FAST DYNAMIC QUANTIZATION ALGORITHM FOR VECTOR MAP COMPRESSION

Minjie Chen[1], *Mantao Xu*[2], *Pasi Fränti*[1]

[1]School of Computing, University of Eastern Finland
[2]School of Electrical Engineering, Shanghai Dianji University, China

ABSTRACT

Vector map compression can be solved by incorporating both data reduction (polygonal approximation) and quantization of the prediction errors, which is the so-called dynamic quantization. This straightforward solution is to calculate all the rate-distortion curves with respect to each of the quantization levels such that the best quantizer is the lower envelope of the set of curves. But computing an entire set of rate-distortion curves is computationally expensive. To solve this problem, we propose a fast algorithm first estimates an optimal Lagrangian parameter λ for each given quantization level *l* and thus only one rate-distortion curve is achievable for constructing the optimal quantizer of prediction errors. An experimental result demonstrates that proposed algorithm reduces the computational complexity significantly without compromising its rate-distortion performance.

Index Terms— Data compression, Computational geometry

1. INTRODUCTION

Vector maps embrace a number of geographic information or objects such as waypoints, routes and areas. Those geographic objects can be represented with a sequence of points in a given coordinate system. However, encoding and achieving the geographic objects in a map image may require expensive data storage and processing time. In order to reduce this computational cost, a variety of algorithms has been studied and developed [1-8]. Existing algorithms have been explored via two classes of strategies: *polygonal approximation* and *quantization*.

The main advantage of polygonal approximations is the high compression rates, which can be achievable either by the fast heuristic methods in [9, 10] or by the graph-based methods in [11, 12]. The number of points in the vector map is reduced by *polygonal approximations* such that the polygonal curve can be represented in a coarser resolution. But they quite often incur a high image distortion. On the other hand, the quantization-based approaches calculate the differential coordinates of adjacent data points as the prediction error and then the residual vectors are quantized using different quantization strategies including *product uniform quantization* [2], *product scalar quantization* [3], and *vector quantization* with fixed-size codebook [4]. However, they often lead to less distortion error with the limited compression gains. A pioneer solution is to combine both the advantage of polygonal approximations and prediction error quantization to achieve the best rate-distortion performance. For instance, the previous *reference line* method [5] first identified a series of references lines by using polygonal approximation, prediction errors are then estimated for the remaining points according to their nearest reference lines followed by *product scalar quantization* in a similar manner to [3]. Likewise in [8], a number of data points were first reduced by *Visvalingam-Whyatt* algorithm, to preserve a consistent topology, and then were quantized and encoded by a clustering-based method.

Motivated by the previous progress made by polygonal approximation and polygonal quantization, a so-called *dynamic quantization* (DQ) in [6] was sincerely investigated. The *dynamic quantization* algorithm performs a joint optimization using both polygonal approximation and vector quantization via *dynamic programming*. For a given quantization level *l*, a naive *product uniform quantization* is employed in the joint optimization using a *Lagrangian* parameter *λ*. Traversing different *Lagrangian* parameter *λ* will construct a rate-distortion curve that depends on the quantization level *l*. The optimal solution is selected according to the *lower envelope* of these curves. However, a main challenge for the joint optimization is its expensive computational cost. To overcome this difficulty, the *error balance principle* was proposed in [7] based on a strict assumption that the total quantization error equals to the error for *polygonal approximation* without quantization. For a given quantization level *l*, an optimal number of points *M* can be identified in the *min-ε polygonal approximation* problem by using binary search. However, in practice, the time complexity for *min-ε polygonal approximation*s equals to $O(N^2)$ [12] as well, which is still intractable for real-time application.

In this work, we have proposed a fast algorithm for vector map compression. For a given quantization level *l*, an optimal *Lagrangian* parameter *λ* was first estimated and the vector map is compressed by solving a shortest path problem in a directed acyclic graph with cost function $J = E + \lambda R$, where *E* is the distortion for the approximation curve and R is the coding cost. Moreover, the algorithm is further

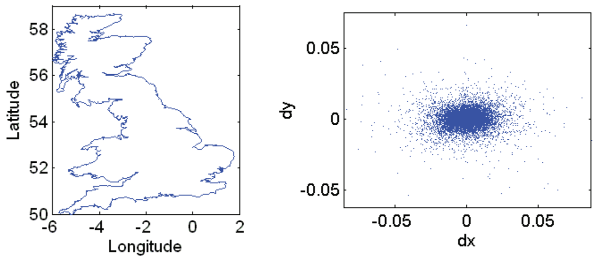

Fig. 1, *Test map Britain with 10,910 points* (*left*), *and corresponding prediction residuals* (*right*).

improved by using a specific search criterion. The structure of this presentation is organized as follows: section 2 describes the improved *dynamic quantization* algorithm; the experimental results are reported in section 3 and finally a conclusion is drawn in section 4.

2. PROPOSED METHOD

2.1 Prediction and encoding

Vector map compression can be formulated as a data compression problem for a 2-dimensional vector sequence $P = (p_1, p_2, \ldots, p_n)$. A common practice for data compression encompasses the three essential procedures: *prediction*, *quantization* of the residual vectors and *entropy coding*. The prediction procedure calculates the differential coordinates of adjacent points as a prediction error instead of using the absolute coordinates for data quantization. It can be assumed that the prediction errors obey a distribution of random variable empirically, e.g., uniform distribution or geometric distribution. An example of polygonal curve can be observed in fig. 1, where the resulting differential coordinates obeys a geometric distribution.

To avoid quantization error propagations, the prediction must be done in a way of closed-loop prediction:

$$p_i^r = Q(\mathbf{v}_i) + p_{i-1}^r \tag{1}$$

where Q is a two-dimensional *product uniform quantizer* $\mathbf{v}_i$ is the residual vector and p_{i-1}^r is the estimation of the previous point. For a given quantization level l, the *product uniform quantizer* is formulated as:

$$Q(\mathbf{v}_i) = [\mathbf{v}_i / l] \cdot l = ([\Delta x_i / l] \cdot l, [\Delta y_i / l] \cdot l) \tag{2}$$

Obviously, coding Q ($\mathbf{v}_i$) is equivalent to coding an integer vector $\mathbf{q}$ = ([$\Delta x_i/l$], $\Delta y_i/l$]), which can be encoded by probability distributions of q_x and q_y.

$$r(\mathbf{v}_i) = -\log_2 f(q_{xi}) - \log_2 f(q_{yi}) \tag{3}$$

where the codebook itself must be also encoded and transmitted to the decoder. But a large-sized codebook is intractable in order to achieve a desirable coding efficiency. An intuitive solution is to adopt a single-parameter *geometric distribution* to model $|q_x|$ and $|q_y|$:

$$f(|q_x|) = (1 - p_x)^{|q_x|} p_x \tag{4}$$

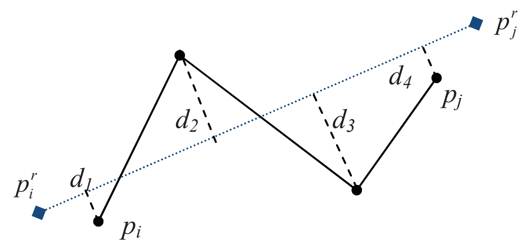

Fig. 2, *Poly-line* $\{p_i, \ldots, p_j\}$ *(solid line) is approximated by* $\{p_i^r, p_j^r\}$ *(dot line)with approximating error* $e_2(p_i^r, p_j^r) = d_1^2 + d_2^2 + d_3^2 + d_4^2$

where p_x can be approximated by using *maximum likelihood estimation*. Thus, the code length led by an arithmetic coding according to the *geometric distribution* is written as

$$\begin{aligned} r(\mathbf{v}_i) &= -(|q_{xi}| \log_2(1 - p_x) + \log_2(p_x)) + 2 \\ &\quad -(|q_{yi}| \log_2(1 - p_y) + \log_2(p_y)) \end{aligned} \tag{5}$$

where no codebook is needed.

We should mention when the data has different property, *uniform distribution*, *negative binomial distribution* or *Poisson distribution* can also be considered instead of *geometric distribution*.

2.2 Dynamic quantization

In *dynamic quantization, polygonal approximation* is embedded into the closed-loop framework by using dynamic programming. Suppose that a poly-line $\{p_i, \ldots, p_j\}$ is approximated by line segment $\{p_i^r, p_j^r\}$, the approximation error can be defined as the sum of square distances from vertices p_k ($i \le k \le j$) to $\{p_i^r, p_j^r\}$ in fig. 2:

$$e_2(p_i^r, p_j^r) = \sum_{k=i}^{j} d^2(p_k, \{p_i^r, p_j^r\}) \tag{6}$$

This approximation error in (6) can be calculated in a constant time [11] by pre-computing the accumulated sum for curve coordinates x^2, x, xy, y^2 and y. The *dynamic quantization* becomes a joint optimization of polygonal approximation and prediction error quantization, which minimizes the cost function:

$$J = E_2 + \lambda R = \sum_{m=1}^{M} (e_2(p_{i_m}^r, p_{i_{m+1}}^r) + \lambda \cdot r(p_{i_m}^r, p_{i_{m+1}}^r)) \tag{7}$$

where M is the number of points output by polygonal approximation. The minimization problem can be solved by the shortest path search on a weighted directed acyclic graph (DAG) or *dynamic programming*. Suppose J_i is the minimum weighting sum from p_1 to p_i on G, A is an array used for backtracking operation, the recursive equation can be defined by:

$$J_i = \min_{\{1 \le k \le i-1\}} (J_k + e_2(p_k^r, p_i^r) + \lambda r(p_k^r, p_i^r)), J_1 = 0 \tag{8}$$

$$A_i = \arg\min_{\{1 \le k \le i-1\}} (J_k + e_2(p_k^r, p_i^r) + \lambda r(p_k^r, p_i^r)) \tag{9}$$

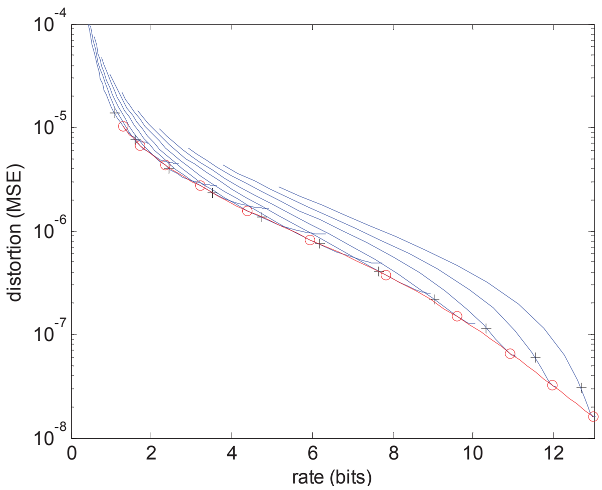

Fig. 3, *Rate-distortion curve for quantization step q_k=0.01/2^k, where k=0, 1/2,1,…, 5 (from left to right), black '+' is the position when error balance principle is applied, red 'o' is the proposed(λ is selected by (10)).The red line is the rate-distortion curve when optimal λ is selected with q_k=0.01/2^k, where k=0, 1/5, 2/5,…, 5.*

Existing approach intuitively calculates all the rate-distortion curves with respect to each of the quantization levels such that the best quantizer is the *lower envelope* of the set of curves. This method computes an entire set of rate-distortion curves which is hugely time-expensive.

To resolve this operational problem, a *fast dynamic quantization (FDQ) algorithm* is proposed. We have proved, for each quantization level l, one optimal *Lagrangian* parameter λ can be estimated as (see appendix):

$$\lambda \approx \frac{1}{6} l^2 \ln 2 \tag{10}$$

Eventually, only one dynamic quantization needs to be conducted for a given quantization level l. However, by traversing different quantization level l, a unique rate-distortion curve can be constructed. An example can be found in Fig. 3. In the figure, it was also illustrated that the pervious methods have investigated different quantization levels by considering 30-40 number of λs for each l, which leads to 300 iterations of minimization of (7).

2.3 Search criterion

The shortest path algorithm on a weighted DAG takes $O(N^2)$ time. This can further be improved by incorporating a specific search criterion:

$$\sqrt{\frac{e_2(p_k^r, p_i^r)}{(i-k)}} > \tau \sqrt{\frac{e_2(p_{A_i}^r, p_i^r)}{(i-A_i)}} \tag{11}$$

where { $p_{A_i}^r$, p_i^r } has a shortest path so far and p_k^r is the current testing point. Namely, for a target point p_i^r , the shorted path search will terminate weight calculation before point p_k once if equation (11) is satisfied. The experimental testing in this presentation has revealed that the processing time can be reduced by more than 95% with τ =40. Pseudo code of proposed algorithm is shown in Fig. 4.

The proposed method can also be applied for entropy-constrained problem, in which we compress the vector map data under a certain bit-rate. The result can then be obtained by several iterations of the algorithm using bisection search on the quantization level l.

```
INPUT   l                                ← quantization step
        P = (p1, p2,…, pn)               ← 2-D vector sequence
OUTPUT: P' = ( p1^r , p2^r…, pn^r )      ← approximation curve
        F                                ← compression file
FUNCTION VectorMapEncoding (P, l) RETURN P' ,F
N       ←    Number of point for curve P;
J       ←    array with N elements, J(1)=0, J(2:N)=∞
λ       ←    1/6l²ln2
p1^r    ←    p1
A       ←    backtracking array with N elements
FOR i = 1 TO N
    FOR k = i-1 TO 1
        ptmp← pk^r + Q(pi - pk^r )
        Jtmp ← Jk + e2(pk^r , ptmp) + λr(pk^r , ptmp)
        IF Jtmp < Ji
                Ji      ← Jtmp
                Ai      ← k
                pi^r    ← ptmp
        ELSEIF (11) met
                BREAK;  // Stop searching criterion
        END IF
    END FOR
END FOR
P'      ←    Backtracking on A and do sub-sampling on P'
F       ←    Encoding on P'  using arithmetic coding by (5)
```

Fig. 4 *Pseudo code of fast dynamic quantization method*

3. EXPERIMENTS

The proposed *fast dynamic quantization* algorithm (FDQ) is evaluated on a 10,910-point vector map representing the contour of Britain (Fig. 1). We compare the performance with the previous *dynamic quantization* (DQ) algorithm [7], and with several other approaches as well: *clustering-based method* (CBC) [4], and *reference line method* (RL) [5]. The distortion is measured here by mean squared error (MSE). The corresponding rate-distortion curves are plotted in Fig. 5. It can be observed that the proposed algorithm achieves significantly better rate-distortion result than the other approaches in this work considered and can be also comparable with the DQ algorithm. The computational cost for solving the entropy-constrained problem can be reduced within 1 second by using the proposed dynamic quantization. In the experiments, the proposed algorithm only takes 5% of the time as the previous approaches do. The proposed algorithm is also applicable to variable-resolution compression problem for a real-time application. The visualization performance in the decoder for different compression bit-rate can be found in fig. 6.

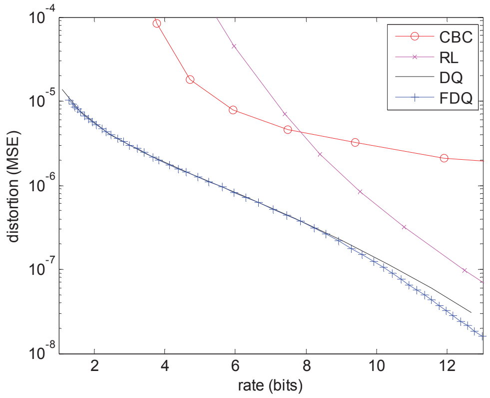

Fig. 5, *Performance comparison*

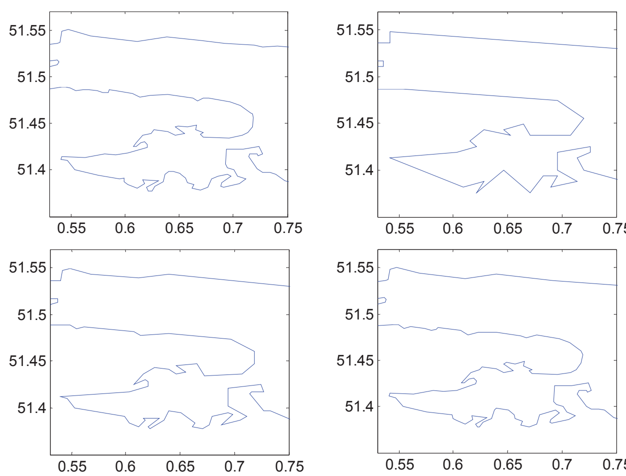

Fig. 6, *Performance under different bit-rate on a fragment of the test curve. Original 128 bits/point (top-left), 2 bits/point (top-right),5 bits/point (bottom-left),10 bits/point(bottom-right)*

4. CONCLUSION

We have proposed a fast *dynamic quantization* algorithm for lossy compression of vector map. The underlying algorithm first identified an optimal *Lagrangian* multiplier λ value for each quantization step *l* and then constructed only one rate-distortion curve for design of predicted-error quantizer. In addition, a powerful searching criterion was exploited for the sake of speeding up the dynamic quantization.

Experimental results have shown that the proposed method is twenty times faster than the previous dynamic quantization algorithm but achieves a similar or better compression performance. Future work can be considered in the following perspectives:

1. The dynamic quantization can be improved by combining vector quantization and uniform product quantization.
2. Lattice VQ can be used instead of uniform quantization.
3. Linear prediction can be considered to improve the prediction of the residual vectors.

5. REFERENCES

[1] G. M. Schuster and A. K. Katsaggelos, "An optimal polygonal boundary encoding scheme in the rate-distortion sense", *IEEE Trans. on Image Processing*, vol.7, pp. 13-26, 1998.

[2] Z. Li, and S. Oppenshaw, "A natural principle for the objective generalization of digital maps", *Cartography and Geographical Information Systems*, vol. 20, pp. 19-29, 1993.

[3] A. Akimov, A. Kolesnikov and P. Fränti, "Coordinate quantization in vector map compression", *IASTED Conference on Visualization, Imaging and Image Processing* (*VIIP'04*), pp. 748-753, 2004.

[4] S. Shekhar, S. Huang, Y. Djugash, J. Zhou, "Vector map compression: a clustering approach", *10th ACM Int. Symp. Advances in Geographic Inform*, pp.74-80, 2002.

[5] A. Akimov, A. Kolesnikov and P. Fränti, "Reference line approach for vector data compression", *IEEE Int. Conf. on Image Processing (ICIP'04)*, vol. 2, pp. 1891-1894, 2004.

[6] A. Kolesnikov, "Optimal encoding of vector data with polygonal approximation and vertex quantization", *SCIA'05*, LNCS, vol. 3540, 1186–1195. 2005.

[7] A. Kolesnikov, "Optimal algorithm for lossy vector data compression", *Int. Conf. on Image Analysis and Recognition* (ICIAR'07), LNCS 4633, pp. 761-771, 2007.

[8] B. Yang and Q. Li, "Efficient compression of vector data map based on a clustering model", *Geo-spatial Information Science*, 12(1), 13-17, 2009.

[9] A. Masood, "Optimized polygonal approximation by dominant point deletion", P*attern Recognition,* 41 (1), 227-239, 2008.

[10] P. Bhowmick and B. Bhattacharya, "Fast polygonal approximation of digital curves using relaxed straightness properties", *IEEE Trans. on PAMI,* 29 (9), 1590-1602, 2007.

[11] J. C. Perez, E. Vidal, "Optimum polygonal approximation of digitized curves", *Pattern Recognition Letters,* 15, 743–750, 1994.

[12] M. Salotti, "Optimal polygonal approximation of digitized curves using the sum of square deviations criterion", *Pattern Recognition* 35, 435–443, 2002.

APPENDIX: PROOF OF (10)

The cost function is defined as: $J= E + \lambda R$. In uniform product quantization, for a given quantization step *l*, mean square error *E* can be calculated by:

$$\int_0^{l/2}\frac{1}{l}x^2dx+\int_{l/2}^{l}\frac{1}{l}(l-x)^2=\frac{1}{2}E$$

where $E=l^2/6$. For residual vector $\boldsymbol{v}_i$, after uniform product quantization, if it is estimated by *geometric distribution*,

$$f(|q_x|)=(1-p_x)^{|q_x|}p_x, f(|q_y|)=(1-p_y)^{|q_y|}p_y$$

The coding length can be approximated as:

$$r(\boldsymbol{v}_i)=-(\log_2((1-p_x)^{|q_{xi}|}\cdot(p_x))+-(\log_2(1-p_y)^{|q_{yi}|}\cdot p_y)+2$$

The maximum likelihood estimation of *geometric distribution* is:

$$p_x=1/(\frac{1}{n}\sum_{i=1}^{n}|q_{xi}|+1)\approx\frac{1}{\overline{|\Delta x_i|}/l+1}=\frac{\overline{|\Delta x_i|}}{\overline{|\Delta x_i|}+l}$$

$$\therefore\ R=\frac{1}{n}\sum_{i=1}^{n}r(\boldsymbol{v}_i)\approx-(\frac{\overline{|\Delta x_i|}}{l}\log_2\frac{\overline{|\Delta x_i|}}{\overline{|\Delta x_i|}+l}+\log_2\frac{l}{\overline{|\Delta x_i|}+l})+$$

$$-(\frac{\overline{|\Delta y_i|}}{l}\log_2\frac{\overline{|\Delta y_i|}}{\overline{|\Delta y_i|}+l}+\log_2\frac{l}{\overline{|\Delta y_i|}+l})+2$$

Define $r_x=\overline{|\Delta x_i|}/l, r_y=\overline{|\Delta y_i|}/l$,

$$R\approx-(r_x\log_2\frac{r_x}{r_x+1}+\log_2\frac{1}{r_x+1})+-(r_y\log_2\frac{r_y}{r_y+1}+\log_2\frac{1}{r_y+1}))+2$$

As $r_x, r_y >> 1, r_x\cdot\log_2\frac{r_x}{r_x+1}\approx\frac{1}{\ln 2}, \log_2\frac{1}{r_x+1}\approx\log_2\frac{1}{r_x}=-\log_2 r_x$,

$$\therefore\ R\approx-(1/\ln 2-\log_2 r_x)+-(1/\ln 2-\log_2 r_y))+2$$

$$\approx-(1/\ln 2-\log_2\overline{|\Delta x_i|}+\log_2 l)+-(1/\ln 2-\log_2\overline{|\Delta y_i|}+\log_2 l))+2$$

$$\partial J_0/\partial l\approx\frac{l}{3}-\frac{\lambda}{l\ln 2}-\frac{\lambda}{l\ln 2}\approx\frac{l}{3}-\frac{2\lambda}{l\ln 2}$$

By setting $\partial J_0/\partial l=0$, We got $\lambda\approx 1/6\cdot l^2\ln 2$ $\square$