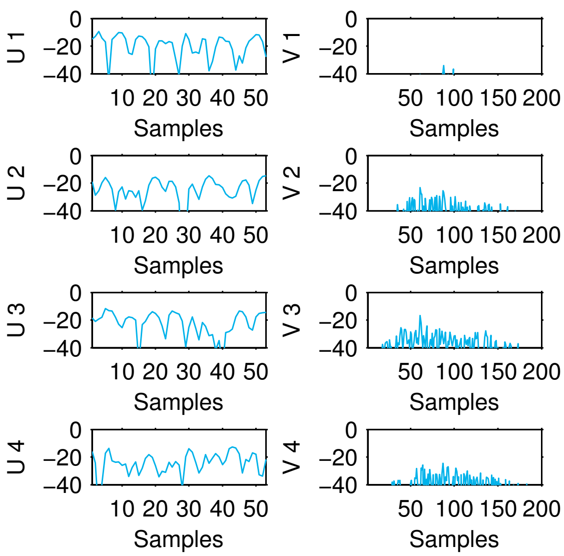


Fig. 3: The IIR approximation error (dB) in the first 4 $\mathbf{u}_m$ and $\mathbf{v}_m$ filters for the reverberation filter shown in Figure 1 using 9^{th} order and 41^{st} order, respectively.

As mentioned in Section 3, the PTSVD structure becomes very efficient when the input and output filters are modeled with an IIR approximation. This results in a structure with $2.5M(Q_U + Q_V)$ instructions and $3.5M(Q_U + Q_V)$ variables, where Q_U and Q_V are the IIR approximation orders of the U and V filters (direct-form II transpose using second order sections is assumed). The complexity and memory of the PTSVD-IIR are shown in Figure 4 ($Q_U + Q_V = 60$ is assumed). From this plot it is clear that the method can both significantly save memory usage as well as complexity for filters of length 1,000 coefficients or more.

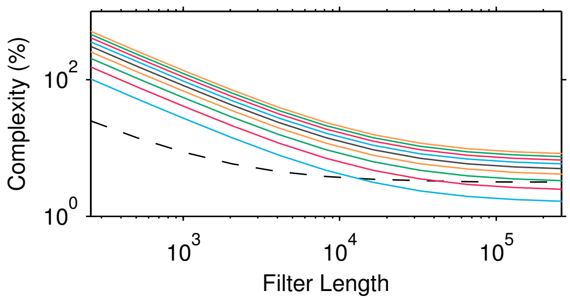
Note that these graphs have fixed $N = 128$ and $Q_U + Q_V = 60$ and thus don't show the full picture. However, they provide a reasonable view of where the proposed method becomes useful in real systems where 128 samples is a common frame size and $Q_U + Q_V = 60$ is a typical combined IIR approximation order above which the error becomes negligible.

Furthermore, the PTSVD filter structure also permits other models which may achieve better performance in certain contexts, such as using frequency domain processing for the $\mathbf{u}_m$ or $\mathbf{v}_m$ filters, using an IIR model for only one section, or using varying IIR approximation orders for each $\mathbf{u}_m$ or $\mathbf{v}_m$, which are not analyzed in this work.

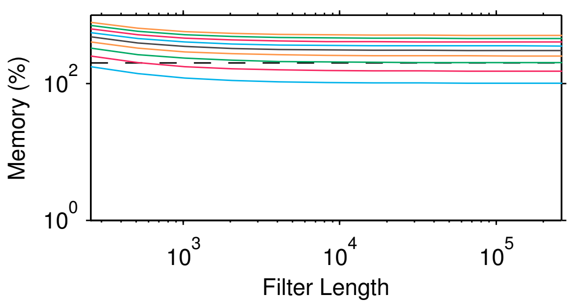
5. SIMULATION

Since the N, M, Q_U, and Q_V are all integer valued, it is possible to calculate the finite set of (n_i, m_i, qu_i, qv_i) that meet a given memory and complexity requirement on a particular platform. Although the choice of error metric should be application dependent (e.g. a spectro-temporal metric for audio applications), as a simple choice, the point that results in the lowest ℓ_2 approximation error can be chosen.

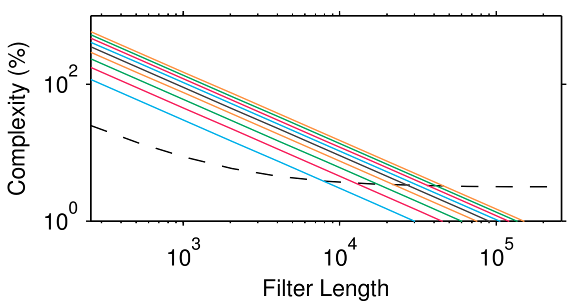
As an example, the impulse response from Figure 1 is approximated by the PTSVD-IIR method. A search over possible N, M, Q_U, and Q_V with a maximum complexity of 500 operations per sample and memory usage of 1000 variables was performed. The resulting filter design used the parameters $N = 53$, $M = 4$, $Q_U = 9$, and $Q_V = 41$ resulting in a complexity of 500 operations per sample and memory usage of 700 variables. Figure 5 shows the error for the PTSVD filter using this approximation and the PTSVD-IIR filter. The resulting U and V filters and their IIR approximations are shown in Figure 3. For reference, the time domain FIR version of this filter requires 20,315 operations per sample and 40,630 variables and a partitioned convolution with



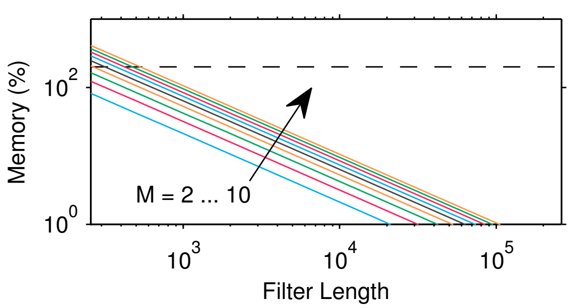
(a)



(b)

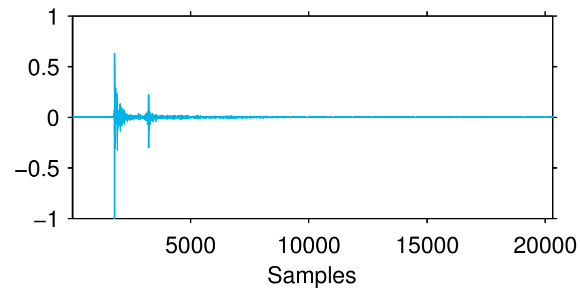


(c)

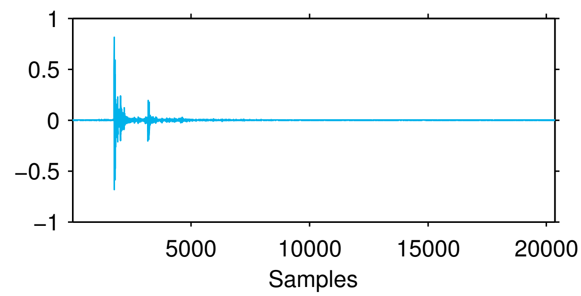


(d)

Fig. 4: The complexity and memory usage, expressed as percent of time-domain FIR implementation, for the PTSVD and frequency domain partitioned convolution (dashed). Figures (a) and (b) are for time domain PTSVD and (c) and (d) are for PTSVD using an IIR model of the filters ($Q_U + Q_V = 60$).



(a) PTSVD



(b) PTSVD-IIR

Fig. 5: The error in the PTSVD approximation and PTSVD-IIR approximation for the filter in Figure 1 using $N = 53$, $M = 4$, $Q_U = 9$, and $Q_V = 41$.

$N = 53$ requires 1,583 operations per sample and 81,408 variables (assuming $\alpha = 1.7$). This is a 98% improvement over regular FIR filtering and 68% improvement over partitioned convolution in addition to the benefit of no block delay (53 samples).

6. CONCLUSIONS

In this paper we have shown how a conventional convolution can be approximated with a lower complexity rank-M filter that is optimal in the ℓ_2 sense. An efficient filter structure for the approximation and an IIR implementation that delivers low-error results with minimal complexity and memory usage for real-time systems was presented.

The method opens up many possible avenues for further study including alternative low-rank approximations (as opposed to the SVD), joint spatio-temporal filter design for spatial audio rendering and beamforming (explored in [15]), and adaptive implementation for applications such as echo cancelation. Adaptations of the PTSVD filtering structure, such as varying IIR approximation orders for each $\mathbf{u}_m$ and $\mathbf{v}_m$ filter, frequency domain SVD analysis, and combination with subband methods, will be discussed in a future work.

7. REFERENCES

[1] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing*, Prentice Hall, Upper Saddle River, New Jersey, Jan. 1999.

[2] W. G. Gardner, "Efficient Convolution without Input-Output Delay," *Journal of the Audio Engineering Society*, vol. 43, no. 3, pp. 127–136, Feb. 1995.

[3] F. Wefers and M. Vorländer, "Optimal Filter Partitions for Real-Time FIR Filtering Using Uniformly-Partitioned FFT-Based Convolution in the Frequency Domain," in *14th International Conference on Digital Audio Effects*, 2011, pp. 155–161.

[4] V. Valimaki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty Years of Artificial Reverberation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1421–1448, July 2012.

[5] W. Lee, C. Liu, C. Yang, and J. Guo, "Fast perceptual convolution for room reverberation," in *Proc. of the 6th Int. Conference on Digital Audio Effects*, London, 2003, number 6, pp. 6–9.

[6] F. Jabloun and B. Champagne, "A Fast Subband Room Response Simulator," *IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 925–928, 2000.

[7] K. Steiglitz and L. E. McBride, "A Technique for the Identification of Linear Systems," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 461–464, 1965.

[8] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

[9] E. C. Levi, "Complex-Curve Fitting," *IRE Trans. on Automatic Control*, vol. AC-4, pp. 37–44, 1959.

[10] S. K. Mitra, M. D. Grosen, and Y. Neuvo, "Design of Computationally Efficient FIR Filters Using Singular Value Decomposition," in *Proc. 1985 China International Conference on Circuits and Systems*, Beijing, China, 1985, pp. 268–271.

[11] S. K. Mitra, M. D. Grosen, and Y. Neuvo, "Design of Fault-Tolerant 1D FIR Digital Filters Using Outer Product Decompositions," in *Proc. European Conference on Circuit Theory and Design*, Prague, Czechoslovakia, 1985, pp. 545–548.

[12] T. Deng and E. Okamoto, "SVD-Based Design of Fractional-Delay 2-D Digital Filters Exploiting Specification Symmetries," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 50, no. 8, pp. 470–480, 2003.

[13] E. Biglieri and K. Yao, "Some properties of singular value decomposition and their applications to digital signal processing," *Signal Processing*, vol. 18, no. 3, pp. 277–289, Nov. 1989.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, Jan. 1985.

[15] J. Atkins, A. Strauss, and C. Zhang, "Approximate Convolution Using Partitioned Truncated Singular Value Decomposition Filtering for Binaural Rendering," in *International Congress on Acoustics*, Montreal, Canada, 2013.