

Hybrid Golomb codes for a group of quantised GG sources

S. Xue, Y. Xu and B. Oelmann

Abstract: The authors develop a class of codes for quantised generalised Gaussian (GG) sources. This new class of codes, that are called hybrid Golomb (HG) codes, are hybrids of Golomb–Rice (GR) codes and exp-Golomb (EG) codes. They integrate the properties of both GR and EG codes, which makes their performance more robust under variable source parameters. The efficiencies of the three classes of codes are compared and it is shown that the set of HG codes have efficiencies of approximately 70% or greater for a wide range of parameter values, whereas GR and EG codes can have efficiencies lower than 20%. The efficiencies of the set of HG codes are also compared with the set of EG codes that have the best performance under parameter scaling. It is shown that this set of HG codes still achieve a higher efficiency.

1 Introduction

Generalised Gaussian (GG) sources are a group of sources with exponentially distributed source symbols and are often encountered in image and video coding algorithms. Quantisation methods are used to map the GG probability density functions (pdf) to the actual GG sources. Traditionally, Golomb–Rice (GR) codes and exp-Golomb (EG) codes are used in the coding of GG sources and both are claimed to be nearly optimal. GR and EG codes can both be represented in a form such that each codeword consists of a prefix and a suffix. The efficiencies of GR and EG codes are related to the shapes of the GG pdfs, as well as to the parameters used in quantisation and the chosen suffix length. The sets of GR and EG codes with the same suffix length usually have complimentary performances regarding efficiency when used in the coding of the same GG source. Furthermore, the efficiencies of GR and EG codes are both quite sensitive to changes of the source parameters such as the pdf shape or quantisation parameters. Therefore, efficient coding using GR or EG codes can only be achieved by carefully mapping one set of codes to the source with particular source parameters. When the parameters of the quantised GG source are changing or are not known, such as in scalable video coding, it is difficult to choose a set of codes with good overall performances.

In this paper we present a new class of codes called ‘hybrid Golomb’ (HG) codes. They integrate the properties of both GR and EG codes and therefore offer compromised performance. The efficiency of the selected set of HG codes is never below approximately 70% despite the changes of the values of the source parameters. This means that we are able to achieve high efficiency for a large range of parameter values with minimum regard to the selection of the length of

the suffix. We also compare HG codes with the set of codes that have the highest overall efficiency in a group of sets of GR and EG codes. The result shows that HG codes are always better and have efficiencies greater by up to 8.5% for the range of parameter values that we have studied.

2 Related work

By applying the Huffman source coding algorithm, we are able to encode the letters of a finite source alphabet into Huffman codes [1], which are uniquely decipherable codes with minimum expected codeword length. However, for letters of an infinite source alphabet, the Huffman algorithm cannot be applied directly, as the algorithm requires the coding to start by ‘merging’ the least probable letters in the alphabet. S.W. Golomb [2] initiated the early work in coding infinite alphabet of non-negative integers with probability assignment:

$$P(i) = (1 - \theta)\theta^i, \quad i \geq 0 \quad (1)$$

into optimal codes. The parameter he studied satisfies the condition $\theta^m = 1/2$, where i is the integer to be coded and m is a known integer. Gallager and Van Voorhis [3] generalised Golomb’s result by allowing θ to be an arbitrary value within the range $0 < \theta < 1$. They also showed that optimum codes can be generated by expressing source integers in terms of a quotient j and a remainder r , $i = lj + r$ where r satisfies the condition

$$\theta^l + \theta^{(l+1)} \leq 1 \leq \theta^{(l-1)} + \theta^l \quad (2)$$

The codeword is constructed by concatenating the unary expression for $j = \lfloor i/l \rfloor$ with a Huffman code for r whose length is $\lfloor \log_2 l \rfloor$ or $\lfloor \log_2 l \rfloor + 1$. Rice [4] studied the special case where l is a power of 2: $2^k = l$, and developed a code referred to as Golomb–Rice codes (GR). Each GR code has a fixed k -bit suffix instead of a suffix with length $\lfloor \log_2 l \rfloor$ or $\lfloor \log_2 l \rfloor + 1$. For example, if the suffix is 1 bit, the integer 7 will be coded into GR code 11101. GR codes with nearly optimal performance can be found and these have been applied to the coding of Laplacian distributed integer sources such as in the coding of prediction errors in lossless image coding algorithms [5]. Laplacian integer sources have been emphasised for many years as there are a wide range of

applications for which these models are suitable. However, several circumstances exist which require the handling of sources of integer distributions more highly peaked than those of Laplacian models. This occurs, for example, in scalar quantised, run-length coded image subbands, which produce integer sources that are often modelled using a generalised Gaussian (GG) pdf. The GG is a family of pdfs parameterised by a shape parameter ν and can be expressed as:

$$f_X(x) = c_1 e^{-c_2 |x|^\nu} \quad (3)$$

where c_1 and c_2 are both functions of the shape parameter ν and the standard deviation σ :

$$c_1 = \frac{\nu \eta(\sigma, \nu)}{2\Gamma(\frac{1}{\nu})} \quad (4)$$

$$c_2 = [\nu \eta(\sigma, \nu)]^\nu \quad (5)$$

$$\eta(\sigma, \nu)^\nu = \frac{1}{\sigma} \left[\frac{\Gamma(\frac{\nu}{\nu})}{\Gamma(\frac{1}{\nu})} \right]^{\frac{1}{\nu}} \quad (6)$$

When ν is 1 the GG is Laplacian, and for ν equal to 2 it is Gaussian. As $\nu \rightarrow \infty$ the distribution will become uniform.

A GG pdf is usually quantised by a step size δ to be mapped onto a discrete GG source.

At the same time another class of codes were also developed by Teuhola [6] called the exp-Golomb (EG) codes. EG codes, in contrast to GR codes, in which the number of codewords of any length is constant, have an exponentially growing number of codewords that have the same codeword length. EG codes can be thought of as a special case of a class of codes described by Elias in [7]. Because the number of codewords increases exponentially with the code length, EG codes are matched to pdfs with high peaks such as GG pdfs with $\nu < 1.0$ in which the exponential decay rate reduces in relation to the distance from the origin.

The efficiencies of GR and EG codes are actually functions of the shape parameter ν , the length of the suffix k , and the normalised quantiser step size δ/σ . For the sets of codewords with the same suffix length, GR codes perform better with larger ν and larger δ/σ , whereas, this is reversed

for EG codes. For a particular ν , GR or EG codes with a given suffix length can only perform optimally or nearly optimally within a small range of δ/σ values. This is especially true for GR codes. The efficiencies degrade rapidly outside this small range of δ/σ values and can be as poor as 20% or even less. Therefore, efficient coding using GR or EG code can only be achieved by careful selection of one set of codes by determining the suffix length of the codes according to a quantised GG source with fixed δ/σ and ν . When the parameters of the quantised GG source change or are not known, it is difficult to choose a set of codes whose performance is in generally acceptable. This is often the case in scalable video transmission [8], where δ/σ is changeable according to the bandwidth. Recent work in subband image coding has resulted in the highly peaked, wide-tailed integer distributions such as GG becoming more common. In several recent published works [9–12], it was pointed out that wavelet-transformed image data can be modelled using GG sources with a shape parameter in the range $0 < \nu < 1.0$. Therefore, in this paper we will focus on the study of GG sources with the shape parameter in the interval (0,1).

3 Code description

The HG codes we present in this paper are constructed in a similar manner to GR and EG codes. The prefix of a GR code is a unary expression of a quotient and the suffix of a GR code has a fixed length. Thus the codeword length grows linearly whereas the number of codewords with the same length is constant. As for EG codes, the length of the prefix also increases linearly, but the number of codewords that have the same length increases exponentially. In constructing the HG codes, we allow the length of the prefix of a HG code to increase linearly. However, in contrast to both GR and EG codes, the number of codewords of the same length increases exponentially for codewords with length $2i + k$, $i > 1$. Yet the number of codewords with length $2i + 1 + k$, $i > 1$ remains constant, where k is the number of bits of the suffix.

When the suffix of HG codes is set to be k bits, the first group of 2^k codes are allocated the prefix '0', and the second group of 2^k codes are allocated the prefix '10'. The prefixes of the first and the second groups of codes are both concatenated with k -bit suffixes. For the rest of the groups of

Table 1: Codeword table of GR, EG and HG codes

	GR codes		EG codes		HG codes	
	Prefix	Length	Prefix	Length	Prefix	Length
0	0	1	0	1	0	1
1	10	2	100	3	10	2
2	110	3	101	3	1100	4
3	1110	4	11000	5	11010	5
4	11110	5	11001	5	11011	5
5	111110	6	11010	5	111000	6
6	1111110	7	11011	5	111001	6
7	11111110	8	1110000	7	111010	6
8	111111110	9	1110001	7	1110110	7
9	1111111110	10	1110010	7	1110111	7
10	11111111110	11	1110011	7	11110000	8
...

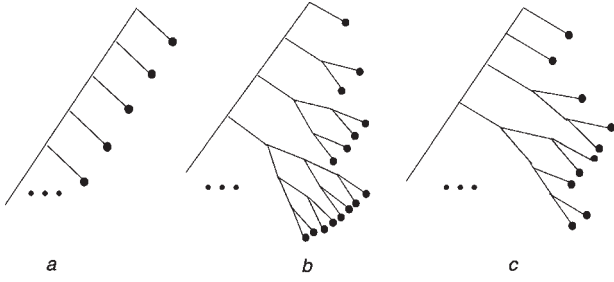


Fig. 1 Codeword trees of the three types of codes

- a GR codes
- b EG codes
- c HG codes

codewords that have the same prefix length, codes are allocated according to two different situations: for codewords with length $2i + k$, $i > 1$, that is when the length of the prefix is an even number, we use i '1's (a unary expression for i) plus a '0' concatenated by a binary expression from 0 to $2^k(2^i - 1) - 1$ for the remaining $i + k - 1$ bits; whereas for codewords with length $2i + 1 + k$, $i > 1$, that is when the length of prefix is an odd number, we use i '1's (a unary expression for i) plus a '0' concatenated by a binary expression from $2^{k+1}(2^{i-1} - 1)$ to $2^{i+k} - 1$ for the remaining $i + k$ bits. Table 1 gives the codeword tables for the GR, EG codes and HG codes ($k = 0$). Figure 1 provides the codeword trees for the GR, EG and HG codes ($k = 0$).

4 Comparison of performance

In this Section, we compare the efficiencies of HG codes to those of EG and GR codes. For the comparison, we select GR codes with $k = 0, 1, 2, 3$, and EG codes with $k = 0, 1, 2, 3$. These are compared with HG codes without any suffix.

The efficiency of a set of codes is calculated as:

$$\eta = h/L_{av} \quad (7)$$

where η represents the efficiency of the codes, L_{av} stands for the average codeword length of a certain class of codes, and h is the source entropy.

In associating a discrete source with a GG pdf, a uniform scalar quantiser of step size δ with a dead zone of $(1 + \alpha)\delta$ at the origin is used. This is illustrated in Fig. 2. Such uniform scalar quantiser with a deadzone is commonly used in many coding systems, especially in many standard image and video coding algorithms [13]. There are three types of mappings of the output of the uniform scalar quantiser. As Fig. 2 also shows, the output of the quantiser can be

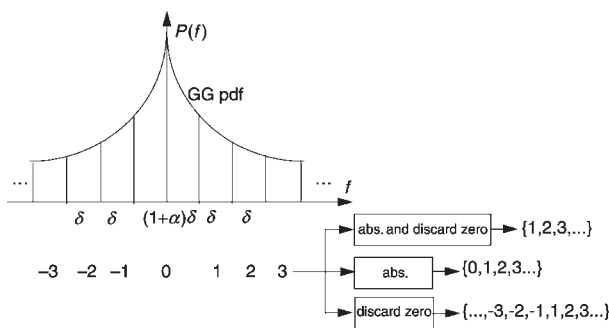


Fig. 2 GG pdf and three types of mapping of uniform scalar quantisation

mapped to positive, non-negative, and two-sided non-zero discrete sources. A GG discrete source is specified after the shape parameter ν , the normalised quantiser step size δ/σ and the deadzone parameter α are identified and one of the three mappings is selected. Positive, non-negative, and two-sided non-zero distributions are all important in the image coding applications [13]. In many wavelet, JPEG, and MPEG image and video coding applications, transformed, scalar quantised image data are raster-scanned to generate a description using pairs of the form (run, level). The 'run' is the number of coefficients with value '0' encountered before the next significant coefficient. The 'level' is the magnitude and sign of the integer representing the significant coefficient. The run values are actually non-negative integers or positive integers if runs of zero are not encoded. The levels are two-sided, non-zero integers. In studying the source properties, as which one of the three types of mapping is chosen makes no essential difference, we use a positive mapping.

Therefore, the entropy h is calculated as:

$$h = - \sum_{i=1}^{\infty} \frac{P(i)}{1 - P(0)} \log_2 \left(\frac{P(i)}{1 - P(0)} \right) \quad (8)$$

$$P(0) = 2 \int_0^{(1+\alpha)\frac{\delta}{2}} f_X(x) dx \quad (9)$$

$$P(i) = \frac{2}{1 - P(0)} \int_{(2i-1+\alpha)\frac{\delta}{2}}^{(2i+1+\alpha)\frac{\delta}{2}} f_X(x) dx \quad (10)$$

Here $1 - P(0)$ is used as a normalisation to eliminate the influence of the deadzone parameter α on the source properties. The average codeword length is then calculated as:

$$L_{av} = - \sum_{i=1}^{\infty} \frac{P(i)}{1 - P(0)} l(i) \quad (11)$$

where $l(i)$ is the length of the i th codeword of a particular class of codes.

Figure 3 shows a comparison of the efficiencies of the three classes of codes under different values of ν . δ/σ is chosen within the range $(10^{-2}, 10^0)$, which is adequate for modelling image and video data [13,14]. As can be noticed from Fig. 3, for a particular ν , GR or EG codes can be optimal or nearly optimal through careful selection of the length of the suffixes. For a particular suffix length, EG codes are more efficient for smaller step sizes while GR codes are better for larger step sizes. However, in most cases, the codes only perform well within a small range of δ/σ values. Efficiency can decrease to below 20% outside this range. For the selected set of HG codes, although the efficiency peak may not be as high as those for GR or EG codes, an efficiency of more than nearly 70% can always be achieved. That is to say, HG codes are not optimal for any particular GG source but are more robust to code GG distributed data when the shape parameter and the quantiser step size are unknown.

We also notice here that the EG codes with $k = 0$ perform in a smaller manner to the HG codes we select ($k = 0$). This is a reasonable result as the HG codes are constructed to be an integration of GR and EG codes. The number of codewords that have the same length grows both linearly and exponentially, however it is obvious that the dominant effect will be that of the exponential increase. This means

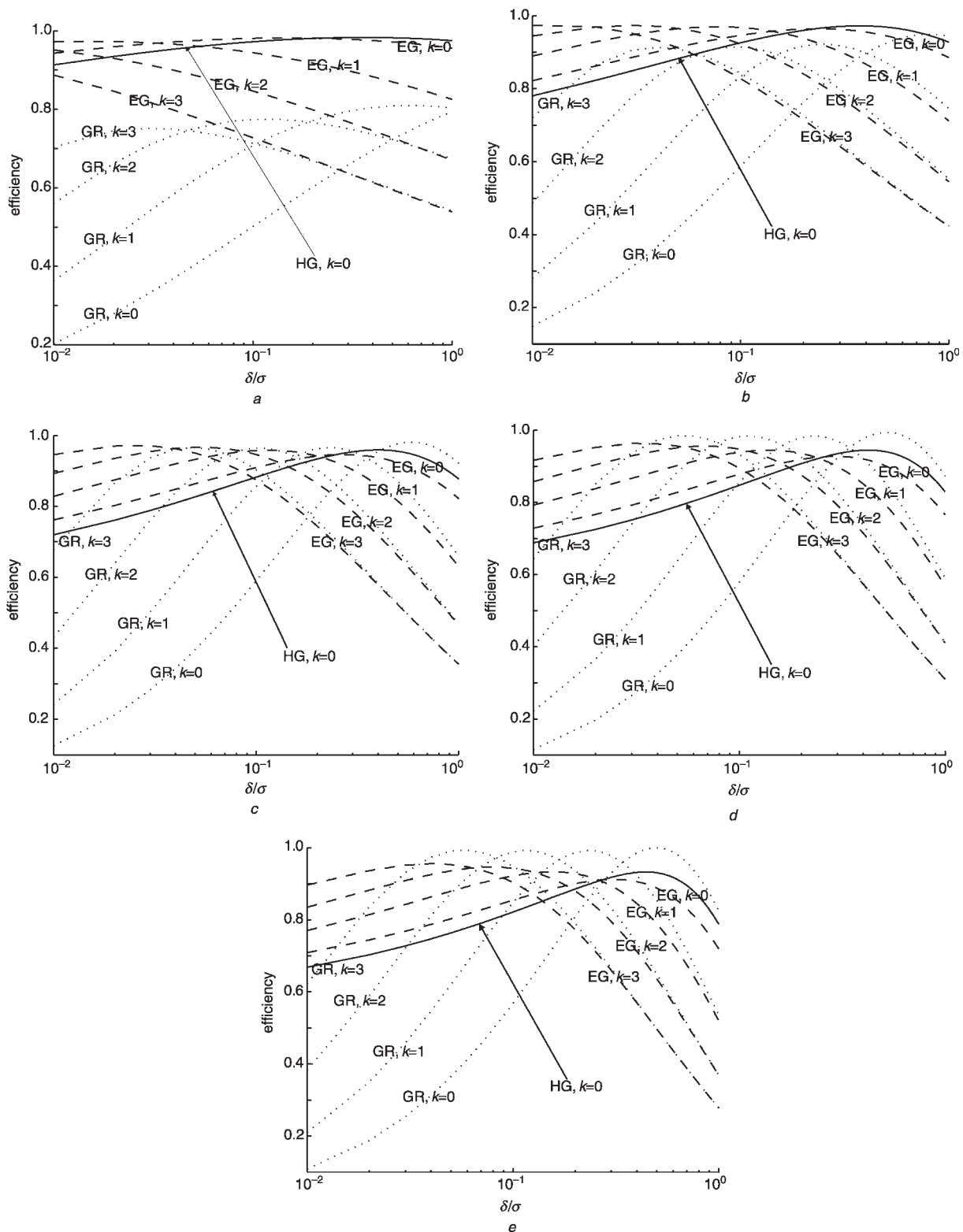


Fig. 3 Comparison of efficiencies of the three types of codes under different shape parameter v

- a* $v = 0.1$
- b* $v = 0.3$
- c* $v = 0.5$
- d* $v = 0.7$
- e* $v = 0.9$

that HG codes perform more like EG codes, especially when the suffix is of the same length.

Figure 4 shows the efficiency difference between HG codes and EG codes with $k = 0$. As we can see from Fig. 3, the performances of the two sets of codes are quite close to each other. The efficiency difference is

calculated as:

$$D(v) = \frac{\int_a^b \eta_{HG}(v, \delta/\sigma) d(\delta/\sigma) - \int_a^b \eta_{EG}(v, \delta/\sigma) d(\delta/\sigma)}{\int_a^b \eta_{EG}(v, \delta/\sigma) d(\delta/\sigma)} \quad (12)$$

where $D(v)$ is the efficiency difference, which is a function of v , $\eta_{HG}(v, \delta/\sigma)$ and $\eta_{EG}(v, \delta/\sigma)$ are the

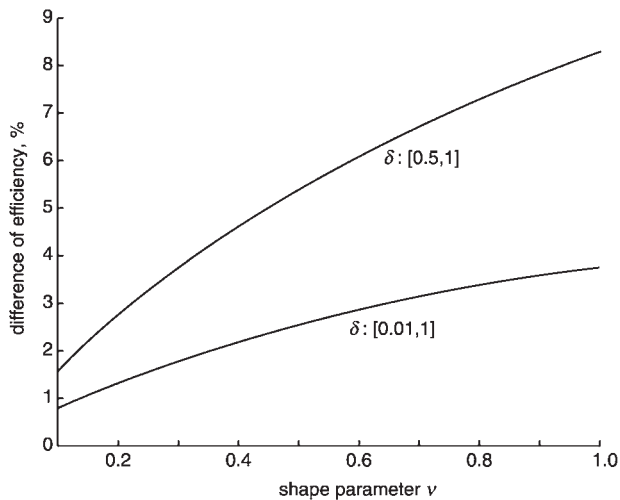


Fig. 4 Efficiency difference between HG codes and EG codes ($k = 0$)

efficiencies of the sets of HG codes and EG codes that are both functions of v and δ/σ and (a, b) is the range of the integration. Here we compare the two sets of codes for δ/σ within the range $[10^{-2}, 1]$ and $[0.5, 1]$.

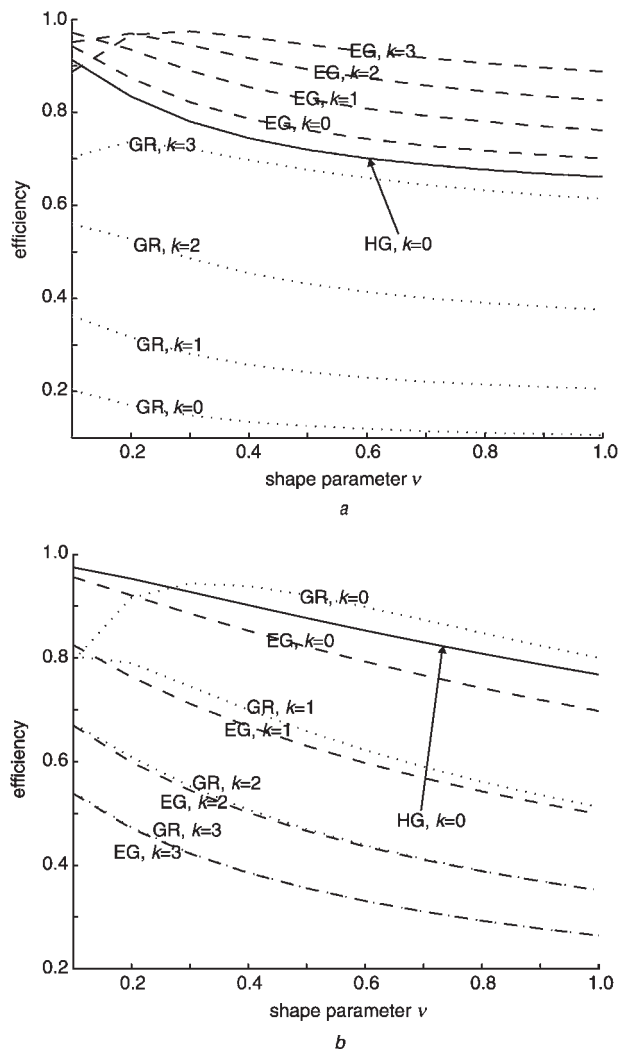


Fig. 5 Comparison of efficiencies of the three types of codes under different step size δ/σ

a $\delta/\sigma = 0.01$
b $\delta/\sigma = 1$

The results show that the set of HG codes have better performances than EG codes in both of these ranges of δ/σ values. HG codes outperform EG codes by up to nearly 8.5% when δ/σ is within the range $[0.5, 1]$. Whereas for the range $[10^{-2}, 1]$, the advantage is not as evident as for the range $[0.5, 1]$. It can also be seen that for larger v values, HG codes have better performances. These results indicate that in addition to robustness over different shape parameters and different step sizes, the selected set of HG codes perform even better under larger v and when δ/σ lies in the interval of larger values. Figure 5 shows a comparison of the efficiencies of the three types of codes for different δ/σ . Here, for the selected δ/σ , our set of HG codes can almost always achieve an efficiency greater than 70% for different values of shape parameter v , whereas GR and EG codes have lower efficiencies.

5 Conclusions

We have proposed HG codes for coding a group of quantised GG sources. For the selected set of HG codes ($k = 0$), an efficiency of approximately 70% or greater can always be observed for all the explored v and δ/σ values, whereas the efficiencies of GR and EG codes can fall below 20%. This indicates that this set of HG codes ($k = 0$) will be more efficient for a group of quantised GG sources because of their insensitivity to the changing of values of source parameters. In particular, we made comparisons between the set of EG codes ($k = 0$) that have the best overall performances among different GR and EG codes and the set of HG codes we have proposed. Results show that the set of HG codes ($k = 0$) also exceed the set of EG codes ($k = 0$) in performance in wide ranges of δ/σ values. The results also indicate that the set of HG codes ($k = 0$) is especially suitable for use in the situations when δ/σ lies in the larger value interval and when the shape parameter is larger.

6 References

- Huffman, D.: 'A method for the construction of minimum redundancy codes', *Proc. Inst. Radio Eng.*, 1952, **9**, (40), pp. 1098–1101
- Golomb, S.W.: 'Run-length encodings', *IEEE Trans. Inf. Theory*, 1966, **7**, (12), pp. 399–401
- Gallager, R.G., and Van Voorhis, D.C.: 'Optimal source codes for geometrically distributed integer alphabets', *IEEE Trans. Inf. Theory*, 1975, **3**, (21), pp. 228–230
- Rice, R.F.: 'Some practical universal noiseless coding techniques', Tech. Rep. JPL-79-22, Jet Propulsion Laboratory, Pasadena, CA, March 1979
- Weinberger, M.J., Seroussi, G., and Sapiro, G.: 'LOCO-I, A low complexity, context-based lossless image compression algorithm', *Proceedings of 1996 IEEE Data Compression Conf.*, Snowbird, UT, April 1996, pp. 140–149
- Teuhola, J.: 'A compression method for clustered bit-vectors', *Inf. Process. Lett.*, 1978, **10**, (7), pp. 308–311
- Elias, P.: 'Universal codeword sets and representations of the integers', *IEEE Trans. Inf. Theory*, 1975, **3**, (21), pp. 194–203
- Wang, Y., Ostermann, J., and Zhang, Y.Q.: 'Video processing and communications' (Prentice Hall, Upper Saddle River, NJ, USA)
- Laroia, R., and Farvardin, N.: 'A structured fixed-rate quantizer derived from a variable length scalar quantizer – Part I: Memoryless sources', *IEEE Trans. Inf. Theory*, 1993, **5**, (39), pp. 851–867
- Lopresto, S.M., Ramchandran, K., and Orchard, M.T.: 'Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework', *Proceedings of 1997 IEEE Data Compression Conf.*, Snowbird, UT, March 1997, pp. 221–230
- Calvagno, G., Ghirardi, C., Mian, G.A., and Rinaldo, R.: 'Modeling of subband image data for buffer control', *IEEE Trans. Circuits Syst. Video Technol.*, 1997, **4**, (7), pp. 402–408
- Birney, K.A., and Fischer, T.R.: 'On the modeling of DCT and subband image data for compression', *IEEE Trans. Image Process.*, 1995, **2**, (4), pp. 186–193
- Wen, J., and Villasenor, J.D.: 'Structured prefix codes for quantized low-shape-parameter generalized Gaussian sources', *IEEE Trans. Inf. Theory*, 1999, **5**, (45), pp. 1307–1314
- Wen, J., and Villasenor, J.D.: 'A class of reversible variable length codes for robust image and video coding', *Proceedings of 1997 Int. Conf. on Image processing*, Washington, DC, USA, October 1997, vol. 2