Context Clustering in Lossless Compression of Gray-Scale Image

Mantao Xu and Pasi Fränti

Department of Computer Science, University of Joensuu, P. O. Box 111,
80101 Joensuu, Finland
{Xu, Franti}@cs.joensuu.fi

Abstract. We consider and evaluate the context clustering method for lossless image compression based on the existing LOCO-I algorithm used in JPEG-LS – the latest lossless image compression standard. We employ the LOCO-I Med-predictor to enroll the error pixels. The contexts are defined by calculating gradient of current pixels. The three directional gradients are quantized with different codebook size (7, 9, 19) respectively. The error pixels are then corrected and encoded by the clustered-contexts. A main advantage of using the context clustering method is that it can eliminate the storage of probability vector. An adaptive arithmetic encoder is also introduced to yield a higher compression rate.

1 Introduction

Lossless image compression has been widely used in many applications fields such as picture archiving, geophysics surveying and telemedicine. Recently, several algorithms have been developed so well as the benchmarks of image compression. FELICS is a simple and efficient compression algorithm avoiding the time-consuming arithmetic coding [1]. The latest JPEG-LS standard was implemented based on the LOCO-I algorithm [2, 3]. CALIC, the Adaptive Context-based Lossless Image Codec [4], was evaluated to be in best performance by the JPEG working group. Both of the algorithms employ prediction techniques before context modeling. In addition to prediction techniques, the continuous-tone image can be divided into a set of bitplane images that are coded by CTW, the context tree weighting method [5]. An alternative of bi-level image compression is context clustering [6], replacing the conditional probability vector or density function in each context with the reference probability vector in its cluster of contexts. The method improves bi-level image compression performance by reducing the storage of the probability density functions (*PDF*s).

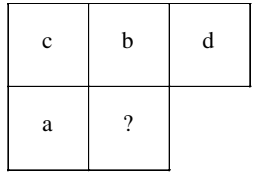
Since the conditional probability density function in each context is represented as a probability vector in a *Euclidean* space, context clustering can be used to quantize the probability density functions of the existing algorithms. Thus, only the referenced probability vectors need to be coded and transmitted to the decoder as a part of the storage of the compressed file. To implement lossless image compression, we first follow the LOCO-I Med-predictor. Then context clustering and an adaptive arithmetic

J. Bigun and T. Gustavsson (Eds.): SCIA 2003, LNCS 2749, pp. 328–334, 2003.

coder are employed. We replace the fixed-size quantizer in LOCO-I with an optimized quantizer. For reduction of *PDF* vector storage, the codeword of each corrected error pixel is divided into two parts. Context clustering is implemented only in *PDF* vector space by using the generalized Kullback-Leibler distance. Finally, we test the performances of JPEG-LS, CALIC, BTPC (bit plane predictive coding) and our context clustering algorithm for several gray-scale images. Experimental results show that the proposed algorithm yields a similar compression rate with JPEG-LS but provides a flexible framework and some variations of methods included in JPEG-LS.

2 Prediction

We first take a 4-pixel neighborhood as the context used for the prediction of current pixel; see figure 1. Since the predictor in CALIC seems more complicated, we here employ the LOCO-I predictor described in the right side of figure 1.



$$\hat{x} = \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b) \\ \max(a, b) & \text{if } c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}$$

Fig. 1. 4-pixel context in gray scale image and the Med-predictor in LOCO-I

The predicted error pixel is calculated as $\varepsilon(t) = x(t) - \hat{x}(t)$ and will be corrected later by bias cancellation for statistical modeling, where t is the current pixel position.

3 Context Modeling

The context modeling includes the following two steps: context quantization, and bias cancellation. The context is determined by calculating three gradients between the four context pixels: $g_1 = d - b$, $g_2 = b - c$, and $g_3 = c - a$.

3.1 Context Quantization

Each gradient variable can be quantized either by a *K-means* quantizer or a fast scalar quantizer [7]. The codebook size of g_1, g_2, and g_3 is 7, 9, and 19 respectively. We here assume that the value of g_1 is very close to the value of g_2 because they are the horizontally neighboring gradients while g_3 is the unique vertical gradient. We therefore assume the number of quantized values for g_3 should be more or less equal to the sum

of the numbers for g_1 and g_2. Then contexts are merged according to the quantized values. That is to say, the contexts with the identical quantized values are merged. The number of models is reduced further by assuming that symmetric contexts $C_i = [g_1, g_2, g_3]$ and $C_j = [-g_1, -g_2, -g_3]$ have the same statistical properties (with the difference of the sign). The total number of models is thus:

$$|C| = \frac{\left((2T_1+1)\cdot(2T_2+1)\cdot(2T_3+1)+1\right)}{2} \tag{1}$$

3.2 Bias Cancellation

Since the quantized context $C(t)$ is known both by encoder and decoder, the prediction error pixel $\varepsilon(t)$, which is used to recover the image pixel $x(t)$, can be corrected adaptively by $C(t)$. The adaptive correction is used to cancel the bias offset of TSGD distribution [3] due to the fixed predictor. We denote the estimated error pixel and the corrected error pixel as $\hat{\varepsilon}(t)$ and $\bar{\varepsilon}(t)$ respectively. We here present the pseudo-codes of the bias cancellation [8] as follows:

```
PROCEDURE BiasCancellation(x, x̂, C) return ε̄
  FOR q ← 1 TO NumberOfQuantizedContexts DO
    S(q)←0;N(q) ←1;
  FOR t ← 1 TO NumberOfImagePixels DO
    q ← QuantizationIndexOf(C(t));
    ε̂(t) ← S(q) / N(q);
    x̄(t) ← x̂(t) + ε̂(t);
    ε̄(t) ← x(t) - x̄(t);
    S(q) ← S(q)+ ε̄(t); N(q) ← N(q)+ 1;
    IF N(q) ≥ 128 THEN
      S(q) ← S(q)/2; N(q) ← N(q)/2;
  return ε̄;
END PROCEDURE.
```

4 Statistical Modeling

To prepare for statistical modeling, we need to interleave the corrected errors into non-negative region by a transformation in LOCO-I; see formula (4) in [2]. Then the conditional *PDF* functions in quantized contexts can be calculated for the encoder. However, the conditional *PDF* functions, which should be stored in the compressed file, will take a large memory space. To reduce the storage of the *PDF* functions, we here separate the corrected error pixel $\bar{\varepsilon}$ into two variables, ε_1 and ε_2, which will be coded separately. A simple solution is to divide each corrected error $\bar{\varepsilon}$ by a constant integer $d = 16$ and calculate the corresponding quotient and modulo as:

$$\varepsilon_1 = \text{floor}(\bar{\varepsilon} / d)\ ,\ \varepsilon_2 = \text{mod}(\bar{\varepsilon}, d) \tag{2}$$

4.1 Context Clustering

To reduce *PDF* function storage further, we employ context clustering in *PDF* vector space and replace the *PDF* vector in each context with the reference *PDF* vector in its context cluster. A given context C is represented by the pair of probabilities $C = (f, p)$, where f is the probability of the context C and p is the conditional *PDF* function or vector of the corrected error pixel in the context C.

An optimized codebook Y consisting of 11 reference vectors is generated by the *K-means* clustering algorithm. The jth reference vector y_j is calculated as:

$$y_j = (\bar{f}_j, \bar{p}_j)\ ,\ \bar{f}_j = \sum_{L(i)=j} f_i\ ,\ \bar{p}_j = \sum_{L(i)=j} \frac{f_i \cdot p_i}{\bar{f}_j} \tag{3}$$

where $L(i)$ is the partition table. Instead of the conditional *PDF* vector p, the reference conditional *PDF* vector $\bar{p}$ is used to encode all error pixels neighbored by any context in cluster j. The number of conditional *PDF* vectors used by encoder is therefore reduced. To optimize the codebook Y in the *PDF* vector space $\{(f, p) \mid f \in R, p \in R^{\dim}\}$, we employ Kullback-Leibler distance as the vector-to-cluster distance in this work. The quantized contexts are therefore reallocated into clusters by clustering only in the *PDF* vector space.

4.2 Clustering Distance and Distortion Function

As shown in section 4.1, each context is optimally allocated into a cluster with a reference conditional *PDF* vector $\bar{p}$, which is used to encode the corrected error pixel instead of its own *PDF* vector p. So the distortion of context clustering should reflect the total difference of entropy gained by using the two kinds of *PDF* vectors above. In compression of gray-scale images, the distortion function is naturally generalized as:

$$f = \sum_{j=1}^{M} \sum_{L(i)=j} f_i \cdot D(p_i \,\|\, \bar{p}_j) \tag{4}$$

where $D(p_i \,\|\, \bar{p}_j)$ is Kullback-Leibler distance and calculated as:

$$D(p_i \,\|\, \bar{p}_j) = \sum_{k=1}^{\dim} p_{ik} \log_2 (\bar{p}_{jk} / p_{ik}) \tag{5}$$

5 Adaptive Arithmetic Coding

In this section we introduce an adaptive arithmetic coder, of which *PDF* function is updated with the encoded pixels. The adaptive coder is demonstrated by encoding a sequence $S = \{ S(i) \mid i = 1, \ldots\ldots, N \}$ that consists of *a*, *b* and *c*. The sequence *S* is described as *aaabbaaccbbccbb*, from which the initial frequency of each symbol is calculated as $f_a = 5, f_b = 6$ and $f_c = 4$. The initial frequencies here need to be transmitted to decoder. Then the sequence can be coded by a set of adaptive probabilities as { 5/15, 4/14, 3/13, 6/12, 5/11, 2/10, 1/9, 4/8, 3/7, 4/6, 3/5, 2/4, 1/3, 2/2, 1/1 }. To make it clear, we present the pseudo-codes of this example adaptive coder as:

```
PROCEDURE AdaptiveCoder(S,fa,fb,fc)
  ga ← fa; gb ← fb; gc ← fc; N ← fa+fb+fc;
  FOR i ← 1 TO NumberOfCharacters DO
    ArithmeticCoding(S(i),ga/N,gb/N,gc/N);
    IF S(i)= a THEN ga ← ga-1;
    ELSE IF char(i)= b THEN gb ← gb-1;
    ELSE gc ← gc-1;
    N ← N-1;
END PROCEDURE.
```

In the adaptive coder above, once if the current character has been coded, it will be removed from the sequence, therefore, the next pixel to be encoded is coded by probability distribution of the rest of the sequence. Hence the adaptive arithmetic coder can yield a better compression rate if the sequence size is not very large. Similarly, the adaptive arithmetic coder works as well in coding the corrected error pixels by updating the conditional *PDF* vectors.

6 Experimental Results

This section describes the testing results of the lossless image compression with context clustering. We used six standard images for testing our context clustering algorithm; see table 1, which lists the final bit rate of six compressed images. We also experimented three lossless image compression algorithms, CALIC, LOCO-I and BTPC, to evaluate the performance of our algorithm.

From testing results, we learned that the context clustering achieves a compression ratio close to LOCO-I but more flexible than others in selection of contexts and the conditional *PDF*s. We chose 7, 9 and 19 as the quantizer size for each gradient respectively. We learned from experiments that the selection of gradient quantizer size could have effect on the compression performance. The adaptive arithmetic coder does improve compression ratio in this work, however, the improvement is sensitive to the number of symbols to be coded: when the number of symbols is very large the improvement is limited.

Table 1. Compression results of testing images in bits/pixel

Testing Image	Context Clustering	CALIC	LOCO	BTPC
Camera	4.34	4.24	4.31	4.94
Bridge	5.76	5.70	5.79	6.36
Missa001	3.50	3.43	3.45	3.99
Missa002	3.49	3.43	3.45	3.98
Lena	4.64	4.50	4.60	4.81
Boats	3.95	3.86	3.94	4.42

7 Conclusions

Context clustering is shown to be an effective alternative for lossless image compression in this work. By replacing the conditional *PDF* vector with the reference *PDF* vector, context clustering solves the storage problem of the *PDF* vectors. Therefore an adaptive arithmetic coder can be directly used to encode the corrected error pixels, however, its coding performance depends much on the number of the symbols to be coded. A variable quantizer size for each gradient should be more reasonable than a fixed size such as 9. We found that when the quantizer size of vertical gradient is equal to the sum of the quantizer sizes of two horizontal gradients, our context clustering method achieves a better performance.

Context clustering still suffers a bottleneck between the entropy of arithmetic coding and storage of *PDF* vectors. Usage of more *PDF* vectors in arithmetic coder is best way to reduce the coding entropy, however, the storage of *PDF* vectors will increase sharply. The further research will concentrate on improving the existing methods in the proposed context clustering algorithm. For example, an optimal division number d in equation (2) needs to be estimated to improve compression performance. Selection of the optimal number of clustered contexts in *PDF* vector space should be investigated in future. Furthermore, the scalar context quantizations in this work will be replaced by the vector quantization in context gradient space, which needs to be solved on the fly.

References

1. Howard, P., Vitter, J.: Fast and Efficient Lossless Image Compression. In: Storer, J., Cohn, M. (eds): Proceedings of the IEEE Data Compression Conference, Snowbird, Utah, March 30 – April 1, 1993. IEEE Computer Society Press (1993) 351–360
2. Weinberger M., Seroussi G., Sapiro G.: A Low Complexity, Context-Based, Lossless Image Compression Algorithm. In: Storer, J., Cohn, M. (eds): Proceedings of the 6th Data Compression Conference, Snowbird, Utah, March 31 – April 3, 1996. IEEE Computer Society Press (1996) 140–149

3. Weinberger M., Seroussi G., Sapiro G.: The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS. IEEE Trans. Image Processing. 8 (2000) 1309–1324
4. Wu, X., Memon, N.: Context-based, Adaptive, Lossless Image Codec. IEEE Trans. Communications, Vol. 45. 4 (1997) 437–444
5. Ekstrand, N.: Lossless Compression of Grayscale Images via Context Tree Weighting. In: Storer, J., Cohn, M. (eds): Proceedings of the 6th Data Compression Conference, Snowbird, Utah, March 31 - April 3, 1996. IEEE Computer Society Press (1996) 132–139
6. Greene, D.H., Yao F., Zhang T.: A Linear Algorithm for Optimal Context Clustering with Application to Bi-level Image Coding. In: Proceedings of the 1998 IEEE Int. Conf. on Image Processing, Chicago, Illinois, October 4–7, 1998. IEEE Computer Society (1998) 508-511
7. Ortega, A., Vetterli, M.: Adaptive Scalar Quantization Without Side Information. IEEE Trans. Image Processing. 5 (1997) 665–676
8. Wu, X.: Lossless Compression of Continuous-tone Images via Context Selection, Quantization, and Modeling. IEEE Trans. Image Processing. 6 (1997) 656–664