

Article

Late Reverberation Synthesis Using Filtered Velvet Noise [†]

Vesa Välimäki *, Bo Holm-Rasmussen [‡], Benoit Alary and Heidi-Maria Lehtonen [‡]

Acoustics Lab, Department of Signal Processing and Acoustics, Aalto University, 02150 Espoo, Finland; bo.holm.rasmussen@gmail.com (B.H.-R.); benoit.alary@aalto.fi (B.A.); hmleht@gmail.com (H.-M.L.)

* Correspondence: vesa.valimaki@aalto.fi; Tel.: +358-50-5691-176

[†] This paper is a revised and extended version of a paper published in the International Conference on Digital Audio Effects (DAFX), Maynooth, Ireland, 2–5 September 2013.

[‡] Current addresses: Bo Holm-Rasmussen—The Royal Danish Academy of Fine Arts, Laboratory for Sound, 1050 Copenhagen, Denmark. Heidi-Maria Lehtonen—Dolby Sweden, 11330 Stockholm, Sweden.

Academic Editors: Woon-Seng Gan and Jung-Woo Choi

Received: 15 March 2017; Accepted: 3 May 2017; Published: 6 May 2017

Abstract: This paper discusses the modeling of the late part of a room impulse response by dividing it into short segments and approximating each one as a filtered random sequence. The filters and their associated gain account for the spectral shape and decay of the overall response. The noise segments are realized with velvet noise, which is sparse pseudo-random noise. The proposed approach leads to a parametric representation and computationally efficient artificial reverberation, since convolution with velvet noise reduces to a multiplication-free sparse sum. Cascading of the differential coloration filters is proposed to further reduce the computational cost. A subjective test shows that the resulting approximation of the late reverberation often leads to a noticeable difference in comparison to the original impulse response, especially with transient sounds, but the difference is minor. The proposed method is very efficient in terms of real-time computational cost and memory storage. The proposed method will be useful for spatial audio applications.

Keywords: audio systems; digital filters; digital signal processing; room acoustics

1. Introduction

Artificial reverberation research started in the 1960s, when Schroeder developed the first methods to simulate the room effect with a computer [1,2]. His methods plus numerous other approaches, which were introduced thereafter, have been reviewed by Gardner [3] and recently in a series of two papers by Välimäki et al. [2,4].

Concert halls and listening rooms are often considered to be linear and time-invariant systems. Therefore, it should be possible to fully reproduce their sonic characteristics by replicating the impulse response, which is measured between a source and a listening point. A room impulse response (RIR) is often divided into three phases: the direct (or dry) sound, early reflections, and the late reverberation. This paper focuses on the modeling of the late reverberation, which is noise-like and contains the contribution of a large number of reflections.

Convolution with a measured RIR is a popular technique resulting in very realistic reverberation [2,4,5]. However, convolution is computationally intensive, and modification or parameterization of the measured RIR can be cumbersome. Partitioned fast convolution methods [6–9] reduce the computational complexity considerably compared to direct convolution and avoid most of the delay introduced by the basic fast convolution, which corresponds to a full-scale FFT (Fast Fourier transform)-based implementation. Moorer suggested that the late part of the RIR can be well characterized as exponentially decaying white noise [10]. This observation led to useful applications when

Rubak and Johansen used a finite-impulse response (FIR) filter with random coefficients in a recursive reverberation algorithm [11,12]. Karjalainen and Järveläinen developed an improved algorithm in which a random coefficient FIR filter is cascaded with a lowpass comb filter [13]. They also introduced velvet noise, which is smooth-sounding ternary random noise [13]. Later, Lee et al. [14] and Oksanen et al. [15] investigated alternative recursive reverberator structures using velvet noise.

This paper focuses on room reverberation modeling using velvet noise, extending our previous work [16,17]. The RIR is divided into short segments and each of them is approximated as a filtered velvet noise (FVN) sequence. The coloration filters and their associated gain account for the spectral shape and level of each RIR segment, so together they enable the approximation of a given frequency-dependent decay behavior in the time domain. Finally, cascaded Schroeder allpass filters are used to obtain a smooth, wideband, noise-like response. This approach is thus orthogonal to the modal filter bank idea, which divides the RIR into slices in the frequency dimension [18,19], and to Jot's idea of estimating the reverberation time across frequency bands [20] and calibrating a feedback delay network reverberator [21,22]. Such methods are best suited for exponentially decaying responses.

This FVN approach leads to a parametric representation and computationally efficient RIR synthesis, since convolution with velvet noise is economical to implement. A novel idea is proposed to cascade the coloration filters, so that the effect of all filtering operations of the previous stages are accounted for by using differential filters in the subsequent stages.

The rest of this paper is organized as follows: Sections 2 and 3 discuss velvet noise and the basic version of the FVN method, respectively, and Section 4 describes a new differential filtering strategy and an impulse response segmentation strategy for it. Section 5 shows how well the algorithm can synthesize the impulse response of a real concert hall, and how the synthetic response can be modified. Section 6 compares the computational complexity and memory usage with other algorithms, and Section 7 presents a subjective evaluation of the proposed method. Section 8 concludes this paper.

2. Velvet Noise

Velvet noise is a special kind of random noise, which was discovered by Karjalainen and Järveläinen [13]. It consists of sample values -1 , 0 , and 1 only. The most surprising attribute of velvet noise is that even when 95% of its samples are zero, it sounds smoother than Gaussian random noise, which is generally thought to be the prototype of white noise [13,23]. Velvet noise is of interest in this work, because it provides a computationally efficient way to convolve an arbitrary signal with white noise [16].

2.1. Generation of Velvet Noise

Velvet noise can be interpreted as a randomly jittered impulse train in which the sign of each impulse is chosen randomly to be positive or negative [23]. To generate velvet noise, one should first select the pulse density N_d , i.e., the number of impulses per second. It yields the main design parameter, the average distance between impulses T_d , as:

$$T_d = \frac{f_s}{N_d}, \quad (1)$$

where f_s is the sample rate. Other randomization techniques have also been proposed, for example the totally random ternary sequence by Rubak and Johansen [11], which does not include any rule to limit how close to or far away from each other two neighboring impulses can occur. However, it is not perceived to be as smooth as velvet noise at low pulse densities [23]. The restriction of having only one impulse within every T_d samples appears to be an economical choice, which minimizes roughness [13,23].

In velvet noise, the impulse locations $k(m)$ are determined as:

$$k(m) = \text{round}[mT_d + r_1(m)(T_d - 1)], \quad (2)$$

where $m = 0, 1, 2, \dots$ is the pulse counter and $r_1(m)$ is a value produced with a random-number generator with uniform distribution (0,1). The term -1 at the end of Equation (2) helps to avoid coinciding pulses [23].

The complete velvet-noise sequence can then be written as:

$$s(n) = \begin{cases} 2 \text{round}[r_2(m)] - 1, & \text{when } n = k(m), \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where n is the sample index, $k(m)$ are the impulse locations determined using Equation (2), and $r_2(m)$ is the value of a second random-number generator with uniform distribution (0,1) used to select the sign of each impulse [23].

When the sample rate of 44,100 Hz is used, the choice of $N_d = 2205$ pulses/s, according to Equation (2), leads to a convenient integer value of $T_d = 20$ samples for the average pulse distance. Figure 1a shows the first 500 samples of an example velvet-noise sequence with these parameters. There is only one non-zero sample seen between any two grid boundaries. The autocorrelation function of the velvet-noise sequence shown in Figure 1b is close to a unit impulse, as its maximum occurring at $n = 0$ is 1.0 and at other lags the correlation is smaller than about 0.01. The power spectrum of the velvet-noise sequence shown in Figure 1c is fairly flat.

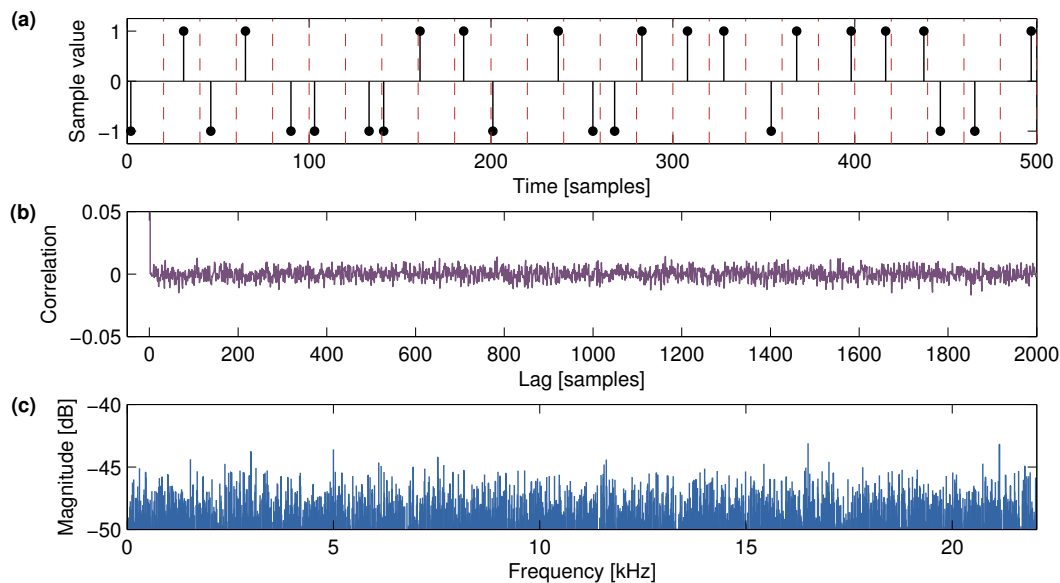


Figure 1. (a) Non-zero sample values, (b) the autocorrelation function, and (c) the estimated spectrum of a velvet-noise sequence. In (a), the vertical dashed lines indicate the grid boundaries. In (b), the value of autocorrelation at zero lag is 1.0, but this first value is truncated in the figure.

2.2. Velvet-Noise Convolution

Time-domain convolution of a signal with velvet noise can be highly economical computationally. The samples of the velvet-noise sequence $s(n)$ are used as FIR filter coefficients. Velvet-noise convolution (VNC) is very fast to compute, because all multiplications by zero can be dispensed as their locations in the sequence are known. Additionally, as the non-zero samples contained in the velvet noise are either -1 or 1 , multiplications are not needed. Thus, convolution with velvet noise reduces to a sparse multiplication-free convolution.

In practice, then, the input signal is propagated in the delay line of the filter, and only those input signal samples which coincide with the non-zero coefficients of the velvet-noise sequence are added together to produce the output. One idea is to separately run through the indices of coefficient values $+1$ and -1 , add the corresponding sample values taken from the delay line, and subtract the two sums. This VNC process can be formulated as:

$$x(n) * s(n) = \sum_{m_+} x[n - k(m_+)] - \sum_{m_-} x[n - k(m_-)], \quad (4)$$

where $x(n)$ is the input signal, $*$ denotes the convolution, and $k(m_+)$ and $k(m_-)$ contain the indices of the positive and negative impulses, respectively, in the velvet-noise sequence $s(n)$. This multi-tap delay-line implementation of VNC is illustrated in Figure 2.

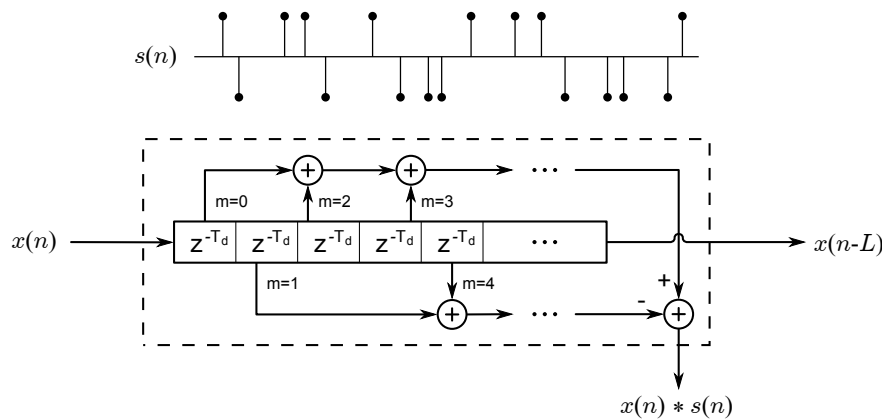


Figure 2. Velvet-noise convolution: Convolving the signal $x(n)$ with a velvet-noise sequence $s(n)$ reduces to the multiplication-free process of computing two sparse sums of delayed input signal samples and their difference. Blocks containing z^{-T_d} , where z is the complex variable of the Z transformation, refer to delay lines of T_d samples. The output tap of each delay-line element is located at the sample point determined by sequence $s(n)$.

For example, when 5% of the velvet noise coefficients are non-zero ($+1$ or -1) and the filter length is L samples, computing an output sample requires $0.05L$ additions and no multiplications. For a $1 - s$ noise sequence at the 44.1-kHz sample rate, the filter length is $L = 44,100$, and this yields 2205 additions per output sample. For comparison, a regular FIR filter of the same length requires $L - 1 = 44,099$ additions and $L = 44,100$ multiplications, or 88,199 operations, to compute each output sample, which is 40 times more than using VNC.

3. Filtered Velvet Noise Reverberation Algorithm

The key idea of the FVN reverberation algorithm is to divide the RIR into short non-overlapping segments and to approximate each segment as filtered white noise. Velvet noise is used instead of standard white noise, such as Gaussian noise, since then the convolution with the input signal is fast to compute.

Figure 3 shows the block diagram of the basic FVN reverberation algorithm. The delay lines of each VNC block serve two purposes: they delay the input signal appropriately for the next stage, as indicated by the right-hand-side output signal $x(n - L)$ in Figure 2, and they provide the state variables of the sparse multi-tap delay line used to implement the VNC, i.e., a very efficient multiplication-free convolution of input signal with the velvet-noise sequence. The sparse sum of each segment is next filtered by its own spectral coloration filter $H_m(z)$ and attenuated appropriately by the gain term G_m , as shown in Figure 3.

Uniform segmentation of an RIR should not be used, as the constant frame rate causes a periodic disturbance in the synthetic response. This is reminiscent of the flutter echo effect, which is a common problem in room acoustics. Much effort has been made to reduce this effect in recursive reverberation algorithms that use a pseudo-random noise sequence [13,14]. Thus, it makes sense to use a non-uniform segmentation scheme in the FVN algorithm, as suggested in [16]. Another motivation to use a non-uniform framing is that the filter for each segment would be sufficiently different. In a typical RIR in which the exponential decay is faster at high frequencies than at low, a constant decrease in bandwidth, such as a 1-kHz narrowing, takes place non-uniformly in time—quickly in the beginning and slower towards the end of the RIR. This also motivates the use of longer segments at the end than at the beginning of the RIR. Figure 4a shows an example of a RIR and its segmentation. The impulse response has been measured in the concert hall in Pori, Finland (this impulse response of the concert hall is available online at <http://legacy.spa.aalto.fi/projects/poririrs/>).

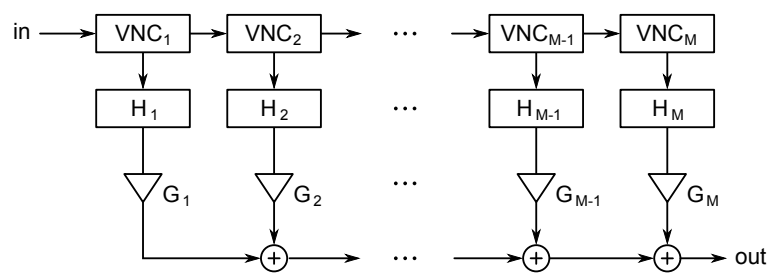


Figure 3. Basic principle of the filtered velvet noise (FVN) algorithm [16]. The delay lines between the filtering branches of length L_m are in practice combined with velvet-noise convolution (VNC) blocks, cf. Figure 2. Blocks H_m and G_m , for $m = 1, 2, \dots, M$, represent the spectral coloration filters and gain factors for each segment, respectively.

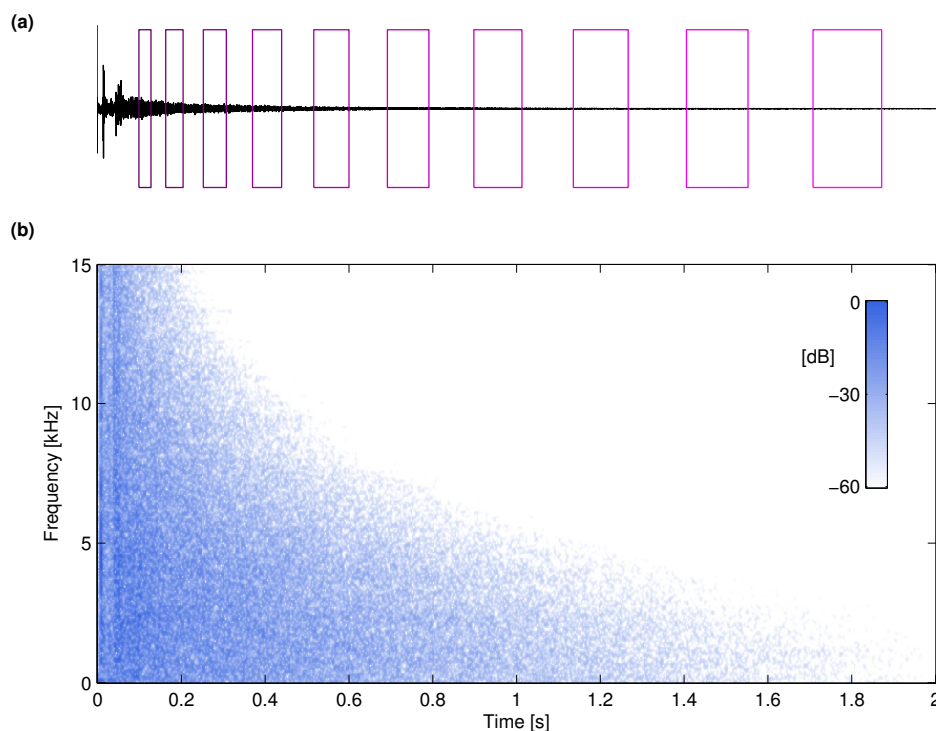


Figure 4. (a) Measured room impulse response (RIR) of the concert hall in Pori, Finland, with the boxes indicating every second segment used for modeling, and (b) its spectrogram showing frequency-dependent decay.

3.1. Coloration Filters

To design the spectral coloration filter $H_m(z)$ linear prediction (LP) can be used for each segment [24]. The coloration filters should match the overall lowpass characteristic of each short segment. For this reason, low-order LP is sufficient in this application. Prediction order 10 is used in this work, which leads to 10th-order all-pole coloration filters. Figure 5 shows examples of coloration filters estimated for the RIR of the Pori concert hall. The overall shape of the responses follows the frequency-dependent decay, as expected.

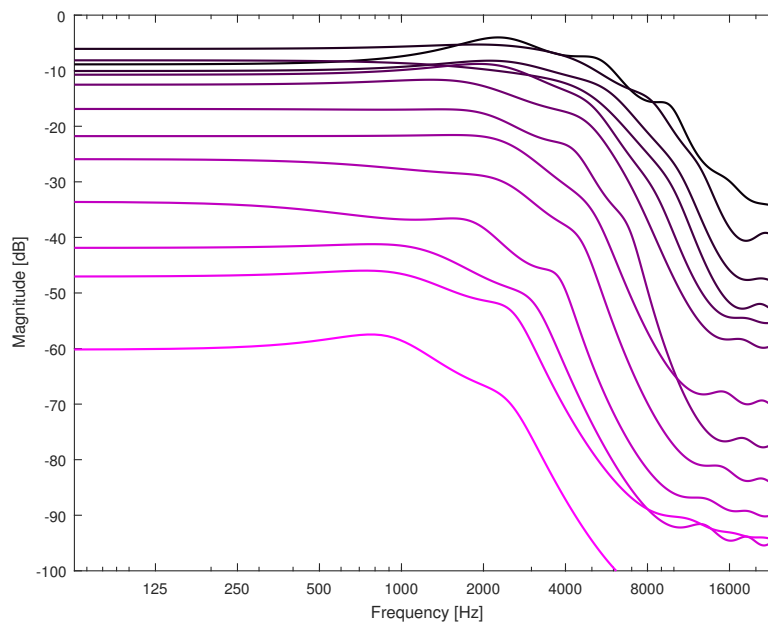


Figure 5. Magnitude responses of coloration filters of order 10 for every second segment of the impulse response of the Pori concert hall. The same color codes as in Figure 4a are used such that the darker lines correspond to the beginning of the RIR and the color gets lighter towards the end of the RIR.

Since only one lowpass filter and one gain coefficient are required per segment, the computation of the VNC becomes the most demanding part of the structure. For this reason, ways to reduce the pulse density without sacrificing the sound quality were investigated. Karjalainen and Järveläinen [13] showed that the sufficient pulse density is lower for lowpass-filtered velvet noise than in the full audio band: in particular, for a cutoff frequency of $f_c = 1.5$ kHz, the lowpass-filtered velvet noise sounds smoother than Gaussian white noise even with the lowest pulse density they tested, 600 pulses/s. Since the bandwidth of the RIR becomes narrower towards its end, the pulse density of velvet noise may also be decreased from one segment to another. Figure 4b clearly shows the narrowing of the bandwidth (blue area) of a measured RIR over time.

3.2. Schroeder Allpass Filters

In order to further smooth the synthetic RIR, a cascade of Schroeder allpass (SAP) filters is used. This allows further reduction of the pulse density in VNC. Each SAP filter has the following transfer function [1]:

$$A(z) = \frac{a + z^{-N}}{1 + az^{-N}}, \quad (5)$$

where $-1 < a < 1$ is the allpass filter coefficient and N is the delay-line length. Figure 6 shows the structure of the FVN algorithm when the total sum of all branches is further processed with a cascade of filters, SAP_1 to SAP_K .

Figure 7 shows the spectrogram of a velvet-noise sequence having only 44 non-zero samples per second and that of a SAP filter consisting of four cascaded filters. The delay line lengths of the SAP filters are 225, 341, 441, and 556 samples, and their filter coefficient is $a = 0.7$. The rightmost spectrogram is the result of convolving the velvet-noise sequence with the SAP filter's response, showing a wideband noise-like behavior. This example shows that the gaps in velvet noise can be filled by cascading SAP filters. The spectrograms in Figure 7 were generated using a 600-sample Hann window with 500 samples of overlap.

By experimenting with different pulse densities and listening to the outcome, it was decided that $N_d = 100$ pulses/s is sufficient in the very beginning of the late reverberation, where segments are very short, whereas $N_d = 40$ pulses/s can be enough at the end where the bandwidth gets narrow. Between these extremes, the density is decreased linearly as a function of the segment index m . The selected pulse density for each segment is shown in Figure 8.

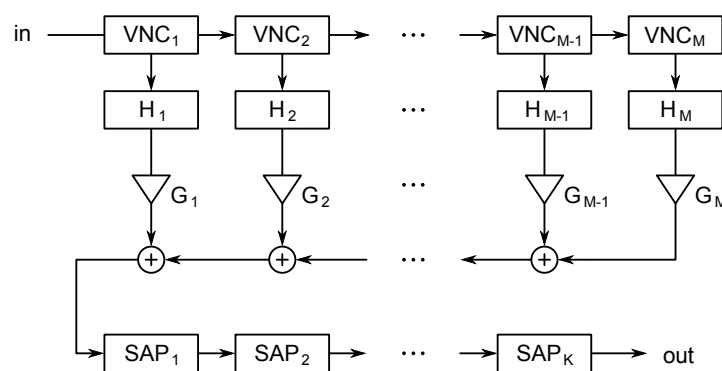


Figure 6. FVN algorithm with Schroeder allpass filters (SAP) [16].

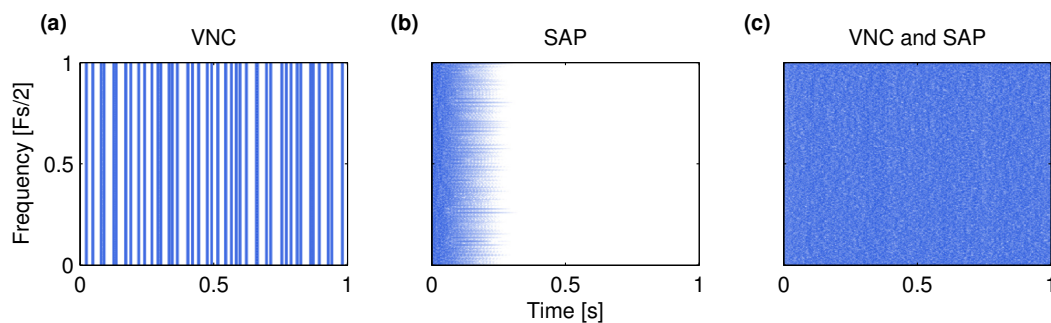


Figure 7. Spectrograms of (a) a velvet-noise sequence, (b) the impulse response of four cascaded SAP filters, and (c) their convolution. White corresponds to 60 dB lower level than blue.

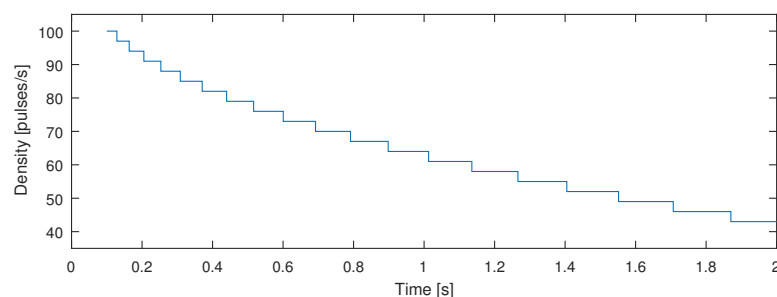


Figure 8. Pulse density and length of each segment for the Pori hall. The pulse density can be decreased towards the end of the RIR.

3.3. Segment Gains

Finally, the gain G_m for each segment, as shown in Figure 6, must be determined so that the overall decay rate of the RIR model is preserved. To ensure that this is the case, an analysis–synthesis approach is used. Each RIR segment is first whitened with the LP inverse filter obtained using the 10th-order LP, and the average signal power of this filtered signal segment is calculated to establish a reference. Then a long sequence (e.g., one second) of velvet noise with the pulse density assigned to that segment is processed with the all-pole coloration filter and with the cascade of SAP filters. The average signal power of this filtered velvet noise is then calculated, and the gain of this segment, G_m , is set based on the ratio of this signal power to the reference signal power. This routine ensures that the gain of each segment is adjusted accurately.

4. Advanced FVN Algorithm

In this section we elaborate on the basic FVN method: coloration filters are redesigned so that they can be cascaded, which helps reduce the filter order for each segment.

4.1. Differential Coloration Filters

Since the cutoff frequency of the filters in each segment usually decreases towards the end of the RIR, it is possible to exploit the previous filters in the subsequent filtering stages. The basic idea is to design the first lowpass coloration filter $H_1(z)$ but to construct the other filters by cascading differential filters $\Delta H_m(z)$, for $m \geq 2$. This structure is illustrated at the top of Figure 9.

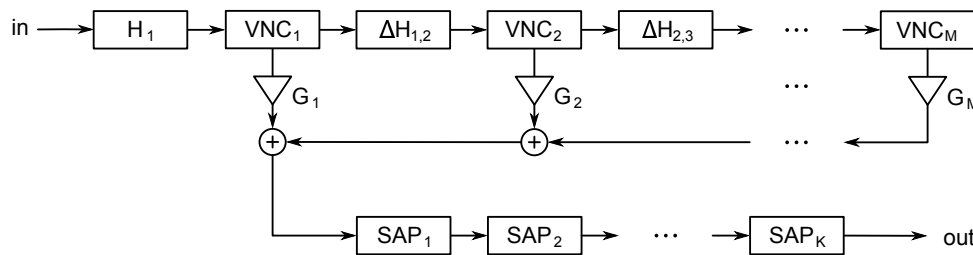


Figure 9. Advanced FVN algorithm with cascaded differential coloration filters $\Delta H_m(z)$.

The first filter can be designed manually to imitate the spectral shape of the initial RIR segment, which has a fairly flat spectrum. Here we use a 10th-order all-pole filter obtained with linear prediction. The magnitude response of this filter is shown in Figure 10a.

The differential filters are second-order notch filters with the transfer function $H(z) = 1 + (V_0 - 1)[1 - A_2(z)]/2$ with

$$A_2(z) = \frac{-c + d(1 - c)z^{-1} + z^{-2}}{1 + d(1 - c)z^{-1} - cz^{-2}}, \quad (6)$$

where $c = [\tan(\pi f_b/f_s) - V_0]/[\tan(\pi f_b/f_s) + V_0]$, $d = -\cos(2\pi f_c/f_s)$ for $0 < V_0 < 1$ is the attenuation at the center frequency f_c and f_b is the bandwidth of the notch (Hz) [25]. The differential filters can be designed to match the difference between the neighboring coloration filters. Figure 10b shows responses of the notch filters designed from the family of 10th-order coloration filters. Figure 10c shows the total effect of cascading 1 to $M - 1$ of these filters with the first filter $H_1(z)$. The overall shapes and cutoff points are very similar to the responses shown in Figure 5.

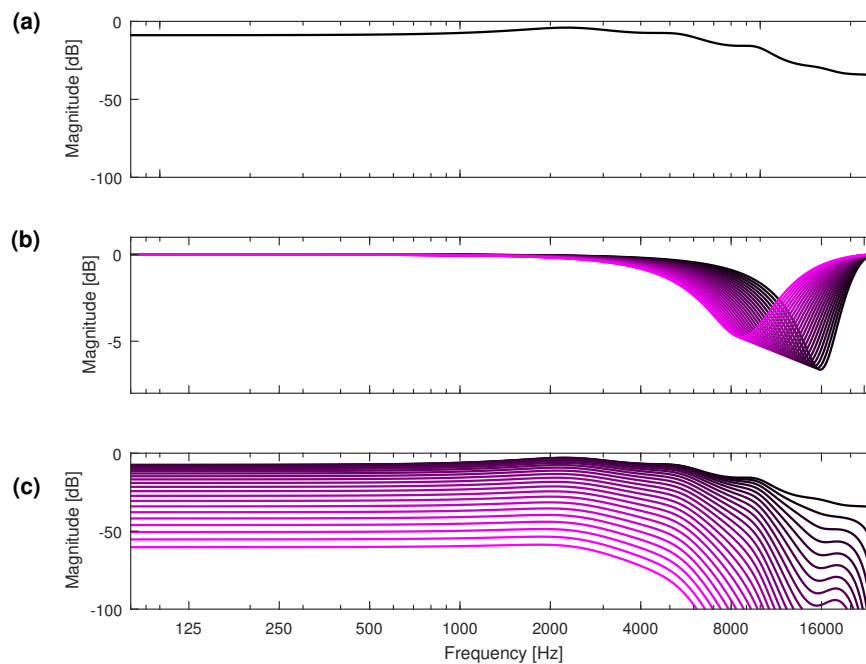


Figure 10. Magnitude responses of (a) the first-segment coloration filter $H_1(z)$; (b) differential filters; and (c) cascaded differential filters with the first-segment filter.

4.2. Revised Segmentation Method

The differential filtering technique was found to benefit more from a different segmentation method than what was used in the basic FVN method. The main idea here is to start a new segment when the difference in the spectrum from the start of the previous segment becomes sufficiently large. The RIR was analyzed in short windows (2048 samples) using low-order linear prediction (order 6 was used). Based on the magnitude responses of the corresponding all-pole filters, which provide an approximation of the spectral envelope of the windowed signals, a bandwidth for each segment was estimated. The bandwidth estimate was determined as the frequency at which the spectral envelope estimate decreased 20 dB from its maximum.

Using a linearly decaying threshold function, the segment boundaries were chosen based on reaching a sufficiently large change in bandwidth in the estimated spectral envelope. Therefore, a larger difference is required at the beginning than at the end of the RIR before starting a new segment. This led to the segmentation of the Pori RIR shown in Figure 11. The main difference compared to the previous method, shown in Figure 8, is that the revised segmentation reflects the significant changes in the magnitude response of the RIR.

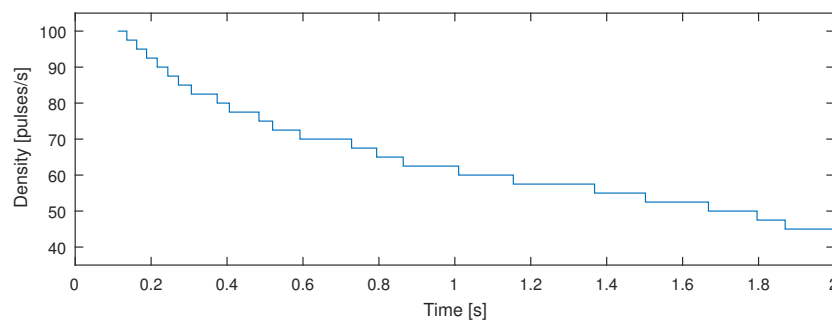


Figure 11. Segment lengths and density based on the revised segmentation strategy, which is used with differential coloration filters.

5. Design Examples

This section shows an example of modeling an RIR and modifying it. We show and analyze here the approximation of the Pori RIR implemented using the advanced method. An example of modeling this RIR using the basic FVN method has been presented earlier [16].

5.1. RIR Modeling Using Advanced FVN

The synthetic RIR produced using the advanced FVN model and its spectrogram are shown in Figure 12. As an objective comparison, Figure 13 shows the reverberation time T_{30} against octave bands for three RIRs (original, basic FVN, and advanced FVN). We decided to use T_{30} instead of T_{60} , because the signal-to-noise ratio near the end of the RIR does not sufficiently measure 60-dB decay; T_{30} is the measured time of a 30-dB decay multiplied by two.

All three RIRs in Figure 13 show the same tendency of lower reverberation time for higher frequencies than low frequencies. The octave-band reverberation times for the basic FVN algorithm stay within $\pm 7\%$ of the reference in all octave bands. For the second algorithm this spread is within $\pm 12\%$. The increased deviation is in accordance with the assumption that the second algorithm is a rougher approximation due to the lower coloration filter order.

An informal listening test comparing the two new reverb algorithms with a reference convolution reverb has been carried out using headphones. The reference RIR and its approximation with the basic FVN algorithm sound very similar even when comparing the impulse responses themselves. The approximation produced by the advanced FVN algorithm has a slightly more unnatural sound when listening to its impulse response. Results of a subjective test comparing the audio signal processed with the original RIRs and their FVN approximations are presented in Section 7 of this paper.

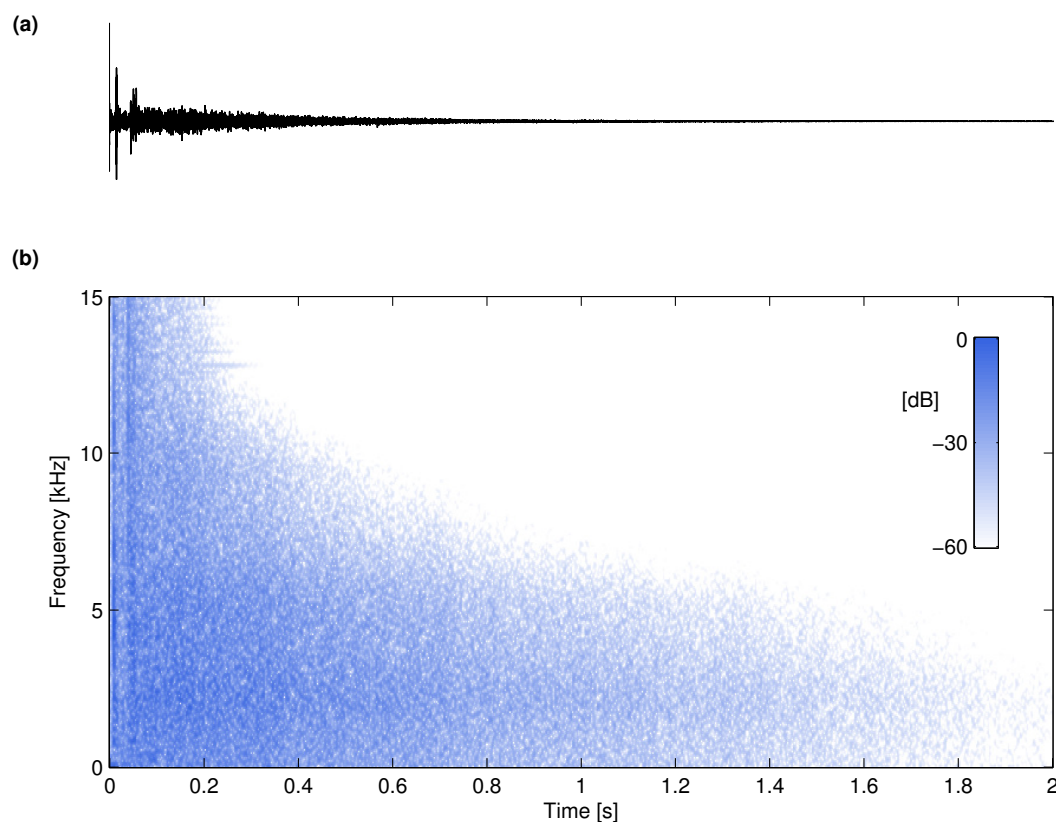


Figure 12. (a) Synthetic RIR produced using the advanced FVN method and (b) its spectrogram. Cf. Figure 4.

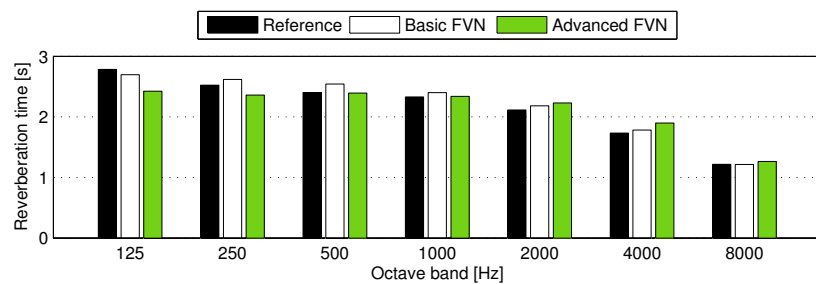


Figure 13. Reverberation time, T_{30} , for the original RIR (reference), its basic FVN synthesis, and advanced FVN synthesis.

5.2. Modification of the Approximated RIR

The parametric representation used in the FVN method allows modifying the modeled RIR in various ways. We have previously shown that the RIR can be dramatically shaped simply by modifying the gain term G_m [16]. In this way it is possible, for example, to increase or decrease the decay rate of the RIR. Here we show another option, time-stretching of the RIR.

Figure 14 shows the result of shortening the RIR by 50%. The number of segments, velvet noise density, coloration filters, or gains have not been changed, but the lengths of the VNC filters have been shortened to half. The early part of the RIR has not been modified, however. The overall shape of the RIR and the spectrogram are both seen to be preserved with respect to Figure 12, but the time scale has been modified. Another option to change the decay rate would be to modify the coloration filters and gains in the FVN model. Figure 15 shows an example in which the VNC filters have been lengthened by 100%, which leads to a twice-longer and, thus, more slowly decaying RIR. These examples demonstrate the possibilities for meaningful parametric modifications allowed by the FVN method.

The modeled impulse responses and test signals are available online at <http://research.spa.aalto.fi/publications/papers/applsci-fvn/>.

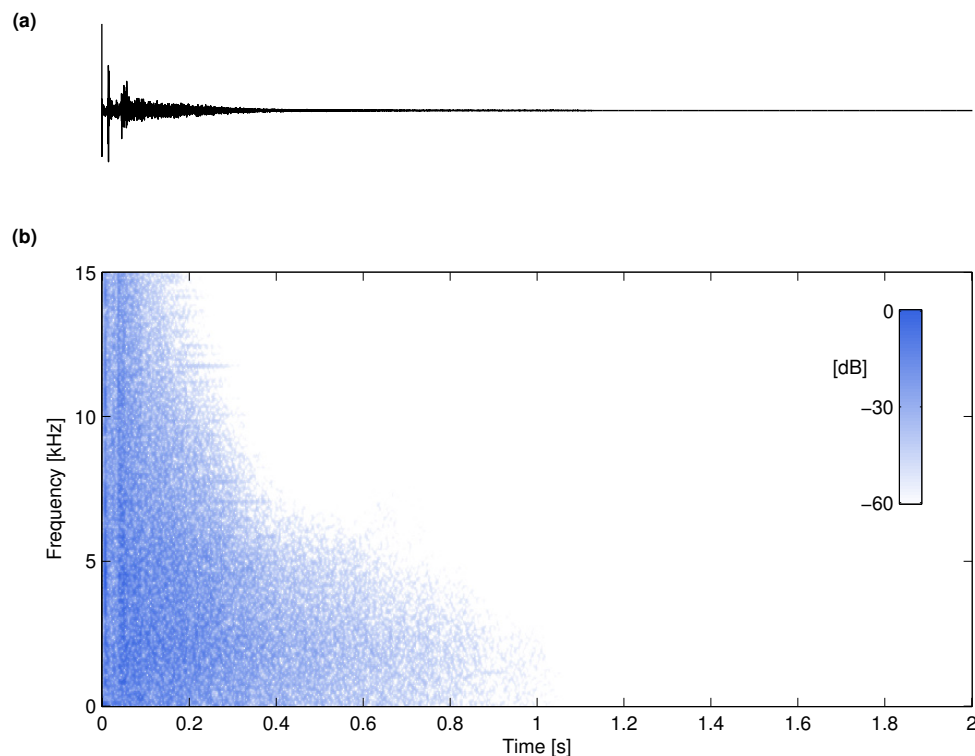


Figure 14. (a) 50% shortened synthetic RIR and (b) its spectrogram.

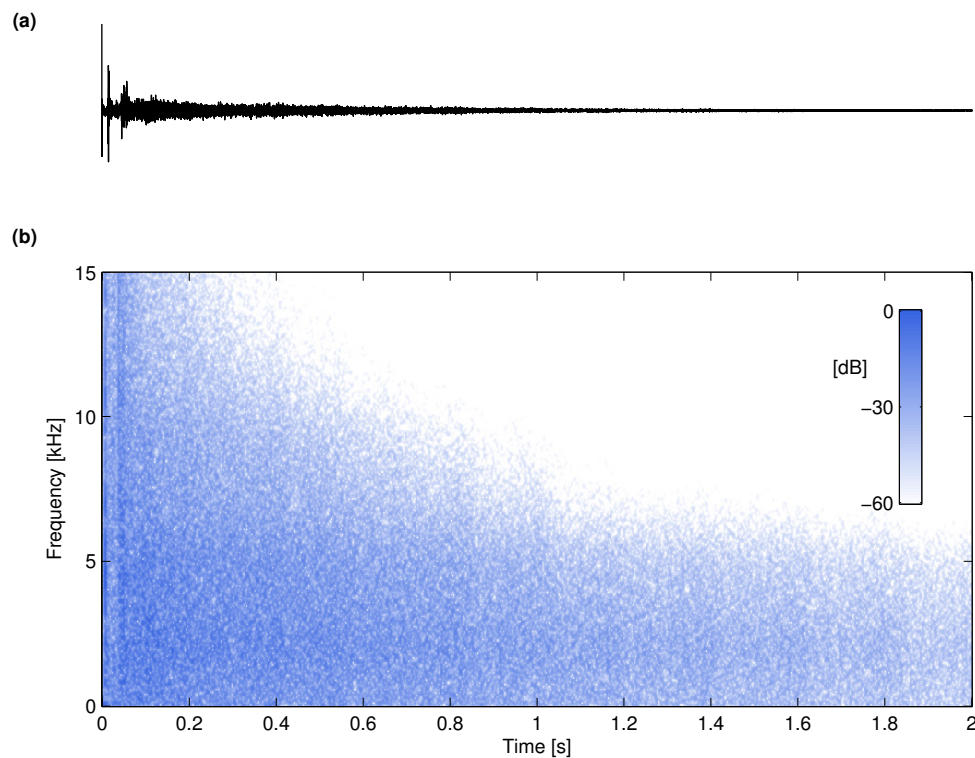


Figure 15. (a) 100% stretched RIR and (b) its spectrogram.

6. Computation and Memory Costs

The computational efficiency of reverberation algorithms is of great importance when they are used for real-time audio processing. Reverberation algorithms are also known to require a considerable amount of fast memory for storing past signal-sample values, which can be critical in implementations on limited hardware. Additionally, multichannel RIRs must be stored in spatial audio, which may require a considerable amount of memory storage. In this section, these implementation costs of the two versions of the FVN algorithm are compared with direct convolution and with partitioned fast convolution. The implementation cost of the early reflections is not included in the calculations, but it is assumed that the late part of the RIR lasts for 2 s.

6.1. Costs of the Basic FVN Algorithm

The number of floating-point operations (FLOPs) per processed sample required by the basic FVN algorithm are listed in Table 1. The numbers given are for the RIR modeling example of the Pori concert hall (see Figure 4). The FLOPs are specified as the number of additions and multiplications for each module of the algorithm. Note that the VNC filters only require additions and no multiplications. In Table 1, ‘H’ and ‘G’ are the coloration filters and gain adjustments, respectively, for each signal segment, and ‘Sum’ refers to the addition of output signals of the 20 branches before they are fed to the SAP filters (see Figure 6). In Table 1, note that the SAP filters only take 4% of total operations, but the coloration filters take 64%. This proves that efforts to reduce the cost of the coloration filtering stage are well motivated.

Table 1. Operations required to process one sample in each module of the basic FVN algorithm. The largest number in each column is in bold.

Module	Additions	Multiplications	Percentage
VNC	160	0	26%
H	200	200	64%
G	0	20	3%
Sum	19	0	3%
SAP	14	14	4%
Total	393	234	100%

6.2. Costs of the Advanced FVN Algorithm

Table 2 dissects the operations of each module in the advanced FVN, which uses the differential coloration filtering approach. Each differential coloration filter is implemented as a direct-form second-order IIR (infinite impulse response) filter, which requires five multiplications and four additions per sample. The VNC and SAP filters used for the two versions of the FVN algorithm are the same, and hence the same numbers of operations appear for these modules in Table 2 as in Table 1. The differential coloration filters ‘ ΔH ’ possess about half of the total arithmetic instructions, showing the advantage of collaborative cascaded filtering.

Table 2. Operations of the advanced FVN algorithm. Note that ΔH also includes the first coloration filter $H_1(z)$. The largest number in each column is in bold.

Module	Additions	Multiplications	Percentage
VNC	139	0	31%
ΔH	131	106	52%
ΔG	0	25	6%
Sum	24	0	5%
SAP	14	14	6%
Total	308	145	100%

6.3. Comparison Against Other Algorithms

Next, we compare the computational and memory costs of the proposed methods to other convolution reverberation approaches. We enumerate the number of FLOPs and the number of signal memory samples required for a 88,200 samples-long impulse response, as in the previous section.

The values listed in Table 3 for the direct convolution are based on the direct-form FIR implementation, which leads to the same number of multiplications as the number of RIR samples (88,200) and one less addition (88,199). In direct convolution, the required amount of fast memory is the same as the RIR length, since it defines the delay-line length (88,200 samples). The values for the partitioned fast convolution are taken from the recent improvement of the algorithm by Wefers and Vorländer (see Table 1 in [8]).

Table 3 shows that the proposed algorithms are over 100 times more efficient computationally than the direct convolution and approximately as efficient as the best partitioned fast convolution algorithm, which is only 12% more efficient than the advanced FVN. The memory consumption of the new method is the same as that of the direct convolution and 50% smaller than that of the partitioned convolution algorithm.

Table 3 also shows that the FVN method is useful for compression of RIR data: whereas the direct and partitioned convolution algorithms must store all RIR samples, the FVN methods only store two arrays of pointers, which give the locations of the positive and negative impulses, an array of segment lengths (20 in this case), plus 12 filter parameters per segment (a gain factor and 11 feedback coefficients of the 10th-order all-pole filter). The advanced FVN method is even more efficient in

this respect, as there are less impulses in the VCN block and the differential filters only require five parameters each. This yields a total of 294 parameters to be stored. The amount of data is only 0.33% compared to the original RIR samples. This implies that the FVN approach enables very efficient storage of multichannel RIR data.

Table 3. Operation count, fast memory and storage memory consumption of various reverberation algorithms for modeling a 2-s RIR at a 44.1-kHz sample rate. The smallest numbers are in bold. FLOPS: floating-point operations.

Algorithm	FLOPs	Delay-Line Memory	Storage Memory
Direct convolution	176,401	88,200	88,200
Partitioned fast convolution	399	176,400	88,200
Basic FVN	627	90,442	420
Advanced FVN	453	88,200	294

7. Subjective Evaluation

The proposed advanced FVN method was evaluated using a subjective test. Three different concert halls impulse responses were approximated from pre-recorded RIR [26]. One of the RIRs was the Pori concert hall response used in the examples above, which has a reverberation time of 2.3 s at middle frequencies. The second hall was the Cologne Philharmonie, which has a shorter mid-frequency reverb time (1.9 s). Its RIR is quite dry, containing mainly the direct sound, a few reflections, and a relatively short reverberation tail. The third hall was the Vienna Musikverein, which has the longest reverberation time (3.2 s) of the selected halls. Its RIR sounds very reverberant, having a lot of early reflections soon after the direct sound.

The beginning of each RIR approximation was taken from the measured RIR. Thus, the early-reflection part of the impulse responses remained the same as the original, and only the tail of the RIR was modified by the basic and advanced FVN approximations. The duration of each early-reflection segment was adjusted manually based on preliminary testing as follows: 110 ms for the Pori Concert Hall, 119 ms for the Cologne Philharmonie, and 52 ms for the Vienna Musikverein.

Three different sound files were processed with the three reference (original) RIRs and their approximations produced using the advanced FVN method, which yielded altogether 18 (3×6) sound files. The three test sounds contained drumming, slowly changing chords played with a synthesizer, and a cappella singing (the first 10 s of “Tom’s Diner” by Suzanne Vega).

The test type was ABX [27], which refers to a pair-wise test in which the subject always compares three sound files, A, B, and X, and is asked to identify whether sound X is the same as A or B. Additionally, in our test, the subjects had to evaluate the perceived difference between A and B on a five-point scale, a variant of the mean-opinion score. Figure 16 shows the user interface used in the listening test. The 18 test sounds were played in pseudo-random order, and they all appeared twice during the test, leading to 36 cases to be evaluated. Additionally, four extra cases were played in the beginning of the test, the answers of which were deleted from the data, since learning was assumed to occur during the first few cases, and only after this are the persons able to carefully and objectively evaluate the sounds. Thus, the total number of cases presented to the subjects in the listening test was 40.

Twelve subjects with no reported hearing problems participated in the listening test. Their age varied between 23 and 41 years. All subjects had previously participated in listening tests. It took typically 30 to 40 min for the subjects to finalize the test. The test can be assumed not to have been too difficult or tiresome.

Table 4 summarizes how the test subjects identified the synthetic reverberation from the original for different sound types. Since the subjects were allowed to listen to all sounds several times, detecting even the smallest differences turned out to be easy. Thus, in 86% of all cases, the persons identified the approximated RIR from the original. Detecting the difference in drumming, which contains transients,

was the easiest, and the identification score was 99%. Chords were the most difficult case, as the sounds are mostly stationary and the synthetic sounds had a slow attack. The difference was still detected in about three cases out of four. The difficulty in detecting the differences in singing was between the two extreme cases, and the recognition was successful in 84% of cases. After the test, the test subjects commented that it was fairly easy to find the different items in drum samples, but for the other two sounds it felt more difficult. However, the average rating for the differences was 3.1, which corresponds to a “small difference”. This implies that although it was often possible to discriminate between the original RIR and its approximation, the perceived difference was not considered to be very large.

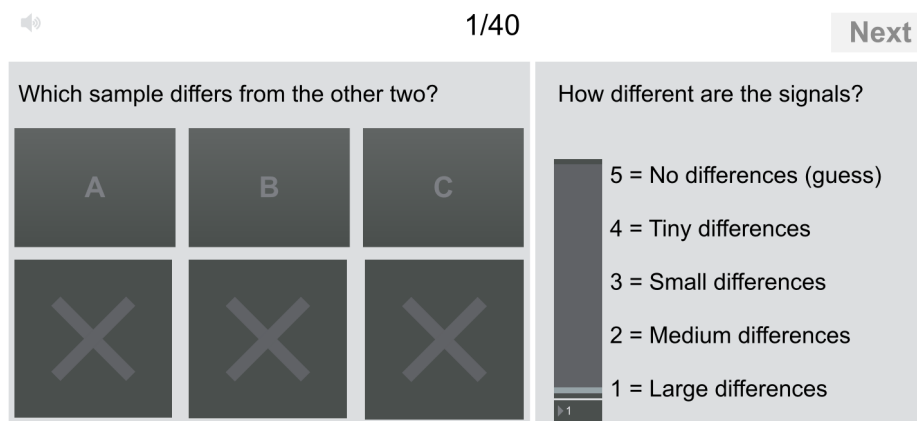


Figure 16. User interface of the ABX test with the 5-point difference rating used in the listening test. The verbal descriptions associated with each quality level appear on the right.

Table 4. Identification of FVN approximation of reverberated sounds in the listening test.

Sound Type	Drums	Chords	Singing	Average
Identification	99%	76%	84%	86%
Quality rating	2.1	3.9	3.3	3.1

Table 5 shows the listening test results for the three different halls. Interestingly, there was no significant difference between the different RIR types, but the identification of all approximations was close to the average, or 86%. The quality rating was, similarly, close to the average for all concert halls. Thus, the FVN method appears to be equally well suited to both short and long RIRs.

Table 5. Identification of the FVN approximation of different RIRs in the listening test.

Concert Hall	Pori	Cologne Philharmonie	Vienna Musikverein	Average
Identification	84%	85%	88%	86%
Quality rating	3.3	3.0	3.0	3.1

8. Conclusion and Future Prospects

This paper discussed the modeling of the late part of a measured room impulse response using filtered velvet-noise sequences. The idea here is to divide the impulse response into many non-overlapping segments of variable length and to approximate each segment using a spectral coloration filter and a sparse FIR filter having its coefficients taken from a velvet-noise sequence. The summed output of these filtering stages is further processed with a cascade of a few Schroeder allpass filters to increase the density and to smooth out the transitions between the segments. In this configuration, velvet-noise convolution can provide a smooth response even with very low pulse

densities. Moreover, the sparsity of the velvet noise may vary along the reverberation tail so that towards the end, where the bandwidth gets narrow, the sequences are sparser. To obtain a realistic model of a target RIR, the coloration filters can be designed by applying linear prediction to the variable-length RIR segments.

Additionally, this paper contributed a method to improve the computational efficiency of the FVN reverberation algorithm: the idea is to link the coloration filters so that each of them receives as the input the output of the previous stage. This way each segment only requires a differential coloration filter, which reduces the bandwidth sufficiently with respect to the previous stages. Instead of being a high-order IIR filter, each differential coloration filter is a second-order notch filter.

The performance of the proposed algorithm was demonstrated with a modeling example, and the results showed that the algorithm is able to accurately model the overall characteristics of the target concert hall impulse response. The design procedure yields a flexible parametric approximation of the late part of the target impulse response, allowing for variations such as time-scale modification. Furthermore, the proposed reverberation algorithm is computationally efficient, providing a major advantage over the direct convolution: in the example case of 2-s RIR modeling, the proposed method reduces the computational cost by over 99.6% compared to direct convolution, and it is in this respect comparable to the best FFT-based partitioned convolution methods.

Results of a subjective test were also reported, showing that the FVN approximations are often perceptually different from the original, but that the difference between the original RIR and its FVN approximation is considered small. The difference is easiest to observe when the audio signal contains transients, such as in drum sounds. However, the FVN method was observed to be equally well-suited for approximating long and short RIRs, as there was not much difference in the identification of different RIRs.

The proposed method can be used to implement convolution reverberation in which instead of directly using the measured impulse response, its FVN model is implemented. This allows the possibility for parametric control of the impulse response characteristics. The proposed FVN method has a computational complexity that is comparable to the partitioned fast convolution method, but with a far reduced memory storage, which is important in spatial audio, where multichannel sound reproduction requires a large set of multidirectional impulse responses.

Acknowledgments: This work has been funded in part by the Academy of Finland (ICHO project, Aalto University project No. 13296390). The authors would like to thank Prof. Tapio Lokki for his help in providing and selecting the room acoustic data used in this work.

Author Contributions: Vesa Välimäki wrote the paper and contributed to the original idea; Bo Holm-Rasmussen contributed to the original idea, programmed the methods, produced audio examples, and produced most figures; Benoit Alary contributed to the advanced method, produced figures 5, 10, 11, and 16, validated the method, and produced audio examples; Heidi-Maria Lehtonen contributed to the original idea, was the advisor of the work of Bo Holm-Rasmussen, and contributed to writing, when she was a postdoctoral researcher at Aalto University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schroeder, M.R.; Logan, B.F. Colorless artificial reverberation. *J. Audio Eng. Soc.* **1961**, *9*, 192–197.
2. Välimäki V.; Parker, J.D.; Savioja, L.; Smith, J.O.; Abel, J.S. Fifty years of artificial reverberation. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 1421–1448.
3. Gardner, W.G. Reverberation algorithms. In *Applications of Digital Signal Processing to Audio and Acoustics*; Kahrs, M., Brandenburg, K., Eds.; Kluwer: New York, NY, USA, 2002; pp. 85–131.
4. Välimäki, V.; Parker, J.D.; Savioja, L.; Smith, J.O.; Abel, J.S. More than 50 years of artificial reverberation. In *Proceedings of the Audio Engineering Society 60th International Conference on Dereverberation and Reverberation of Audio, Music, and Speech*, Leuven, Belgium, 3–5 February 2016.

5. Shelley, S.B.; Murphy, D.T.; Chadwick, A.J. B-format acoustic impulse response measurement and analysis in the forest at Koli national park, Finland. In Proceedings of the International Conference on Digital Audio Effects (DAFX), Maynooth, Ireland, 2–5 September 2013; pp. 351–355.
6. Kulp, B.D. Digital equalization using Fourier transform techniques. In Proceedings of the Audio Engineering Society 85th Convention, Los Angeles, CA, USA, 3–6 November 1988.
7. Gardner, W.G. Efficient convolution without input-output delay. *J. Audio Eng. Soc.* **1995**, *43*, 127–136.
8. Wefers, F.; Vorländer, M. Optimal filter partitions for non-uniformly partitioned convolution. In Proceedings of the Audio Engineering Society 45th International Conference on Applications of Time-Frequency Processing in Audio, Helsinki, Finland, 1–4 March 2012.
9. Wefers, F. Partitioned Convolution Algorithms for Real-Time Auralization. Ph.D. Thesis, RWTH Aachen University, Institute of Technical Acoustics, Aachen, Germany, 2014.
10. Moorer, J.A. About this reverberation business. *Comput. Music J.* **1979**, *3*, 13–28.
11. Rubak, P.; Johansen, L.G. Artificial reverberation based on a pseudo-random impulse response. In Proceedings of the Audio Engineering Society 104th Convention, Amsterdam, The Netherlands, 16–19 May 1998.
12. Rubak, P.; Johansen, L.G. Artificial reverberation based on a pseudo-random impulse response II. In Proceedings of the Audio Engineering Society 106th Convention, Munich, Germany, 8–11 May 1999.
13. Karjalainen, M.; Järveläinen, H. Reverberation modeling using velvet noise. In Proceedings of the Audio Engineering Society 30th International Conference on Intelligent Audio Environments, Saariselkä, Finland, 15–17 March 2007.
14. Lee, K.S.; Abel, J.S.; Välimäki, V.; Stilson, T.; Berners, D.P. The switched convolution reverberator. *J. Audio Eng. Soc.* **2012**, *60*, 227–236.
15. Oksanen, S.; Parker, J.; Politis, A.; Välimäki, V. A directional diffuse reverberation model for excavated tunnels in rock. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 644–648.
16. Holm-Rasmussen, B.; Lehtonen, H.M.; Välimäki, V. A new reverberator based on variable sparsity convolution. In Proceedings of the International Conference on Digital Audio Effects (DAFX), Maynooth, Ireland, 2–5 September 2013; pp. 344–350.
17. Holm-Rasmussen, B. Velvet Noise in Reverberation Algorithms. MSc Thesis, Technical University of Denmark, Kgs. Lyngby, Denmark, October 2013.
18. Karjalainen, M.; Järveläinen, H. More about this reverberation science: Perceptually good late reverberation. In Proceedings of the Audio Engineering Society 111th Convention, New York, NY, USA, 30 November–3 December 2001.
19. Abel, J.S.; Coffin, S.A.; Spratt, K.S. A modal architecture for artificial reverberation with application to room acoustics modeling. In Proceedings of the Audio Engineering Society 137th Convention, Los Angeles, CA, USA, 9–12 October 2014.
20. Jot, J.M. An analysis/synthesis approach to real-time artificial reverberation. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), San Francisco, CA, USA, 23–26 March 1992; pp. 221–224.
21. Jot, J.M.; Chaigne, A. Digital delay networks for designing artificial reverberators. In Proceedings of the Audio Engineering Society 90th Convention, Paris, France, 19–22 February 1991.
22. Schlecht, S.J.; Habets, E.A.P. Feedback delay networks: Echo density and mixing time. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 374–383.
23. Välimäki, V.; Lehtonen, H.M.; Takanen, M. A perceptual study on velvet noise and its variants at different pulse densities. *IEEE Trans. Audio Speech Lang. Process.* **2013**, *21*, 1481–1488.
24. Makhoul, J. Linear prediction: A tutorial review. *Proc. IEEE* **1975**, *63*, 561–580.
25. Dutilleul, P.; Holters, M.; Disch, S.; Zölzer, U. Filters and delays. In *DAFX: Digital Audio Effects*, 2nd ed.; Zölzer, U., Ed.; Wiley: Hoboken, NJ, USA, 2011; pp. 47–81.
26. Pätynen, J. A Virtual Symphony Orchestra for Studies on Concert Hall Acoustics. Ph.D. Thesis, Aalto University, Espoo, Finland, November 2011.
27. Clark, D. High-resolution subjective testing using a double-blind comparator. *J. Audio Eng. Soc.* **1982**, *30*, 330–338.



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).