Optimal Data Shaping Code Design

Yi Liu, Pengfei Huang, Alexander W. Bergman and Paul H. Siegel

Center for Memory and Recording Research, UC San Diego
Outline

1. Introduction

2. Type-I and Type-II Minimization

3. Encoder Design

4. Experiment Results on MLC Shaping Codes

5. Conclusion
Introduction

- Flash memory: the most widely used non-volatile memory
  - fast read/write speed
  - low power consumption
- Flash memory cells gradually wear out during program-erase (P/E) cycling.
- Damage from programming the cell depends on the cell level. Programming a cell to higher level induces more damage.
- Enhancing lifetime by using shaping codes
  - Endurance code\(^1\): shapes random (unstructured) data with a given rate
  - Direct shaping code\(^2,3\): shapes structured data with rate 1

---


\(^2\) E. Sharon, et al., Data Shaping for Improving Endurance and Reliability in Sub-20nm NAND, presented at Flash Memory Summit, Santa Clara, CA, August 4-7, 2014.

Definition of General Shaping Codes

Definition

Let $\mathbf{X} = X_1X_2 \ldots$ be an i.i.d source with alphabet $\mathcal{X} = \{\alpha_1, \ldots, \alpha_u\}$. The distribution of $\mathbf{X}$ will be denoted by $P (P_1 \geq P_2 \geq \ldots P_u)$. 

Example

Input: $\mathbf{X} = \{0, 1\}$, $\mathbf{X} \sim \text{Ber}(\frac{1}{2})$

Output: $\mathbf{Y} = \{0, 1\}$, $U_0 = 0.585$ and $U_1 = 1.585$

Shaping code defined by mapping $\{11 \rightarrow 111, 10 \rightarrow 110, 01 \rightarrow 10, 00 \rightarrow 0\}$. 

Liu, Huang, Bergman, Siegel (CMRR)
Definition of General Shaping Codes

Definition

- Let $X = X_1 X_2 \ldots$ be an i.i.d source with alphabet $\mathcal{X} = \{\alpha_1, \ldots, \alpha_u\}$. The distribution of $X$ will be denoted by $P (P_1 \geq P_2 \geq \ldots P_u)$.
- Let $\mathcal{Y} = \{\beta_1, \ldots, \beta_v\}$ be an alphabet and $\mathcal{Y}^*$ the set of all finite sequences over $\mathcal{Y}$, including the null string $\lambda$ of length 0. Every $\beta_i$ corresponds to a cost $U_i$ ($U_1 \leq U_2 \leq \ldots \leq U_v$).
Definition of General Shaping Codes

Definition

- Let $X = X_1X_2\ldots$ be an i.i.d source with alphabet $\mathcal{X} = \{\alpha_1, \ldots, \alpha_u\}$. The distribution of $X$ will be denoted by $P (P_1 \geq P_2 \geq \ldots \geq P_u)$. 
- Let $\mathcal{Y} = \{\beta_1, \ldots, \beta_v\}$ be an alphabet and $\mathcal{Y}^*$ the set of all finite sequences over $\mathcal{Y}$, including the null string $\lambda$ of length 0. Every $\beta_i$ corresponds to a cost $U_i$ ($U_1 \leq U_2 \leq \ldots \leq U_v$).

A shaping code is defined as a prefix-free mapping $\phi : \mathcal{X}^q \rightarrow \mathcal{Y}^*$ which maps $x_1^q$ to a variable length sequence $y^*$.

Example

- Input: $\mathcal{X} = \{0, 1\}$, $X \sim Ber(\frac{1}{2})$
- Output: $\mathcal{Y} = \{0, 1\}$, $U_0 = 0.585$ and $U_1 = 1.585$
- Shaping code defined by mapping $\{11 \rightarrow 111, 10 \rightarrow 110, 01 \rightarrow 10, 00 \rightarrow 0\}$. 
Expansion Factor

Definition

- The expected length of a codeword is

\[ E(L) = \sum_{x^q_1 \in X^q} P(x^q_1) L(\phi(x^q_1)). \]  

(1)
Expansion Factor

Definition

- The expected length of a codeword is

\[ E(L) = \sum_{x_1^q \in \mathcal{X}^q} P(x_1^q)L(\phi(x_1^q)). \]  

- We define the expansion factor of a shaping code to be

\[ f = \frac{E(L)}{q}. \]
Expansion Factor

Definition

- The expected length of a codeword is

\[
E(L) = \sum_{x_1^q \in \mathcal{X}^q} P(x_1^q)L(\phi(x_1^q)).
\]  

(1)

- We define the expansion factor of a shaping code to be

\[
f = \frac{E(L)}{q}
\]

(2)

Example

- Shaping code defined by mapping \{11 \rightarrow 111, 10 \rightarrow 10, 01 \rightarrow 10, 00 \rightarrow 0\}.

E(L) = \frac{1}{4}(3 + 3 + 2 + 1) = 2.25.

f = \frac{E(L)}{q} = \frac{2.25}{4} = 0.5625.
**Expansion Factor**

**Definition**

- The expected length of a codeword is
  \[
  E(L) = \sum_{x_i^q \in X^q} P(x_i^q)L(\phi(x_i^q)).
  \]  

- We define the expansion factor of a shaping code to be
  \[
  f = \frac{E(L)}{q}.
  \]

**Example**

- Shaping code defined by mapping \( \{11 \rightarrow 111, 10 \rightarrow 10, 01 \rightarrow 10, 00 \rightarrow 0\} \).
- \( E(L) = \frac{1}{4} (3 + 3 + 2 + 1) = 2.25 \)
- \( f = \frac{E(L)}{q} = 1.125 \)
Probability of Occurrence

Definition

- Consider the first \( l \) symbols of \( \phi(X) \), denoted by \( y_1^l \). Its probability is \( Q(y_1^l) \).
Probability of Occurrence

Definition

- Consider the first $l$ symbols of $\phi(X)$, denoted by $y_1^l$. Its probability is $Q(y_1^l)$
- We denote the number of $\beta_i$ in sequence $y_1^l$ by $N_i(y_1^l)$
Probability of Occurrence

Definition

- Consider the first $l$ symbols of $\phi(X)$, denoted by $y_1^l$. Its probability is $Q(y_1^l)$
- We denote the number of $\beta_i$ in sequence $y_1^l$ by $N_i(y_1^l)$

The probability of occurrence $\hat{Y}$ in encoded sequences $\phi(X)$ is

$$\hat{P}_i = \Pr(\hat{Y} = \beta_i) = \lim_{l \to \infty} \sum_{y_1^l} N_i(y_1^l)Q(y_1^l)/l = \lim_{l \to \infty} \frac{E(N_i(Y_1^l))}{l}.$$ (3)
Probability of Occurrence

Definition

- Consider the first $l$ symbols of $\phi(X)$, denoted by $y_1^l$. Its probability is $Q(y_1^l)$
- We denote the number of $\beta_i$ in sequence $y_1^l$ by $N_i(y_1^l)$

The probability of occurrence $\hat{Y}$ in encoded sequences $\phi(X)$ is

$$\hat{P}_i = Pr(\hat{Y} = \beta_i) = \lim_{l \to \infty} \sum_{y_1^l} N_i(y_1^l) \frac{Q(y_1^l)}{l} = \lim_{l \to \infty} \frac{E(N_i(Y_1^l))}{l}. \quad (3)$$

Lemma

For a prefix-free shaping code $\phi: X^q \to Y^*$, $\hat{Y}$ exists and

$$\hat{P}_i = E(N_i(\phi(X^q))) \frac{1}{E(L)} \quad (4)$$

Once we know the probability of occurrence, we can calculate the cost per output symbol $\sum_i \hat{P}_i U_i$ (we also call it average wear cost).
Data shaping codes try to reduce the wear cost, there are two different goals.

- The first goal is to minimize the average cost per output symbol (average cost), given a fixed expansion factor (Type-I minimization).

We try to solve the following type-I minimization problem:

\[
\min \hat{P}_i \sum_i \hat{P}_i U_i \\
\text{subject to } H(\hat{Y}) \geq H(X) f \sum_i \hat{P}_i = 1.
\] (5)

High rate is required in flash memory device for low encoding/decoding time complexity and high storage capacity.
Data shaping codes try to reduce the wear cost, there are two different goals.

The first goal is to minimize the average cost per output symbol (average cost), given a fixed expansion factor (Type-I minimization).

\[
\text{minimize } \sum_i \hat{P}_i U_i \\
\text{subject to } H(\hat{Y}) \geq H(X) \\
f \sum_i \hat{P}_i = 1.
\]

High rate is required in flash memory device for low encoding/decoding time complexity and high storage capacity.
Data shaping codes try to reduce the wear cost, there are two different goals.

- The first goal is to minimize the **average cost per output symbol** (average cost), given a fixed expansion factor (Type-I minimization).
- We try to solve the following type-I minimization problem

\[
\begin{align*}
\text{minimize} & \quad \sum_i \hat{P}_i U_i \\
\text{subject to} & \quad H(\hat{Y}) \geq \frac{H(X)}{f} \\
& \quad \sum_i \hat{P}_i = 1.
\end{align*}
\]
Type-I and Type-II Minimization

- Data shaping codes try to reduce the wear cost, there are two different goals.
- The first goal is to minimize the average cost per output symbol (average cost), given a fixed expansion factor (Type-I minimization).
- We try to solve the following type-I minimization problem

\[
\begin{align*}
\text{minimize} & \quad \sum_i \hat{P}_i U_i \\
\text{subject to} & \quad H(\hat{Y}) \geq \frac{H(X)}{f} \\
& \quad \sum_i \hat{P}_i = 1.
\end{align*}
\]

- High rate is required in flash memory device for low encoding/decoding time complexity and high storage capacity.

Liu, Huang, Bergman, Siegel (CMRR)
Optimal Shaping Codes
March. 2018 7 / 17
Type-I and Type-II Minimization

- The second goal is to minimize the **average cost per input symbol** (total cost) and find the optimal expansion factor (Type-II minimization).
The second goal is to minimize the *average cost per input symbol* (total cost) and find the optimal expansion factor (Type-II minimization).

We try to solve the following *type-II* minimization problem

\[
\begin{align*}
\text{minimize} & \quad f \sum_i \hat{P}_i U_i \\
\text{subject to} & \quad H(\hat{Y}) \geq \frac{H(X)}{f} \\
& \quad \sum_i \hat{P}_i = 1.
\end{align*}
\]
Performance of Shaping Code

Theorem (Optimal Type-I Shaping)

Given the distribution $P$ of source words and a cost vector $U$, the minimum average wear cost we can get from a shaping code $\phi : X^q \rightarrow Y^*$ with expansion factor $f = \frac{E(L)}{q}$ is bounded by $\sum_i \hat{P}_i U_i$, where $\hat{P}_i = \frac{1}{N} 2^{-\mu U_i}$, $\mu$ is a positive constant selected such that $H(\hat{Y}) = \sum_i \hat{P}_i \log_2 \hat{P}_i = H(X)/f$, and $N$ is a normalization constant.

Theorem (Optimal Type-II Shaping)

Let $P$ be the source distribution and let $U$ be a cost vector. If $U_1 \neq 0$, then the minimum total wear cost of a shaping code $\phi : X^q \rightarrow Y^*$ is given by $f \sum_i \hat{P}_i U_i$, where $\hat{P}_i = 2^{-\mu U_i}$, $\mu$ is a positive constant selected such that $\sum_i 2^{-\mu U_i} = 1$, and the expansion factor $f$ is

$$f = \frac{H(X)}{- \sum_i \hat{P}_i \log_2 \hat{P}_i}.$$  \hspace{1cm} (7)

If $U_1 = 0$, then the total cost is a decreasing function of $f$. \qed
Minimal total wear cost vs expansion factor $f$ when source is random with cost $[1,2,3,4]$
Equivalence Theorem and Separation Theorem

Theorem (Equivalence Theorem)

Let $P$ be the source distribution. A type-I shaping code with cost vector $U$ and expansion factor $f$ is a type-II shaping code with cost vector $U'$ where

$$U_i' = - \log_2 \hat{p}_i.$$  

(8)

$\hat{P} = \{\hat{p}_i\}$ is the optimal probability distribution given in optimal type-I shaping theorem.

Theorem (Separation Theorem)

An optimal general shaping code for a given expansion factor $f$ can be constructed by a concatenation of lossless compression with type-II shaping code for uniform i.i.d source.
Equivalence Theorem and Separation Theorem

Theorem (Equivalence Theorem)

Let $P$ be the source distribution. A type-I shaping code with cost vector $\mathcal{U}$ and expansion factor $f$ is a type-II shaping code with cost vector $\mathcal{U}'$ where

$$U'_i = -\log_2 \hat{p}_i.$$ (8)

$\hat{P} = \{\hat{p}_i\}$ is the optimal probability distribution given in optimal type-I shaping theorem.

Theorem (Separation Theorem)

An optimal general shaping code for a given expansion factor $f$ can be constructed by a concatenation of lossless compression with type-II shaping code for uniform i.i.d source.

- Equivalence theorem: There is a bijection between optimal type-I and optimal type-II shaping codes.
Theorem (Equivalence Theorem)

Let $P$ be the source distribution. A type-I shaping code with cost vector $U$ and expansion factor $f$ is a type-II shaping code with cost vector $U'$ where

$$U'_i = -\log_2 \hat{p}_i.$$  \hspace{1cm} (8)

$\hat{P} = \{\hat{p}_i\}$ is the optimal probability distribution given in optimal type-I shaping theorem.

Theorem (Separation Theorem)

An optimal general shaping code for a given expansion factor $f$ can be constructed by a concatenation of lossless compression with type-II shaping code for uniform i.i.d source.

- Equivalence theorem: There is a bijection between optimal type-I and optimal type-II shaping codes.
- Separation theorem: We only need to design shaping code for uniform i.i.d source.
Optimal Shaping Code Design

- Type-I shaping: Converting this problem into a concatenation of optimal lossless compression and a type-II shaping problem.

1. Compress the source file, calculate the compression ratio $g = \log_2 |X| H(X)$. Set $f' = fg$.

2. Calculate symbol probability distribution $\hat{P} = \{\hat{p}_i\}$ minimizing average cost for a uniform random source and expansion factor $f'$ using optimal type-I shaping theorem.

3. Define a cost vector $U' = \{U'_i\}$ by $U'_i = -\log_2 \hat{p}_i$.

4. Design a type-II shaping code for a uniform i.i.d source and cost vector $U'$.

5. Concatenate an optimal lossless compression code with the shaping code designed in the preceding step.
Optimal Shaping Code Design

- **Type-I shaping:** Converting this problem into a concatenation of optimal lossless compression and a type-II shaping problem.

**Require:** Source $X$, cost vector $U$, expansion factor $f$

1. Compress the source file, calculate the compression ratio $g$, for an optimal lossless compression, $g = \log_2 |X| / H(X)$. Set $f' = fg$.

2. Calculate symbol probability distribution $\hat{P} = \{\hat{p}_i\}$ minimizing average cost for a uniform random source and expansion factor $f'$ using optimal type-I shaping theorem.

3. Define a cost vector $U' = \{U'_i\}$ by $U'_i = -\log_2 \hat{p}_i$.

4. Design a type-II shaping code for a uniform i.i.d source and cost vector $U'$.

5. Concatenate an optimal lossless compression code with the shaping code designed in the preceding step.
- Type-II shaping: Varn Codes.
- Tree-based, fixed-to-variable length codes that minimize total cost for a specified codebook size $K$.
- Designed specifically for uniformly distributed i.i.d source.
- Expand the leaf node that has the minimum cost.
- Example: symbol '1' has cost 0.58, symbol '0' has cost 1.58, codebook size $N = 4$ ($q = 2$).
Type-II shaping: Varn Codes.

Tree-based, fixed-to-variable length codes that minimize total cost for a specified codebook size $K$.

- Designed specifically for uniformly distributed i.i.d source.
- Expand the leaf node that has the minimum cost.

Example: symbol '1' has cost 0.58, symbol '0' has cost 1.58, codebook size $N = 4$ ($q = 2$).
Optimal Shaping Code Design

- Type-II shaping: Varn Codes.
- Tree-based, fixed-to-variable length codes that minimize total cost for a specified codebook size $K$.
- Designed specifically for uniformly distributed i.i.d source.
- Expand the leaf node that has the minimum cost.
- Example: symbol '1' has cost 0.58, symbol '0' has cost 1.58, codebook size $N = 4$ ($q = 2$).
- $0.58 = -\log_2 \frac{2}{3}$, $1.58 = -\log_2 \frac{1}{3}$. 

\[ \begin{align*}
27 & : 8 \\
27 & : 4 \\
9 & : 4 \\
3 & : 2 \\
3 & : 1 \\
\end{align*} \]
Optimal Shaping Code Design

- Type-II shaping: Varn Codes.
- Tree-based, fixed-to-variable length codes that minimize total cost for a specified codebook size $K$.
- Designed specifically for uniformly distributed i.i.d source.
- Expand the leaf node that has the minimum cost.
- Example: symbol '1' has cost 0.58, symbol '0' has cost 1.58, codebook size $N = 4$ ($q = 2$).
- $0.58 = -\log_2 \frac{2}{3}$, $1.58 = -\log_2 \frac{1}{3}$. 
Experiment Setup

- MLC shaping code was applied to the ASCII representation of the English-language novel *The Count of Monte Cristo* and Chinese-language work *Collected Works of Lu Xun, Volumes 1–4*.
- The original file and file coded with rate-1 type-I shaping code were written to our flash memory testboard.
- The first half of the data was written on the lower page and the second half of the data was written on the upper page.
- For the next programming cycle, we "rotate" the data. The data written on the i-th wordline is written on the (i+1)-st wordline.
- After every 100 cycles, pseudo-random data is written to the block and then read back to calculate the bit-error-rate (BER).
- To compare the performance of shaping code with compression, we rescaled the P/E cycle count of the shaping code by the compression ratio and compared the result to P/E cycling of pseudo-random data.
Bit Error Rate Results

(a) BER Performance of English-language novel

(b)
Bit Error Rate Results

(a) BER Performance of Chinese-language novel Liu, Huang, Bergman, Siegel (CMRR) Optimal Shaping Codes

(b)
Conclusion

- Shaping code is used to reduce the average wear cost and total wear cost.
  - Type-I shaping: minimize cost per output symbol.
  - Type-II shaping: minimize cost per input symbol.
- Equivalence theorem and separation theorem suggest how to design the shaping code encoder.
  - Type-I shaping: convert this problem into a type-II shaping problem.
  - Type-II shaping: convert this problem into a concatenation of compression and type-II shaping for uniform i.i.d source.
- Optimal type-II shaping codes for a uniform i.i.d source: Varn Codes.
- Experimental results for MLC shaping codes on English and Chinese text show a reduction in bit error rate.