PARAMETER ANALYSIS FOR GLZ AUDIO COMPRESSION

Zeph Landau[†] *and Darko Kirovski*[‡]

[†] Department of Mathematics, The City College of New York, NY, USA
[‡] Microsoft Research, One Microsoft Way, Redmond, WA, USA

ABSTRACT

The generalized Lempel-Ziv (GLZ) paradigm for lossy compression for audio relies upon the fact that music, in particular electronically generated sound, has substantial level of repetitiveness within a single clip. Thus, GLZ compresses each of the overlapped and transformed windows of audio using a linear combination of filtered past windows. Following the introduction of the basic GLZ algorithm, in this paper, we empirically analyze several key algorithm components. We analyze the design of simple band-pass filters used during the similarity search, we investigate the distributions of weights used to create the linear combinations, and finally, we explore how beat detection can be used to significantly speed up the similarity search process. We present preliminary experimental results on a benchmark of electronically generated musical pieces.

1. THE BASIC GLZ PARADIGM

Repetition is often a principal part of composing music and is a natural consequence of the fact that distinct instruments, voices and tones are used to create a soundtrack. The GLZ compression paradigm is based upon a generalization of predictive coding [3] and the Lempel-Ziv compression algorithm [4]; it explores music self-similarity to enable a memory based compression model. In this section, we overview the GLZ paradigm [1]. We consider a normalized input signal $\mathbf{x}$ of N samples, where each sample $x_i \subset \mathbf{x}$ is normalized $x_i \in [-1, 1]$. The signal is partitioned into 50% overlapping blocks of n samples where n is a power of two, commonly within $n \in [512, 4096]$. The proposed compressor windows each signal block using a perfect reconstruction analysis window and individually compresses it. During decompression, a synthesis window reverses the effect of the analysis window [2].

For a given signal block $\mathbf{x}_i = \{x_i, \ldots, x_{i+n-1}\}$, we establish a search window $\mathbf{s}_i = \{x_{\text{start}(i)}, \ldots, x_{i-1}\}$ which either represents the full signal history (i.e., $\text{start}(i) = 1$) or has a certain predetermined length S (i.e., $i - \text{start}(i) = S$). We search for a subset of blocks $W = \{\mathbf{w}_j \subset \mathbf{s}_i, j = 1 \ldots K\}$ and a set of corresponding scalars $A = \{\alpha_j \in \mathbb{A} \subset \mathbb{R}, j = 1 \ldots K\}$ and transforms $F = \{f_j(\mathbb{R}^n) \rightarrow \mathbb{R}^n, f_j \subset \mathbb{F}, j = 1 \ldots K\}$, where $\mathbb{F}$ is the set of all considered transforms. Sets F, A, and W satisfy the following optimization goal:

$$\arg \min_{W,A,F} H \left\{ m \left[\mathbf{x}_i, \sum_{j=1}^{K} \alpha_j f_j(\mathbf{w}_j), \mathbf{b}_i \right] \right\}. \quad (1)$$

The set of transforms $\mathbb{F}$ consists of a time-to-frequency transform such as the MLT [2] with different band-pass filters. For example, an exemplary transform $f_j(\mathbf{w}_j)$ would apply the MLT to the block $\mathbf{w}_j$ and zero out the resulting frequency coefficients below 100Hz.

The representation error $\varepsilon = \mathbf{x}_i - \sum_{j=1}^{K} \alpha_j f_j(\mathbf{w}_j) = \mathbf{x}_i - \mathbf{r}_i$ is masked using a psycho-acoustic filter $m()$ as follows. We first compute the psycho-acoustic mask $\mathbf{b}_i$ of the source block $\mathbf{x}_i$. The mask $\mathbf{b}_i \in \{0, 1\}^n$ distinguishes audible from inaudible frequency coefficients and can be computed using well known psycho-acoustic models [5]. We define the masking function $m(x, r, b)$ on a single signal coefficient x and its reconstruction r and masking bit b:

$$m(x, r, b) \equiv \begin{cases} x - r, & b = 1 \\ 0, & b = 0 \wedge |r| \leq T \\ r - T \cdot \text{sign}(r), & b = 0 \wedge |r| > T \end{cases} \quad (2)$$

where T denotes the hearing threshold for sample x. When applied to vectors, the function $m(\mathbf{x}, \mathbf{r}, \mathbf{b})$ independently performs the steps from Eqn. 2 to each vector element. The goal of the masking function $m()$ is to set the error such that reconstruction of audible samples is exact whereas the reconstruction of inaudible samples is such that the absolute magnitude of the error is minimized.

Finally, function $H()$ computes the entropy of the following information: **[H1.]** quantized pointers to all blocks in W, **[H2.]** quantized pointers to the applied transforms F, **[H3.]** quantized scalars in A used to create the linear combination of transformed blocks, and **[H4.]** the error vector returned by function $m()$.

The encoded information (H1–H4) represents the final compressed stream. Clearly, the optimization goal of the similarity search formalized using Eqn. 1, is to find a set W of K blocks which occur prior to $\mathbf{x}_i$ and a linear combination of their transforms $\mathbf{r}_i$, which represents $\mathbf{x}_i$ as close as possible in the sense of minimizing the entropy of the remaining error vector $m(\mathbf{x}_i, \mathbf{r}_i, \mathbf{b}_i)$.

The decompression process is marginally slower than in the case of a traditional codec. For each reconstructed block $\mathbf{x}_i$, we first decode the information from the compressed stream; we extract the set of pointers to prior blocks W, the scalar amplifiers A, the filters F applied onto each block pointed by W to compute the basis vectors, and the error vector $\varepsilon_i = m(\mathbf{x}_i, \mathbf{r}_i, \mathbf{b}_i)$. In the last step, we compute the reconstruction as $\mathbf{x}_i = \sum_{j=1}^{K} \alpha_j f_j(\mathbf{w}_j) + \varepsilon_i$. Since the basis can be constructed from the source in the time- or frequency-domain, during decompression the reconstruction window which equals the search window is maintained in the appropriate domain.

1.1. Compression Parameters

Choosing the parameters to improve the compression algorithm is a delicate task. Specifically, choices that would reduce the entropy contribution of (H4), such as increasing the sizes of $\mathbb{A}$, $\mathbb{F}$, K, or decreasing the blocksize n, have the effect of increasing the entropy contribution of (H1-H3). Conversely, decreasing the size of $\mathbb{A}$, $\mathbb{F}$, K, or increasing the blocksize n reduces the entropy contribution of (H1-H3) but has the effect of giving a worse approximation of the signal which increases the entropy contribution of (H4). Independent of these considerations are the very important choices of the elements of $\mathbb{F}$ and $\mathbb{A}$.

We limit $K \leq 3$. We fixed blocksize $n = 1024$ though in the general model, n could be variable. We set $\mathbb{A} = \{-1, -.1, .1, .2, .5, .8, .9, 1, 1.1, 1.2\}$ and impose $\mathbb{F} = \{f_1..f_5\}$ as follows: f_1, f_2, f_3 – low-(0,64), mid-(65,167), and high-(168-512)-pass filters (respectively) of the MLT coefficients, f_4 – a low-stop filter that removes the lowest eighth of all n MLT coefficients, and f_5 - an all-pass filter. In our experiments, we used a linear quantizer and an entropy coder with a unified view on all quantized MLT coefficients to digitize the compression error ε_i. Just as in classic compression schemes, using a particular quantizer-encoder system may alter system performance significantly.

1.2. Similarity Search

Searching for similarity based on the L2 (Euclidean) metric is fast. Based upon this effect, the following heuristic can be proposed [1]. We begin by choosing a small set $\mathbb{G} = \{g_l : 1 \leq l \leq L\}$ of linear operators $g_l : \mathbb{R}^n \rightarrow \mathbb{R}^n$, for which we heuristically feel that L2 similarity for $g_l(\mathbf{x})$ and $g_l(\mathbf{w})$ leads to small values for $H\{m(\mathbf{x}, \alpha f(\mathbf{w}), \mathbf{b}_i)\}$ for at least one choice of $\alpha \in \mathbb{A}$, $f \in \mathbb{F}$. Then our search procedure, iteratively performs the following steps. We begin by forming a pool $\mathbb{P}$ of candidate blocks by, for each l, choosing the blocks $\mathbf{s}_{i,j}$ within the search window $\mathbf{s}_i$ of length n for which $g_i(\mathbf{s}_{i,j})$ is maximally correlated in L2 norm to $g_l(\mathbf{x}_i)$. The top P correlated blocks from $\mathbf{s}_i$ for each l are denoted as the most similar ones to $\mathbf{x}_i$ and put into the pool $\mathbb{P}$. Next, each block $\mathbf{s}_{i,j}$ from $\mathbb{P}$ is evaluated as a solution by searching for the minimal $H\{m(\mathbf{x}_i, \alpha f(\mathbf{s}_{i,j}), \mathbf{b}_i)\}$ over the entire search spaces $\alpha \in \mathbb{A}$ and $f \in \mathbb{F}$.

The result of this best-effort search is adopted as the first basis vector $\mathbf{w}_1$ in the set W and its accompanying parameters $f_1()$ and α_1. Next, the most similar point $\mathbf{w}_1$ is subtracted from $\mathbf{x}_i$ as $\mathbf{x}_i - \alpha_1 f_1(\mathbf{w}_1)$. The two steps, search for the best approximation of the signal remainder and its subtraction, are iterated while the improvement in the entropy of the remaining signal is greater than the number of bits required to encode α_K, $f_K()$, and the pointer to $\mathbf{w}_K$.

1.3. Similarity Search Parameters

For the experimental results presented here $\mathbb{G} = \{g_l : 1 \leq l \leq 7\}$ with: g_1, g_2, g_3 – low-, mid-, and high-pass filtering of the bands in the MLT domain responsible for about one third of the entropy of music, e.g., for a block of $n = 512$ MLT coefficients the pass frequencies were set at coefficients 64 and 167, g_4 – a low-stop filter that removes the lowest eighth of all n MLT coefficients, g_5 – a weighting filter that weights each coefficient with the ratio of the entropy of this coefficient over the clip to the average absolute value of this coefficient, g_6 a filter in the time domain corresponding to the window function of the MLT, and g_7 which did not filter at all.

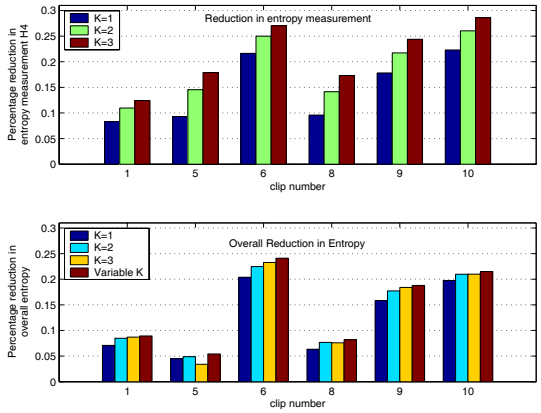

Fig. 1. Entropy reduction performance.

2. EMPIRICAL PARAMETER ANALYSIS

In the experiments presented, we considered ten audio clips, mostly electronically generated. Then benchmark is described in detail in [1]. For each clip, we randomly selected 100 blocks and ran our compression scheme with the parameters K, n, $\mathbb{A}$, $\mathbb{F}$, $\mathbb{G}$ as described in the previous section. For each g_l in the similarity search, we used the top 200 correlated blocks ($P = 200$) to use in $\mathbb{P}$.

The resulting compression in entropy is shown in Figure 1 for clips **1,5,6,8,9,10**. The remaining clips **2,3,4,7** showed little compression gain. The top graph shows the reduction in the entropy of the error signal – the (H4) contribution to the total entropy measurement – after one, two

and three rounds. The lower graph factors in the cost of (H1-H3), which is approximately 26.5 bits per round per block (about 21 bits for the pointer to the similar block, and 5.5 bits for the specification of α and f_i). One can consider a variable scheme for choosing K as follows: for each block, factor in the bit cost for each additional round, i.e., if the resulting compression in entropy does not exceed (26.5) then stop. The first three bars for each clip on the bottom graph correspond to the total entropy compression for one, two and three rounds; the final bar corresponds to the total entropy using the variable scheme for choosing K. This scheme produces a total entropy reduction of 8.9%, 5.4%, 24.1%, 8.1%, 18.7% and 21.5% respectively for the clips **1,5,6,8,9,10**. Note that the improvement slows rapidly, round by round, and thus it appears that increasing the size of K should not produce much improvement.

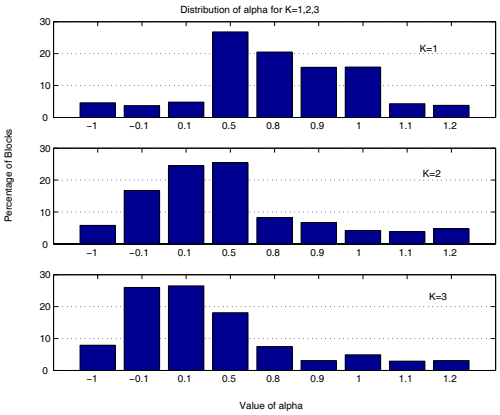

Fig. 2. Distribution of α. The above three histograms represent the distribution per round of the scaling parameter over the chosen set $\mathbb{A} = \{-1, -.1, .1, .2, .5, .8, .9, 1, 1.1, 1.2\}$.

The distribution of α. Figure 2 shows the histogram of the choice of α used round by round. These results bare out the intuition that the average value of α should decrease round by round. The fact that 28% of the first round choices of α were equal to $1/2$ suggests that the set $\mathbb{A}$ should have included more numbers near $1/2$. Based on these observations, a clear improvement would be to have round dependent sets $\mathbb{A}_i$, $i = 1, 2, 3$, with $\mathbb{A}_1$ concentrated mostly in the range $\alpha \in [.4, 1]$, $\mathbb{A}_2$ concentrated mostly in a lower range including some small negative values and $\mathbb{A}_3$ concentrated in a still lower range.

Alternatively we suggest two heuristics to find the best α parameter associated to a given block $\mathbf{x}_i$ and f_1: [(*i*)] $\alpha_1 = \mathbf{x}_i \cdot f_1(\mathbf{s}_{i,j})/(||\mathbf{x}_i|| \cdot ||f_1(\mathbf{s}_{i,j})||)$, which aims at minimizing the Euclidean distance between $\mathbf{x}_i$ and its basis vector $f_1(\mathbf{s}_{i,j})$, and [(*ii*)] two-step search for the best α. In the first step, we search within a particular set of common values for α. We denote this intermediate search result as α^*. In the second step, we fine-tune α^* by searching in its nearest locality, e.g., we exhaustively search for the next best decimal to obtain the final α.

The choice of $\mathbb{F}$. Figure 3 describes the performance of the filters $f \in \mathbb{F}$. The left of the pair of bars gives the percentage of time that the use of filter f_i produces a reduction in entropy within $.5\%$ of the best result found. We see that for each round, f_5, the identity filter produces a result within $.5\%$ of the best result the most often. This points to the inclusion of f_5 in a future choice of $\mathbb{F}$. The difference in the graphs round by round suggests that a round dependent set $\mathbb{F}_i$, $i = 1, 2, 3$ might improve results. The right hand bar of each pair represents the percentage of time that the filter produces a reduction in entropy within $.5\%$ of the best and that the best filter, f_5, does not give a result within $.5\%$ of the best result. This right hand bar gives some indication of other important filters that might be chosen to complement f_5. As we see, in the second two rounds, the medium pass filter stands out, which suggests that the choice of f_2 or some other medium pass filter would be a good complement to include with f_5 in a round dependent $\mathbb{F}_2$ or $\mathbb{F}_3$.

The choice of search filters $\mathbb{G}$. Figure 4 describes the performance of the search filters $g \in \mathbb{G}$ for each round. The left bar of the pair indexed by i gives the percentage of time that a similar block generated from g_i had an entropy compression within .5% of the best compression that was found. We see that with this set up, filter g_5, the filter with each MLT coefficient weighted by the ratio of the entropy of that coefficient for the clip to the average absolute value of that coefficient, outperformed the other filters, finding a candidate within .5% of the best candidate about half the time. For the second two rounds, the mid-pass filter g_2 outperformed the other filters, finding the best candidate 40% and 43% percent of the time respectively. The right bar of the pair considers the effect of the remaining filters on those blocks where the top performing filter, g_5 for $K = 1$, g_2 for $K = 2, 3$, did not produce a candidate block with compression within .5% of the best compression that was found. For instance in the case of $K = 1$, the left hand bars indicate that g_6 and g_7 might be considered for removal since they do not perform well as often as g_5 and tend to mostly perform well on blocks that g_5 also has performed well on.

The fact that there are better filters than g_6 and g_7 on all rounds emphasizes that there is a significant difference between small entropy and small Euclidean distance. These results point towards making the set $\mathbb{G}$ round dependent, as well as suggesting that careful choices of its elements may produce significant improvements. The choice of g_5 is the only choice that tries to take into account the difference between similarity in the L2 sense and the entropy sense.

2.1. Using the Beat Information

One of the most robust events in music is its beat. In this section, we observe that similarity in music occurs corre-

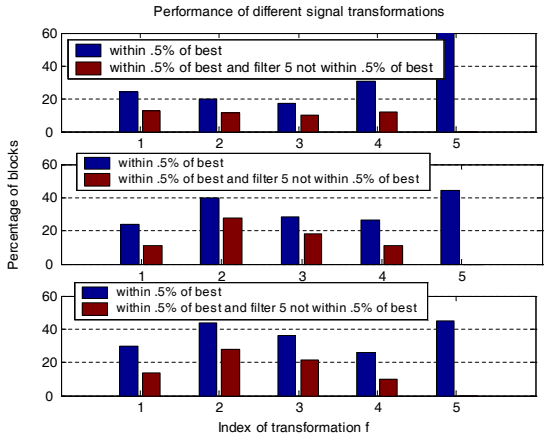

Fig. 3. Performance of f_i.

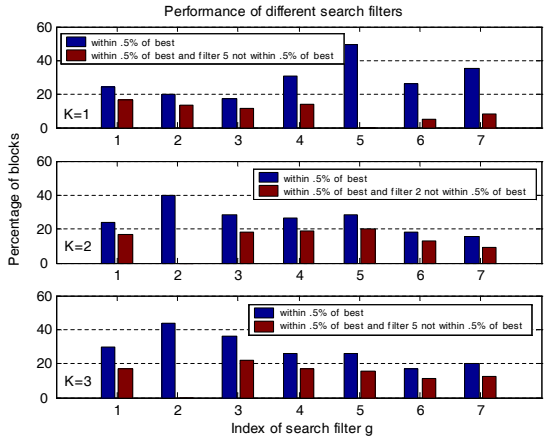

Fig. 4. Performance of g_i.

lated with its rhythmic behavior. In order to analyze this phenomenon, we sequenced music clips into beats using a fast, off-the-shelf beat detection system based on the EM algorithm [6]. We denote the beat information as a pseudo-periodic sequence $T = \{t_0 = 1 \cup t_j, j = 1 \ldots |T|-1\}$. Any two consecutive pointers in T are such that $|t_j - t_{j-1} - \tau| \leq \epsilon, j > 1$, where τ is the beat period and ϵ is its maximum deviation.

We derive the main empirical observation using the following experiment. After computing T for each clip in our audio benchmark, we computed the similarities for 100 1024-long MLT blocks randomly sampled from the first 45 seconds of the clip. For each sampled block $\mathbf{x}_i$ and each considered filter $f_j() \in \mathbb{F}$, we identified a block candidate pool: 10 blocks that correlated the best with $\mathbf{x}_i$ in the $f_j()$-domain. The goal was to compute the relative position offset $\rho(\mathbf{x}_i, \mathbf{x}_m)$ for a block $\mathbf{x}_m$ in the candidate pool with respect to $\mathbf{x}_i$ and within their respective containing beats. For each block $\mathbf{x}_i$ and its candidate block $\mathbf{x}_m$ starting at the i-th and m-th sample of $\mathbf{x}$ respectively, we denote the borders of

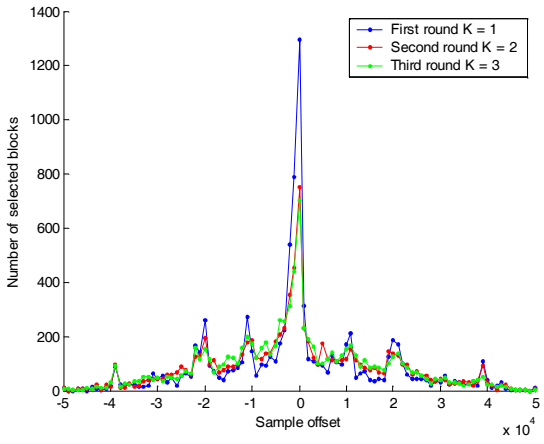

Fig. 5. Distribution of similar blocks with respect to music's beat.

their containing beats as t_i, t_{i+1} and t_m, t_{m+1} respectively. We set $a = (i - t_i) - (m - t_m)$, $b = (i - t_{i+1}) - (m - t_m)$, and $c = (i - t_i) - (m - t_{m+1})$. Now, we define:

$$\rho(\mathbf{x}_i, \mathbf{x}_m) \equiv \begin{cases} a, & |a| = \min(|a|, |b|, |c|) \\ b, & |b| = \min(|a|, |b|, |c|) \\ c, & |c| = \min(|a|, |b|, |c|) \end{cases} \quad (3)$$

In all cases, large number of selected blocks were found in a relatively narrow co-located region within respective beats. Within a region of size equal to 1/8th of the beat length, typically more than half the selected blocks are found. This points to to improvements that can be enabled using beat detection. First, the similarity search can be significantly sped up (factor 2-5x) by restricting the search space to the narrow regions co-located with the target block. Second, such a restriction reduces the pointer-set W entropy, about 2.5-3 bits lower than equiprobable pointer encoding.

3. REFERENCES

[1] D. Kirovski and Z. Landau. Generalized Lempel-Ziv Compression of Audio. IEEE *International Workshop on Multimedia and Signal Processing*, to appear, 2004.

[2] H. Malvar. A modulated complex lapped transform and its application to audio processing. IEEE *International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[3] B. Derjavitch, E.M. Deloraine, and S. Van Mierlo. French Patent No. 932140, August 1946; also filed as U.S. Patent No. 2629859, October 1947.

[4] J. Ziv and A. Lempel. A Universal Algorithm for Sequential Data Compression. IEEE *Transactions on Information Theory*, Vol.23, no.3, pp.337–343, 1977.

[5] K. Brandenburg. Perceptual coding of high quality digital audio. *Applications of Digital Signal Processing to Audio and Acoustics*, Kluwer, 1998.

[6] D. Kirovski and H. Attias. Audio Watermark Robustness to Desynchronization via Beat Detection. *Info Hiding Workshop*, 2002.