Vector Quantization

Hai Zhu

HPC 12

Quantization

- Definition:
  - Quantization: a process of representing a large – possibly infinite – set of values with a much smaller set.
  - Widely Used in Lossy Compression
  - Represent certain image components with fewer bits (compression)
  - With unavoidable distortions (lossy)
- Design:
  - Find the best tradeoff between

maximal compression ←→minimal distortion

Scalar quantization

- A mapping of an input value x into a finite number of output values, y:

$$Q: x \rightarrow y$$

- An example: any real number x can be rounded off to the nearest integer, say

$$q(x) = round(x)$$

Vector Quantization (VQ)

- Vector Quantization is used in many applications such as data compression, data correction, and pattern recognition.

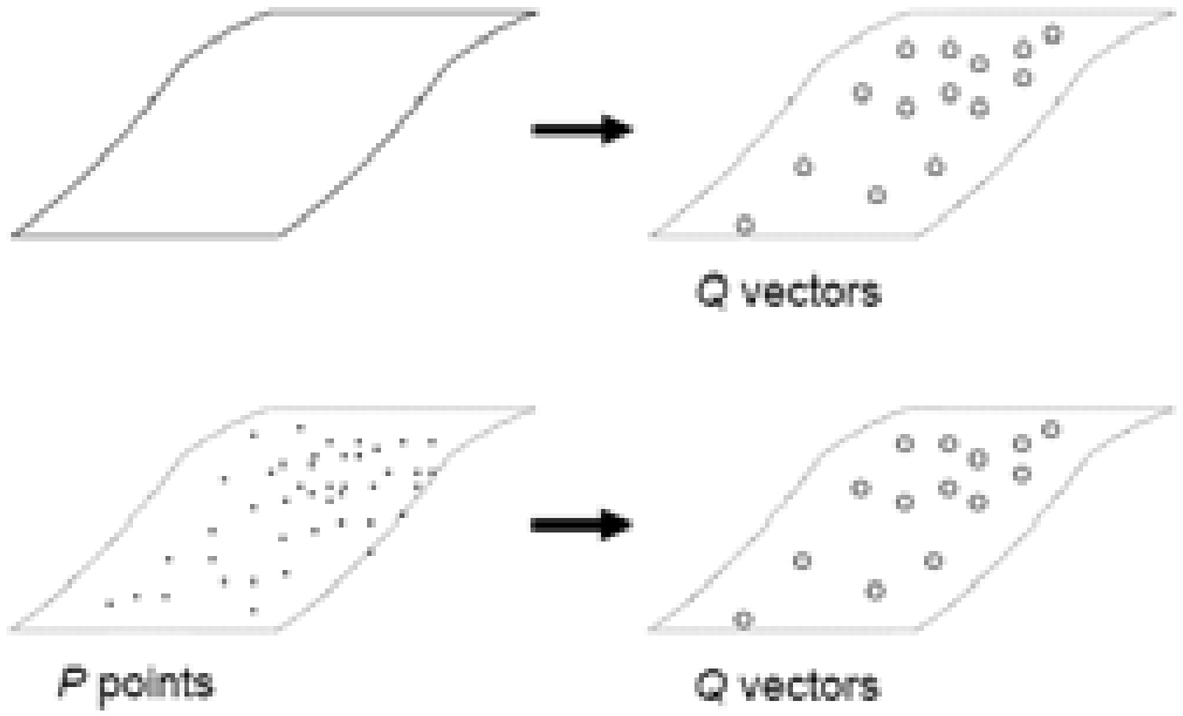
- Vector quantization is a lossy data compression method.

- It works by dividing a large set of vectors into groups having approximately the same number of points closest to them.

- Each group is represented by its centroid point, as in k-means and some other clustering algorithms.

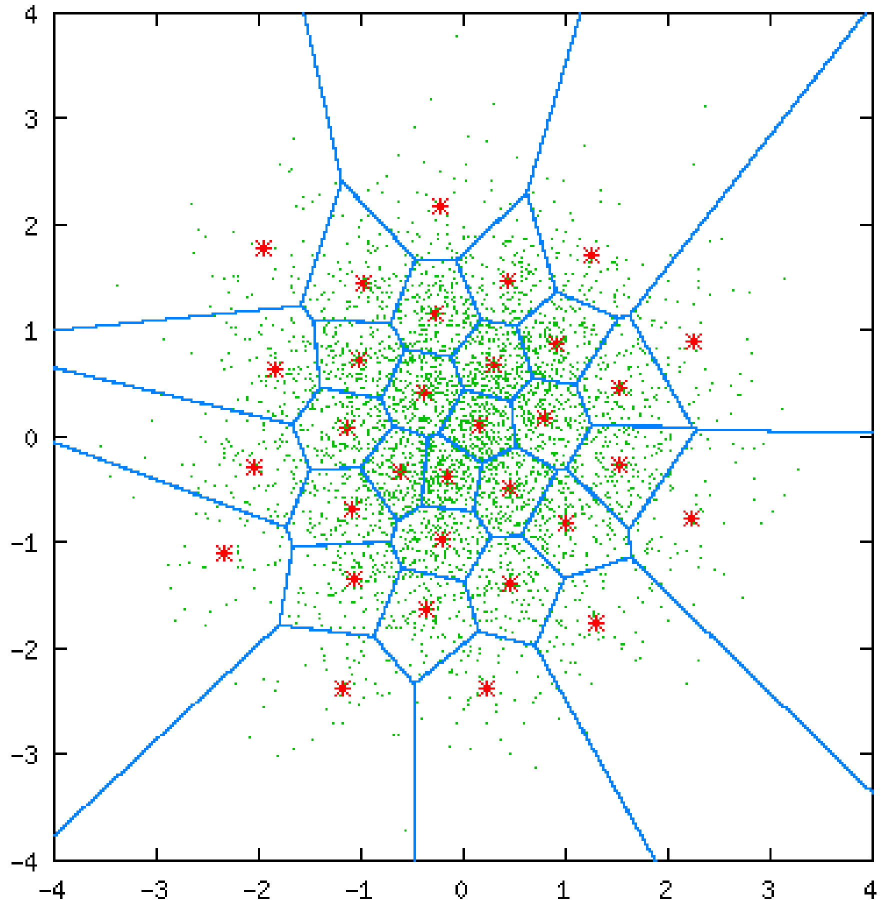
Vector Quantization (VQ)

- To project a continuous input space on a discrete output space, while minimizing the loss of information



Vector Quantization (VQ)

- To define zones in the space, the set of points contained in each zone being projected on a representative vector (**centroid**)
- Example: 2-dimensional spaces

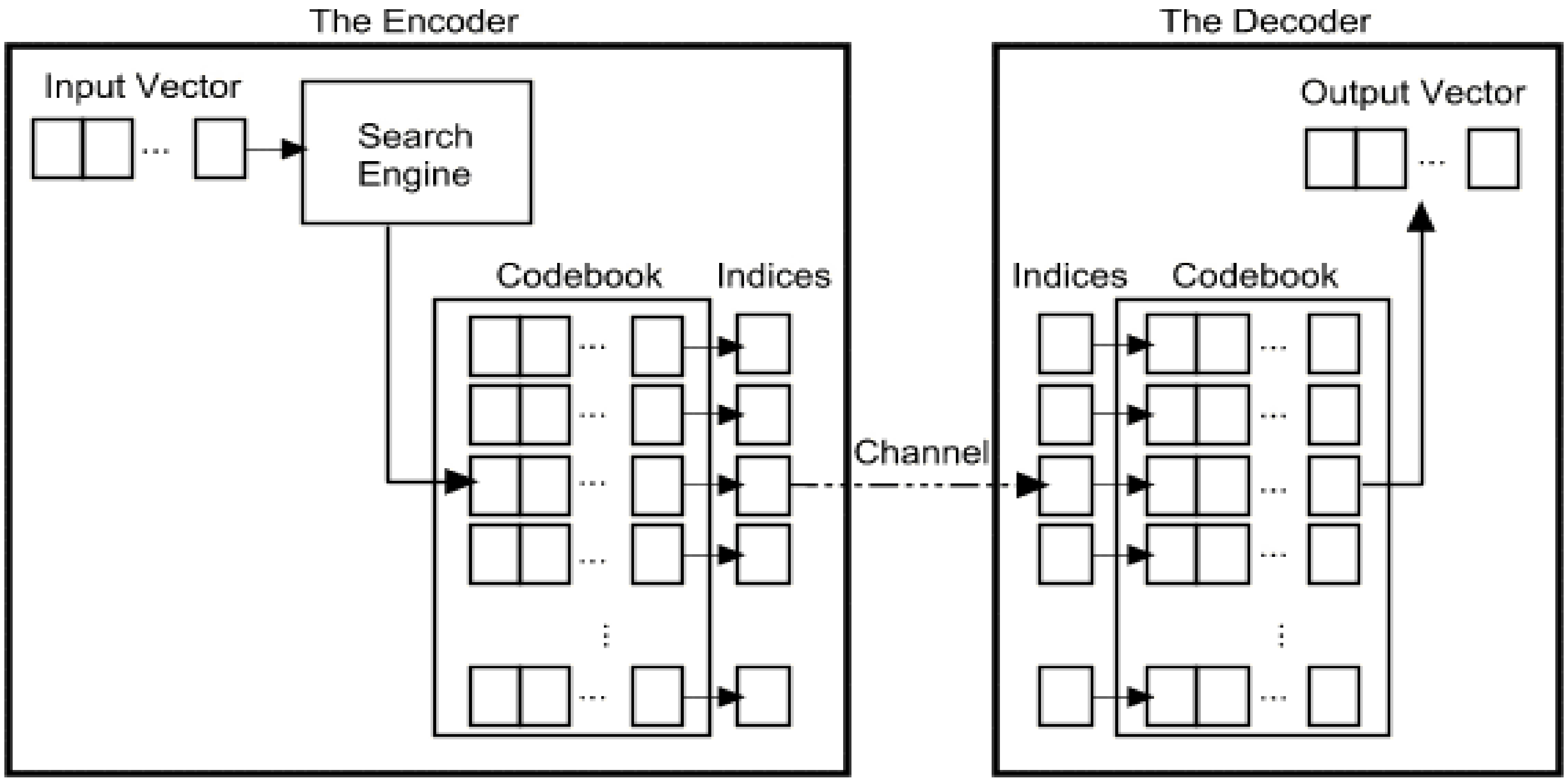


Vector Quantization (VQ)

- Map k-dimensional vectors in the vector space R^k into a finite set of vector

$$Y = \{yi : i = 1, 2, \ldots, N\}$$

Vector Quantization (VQ)

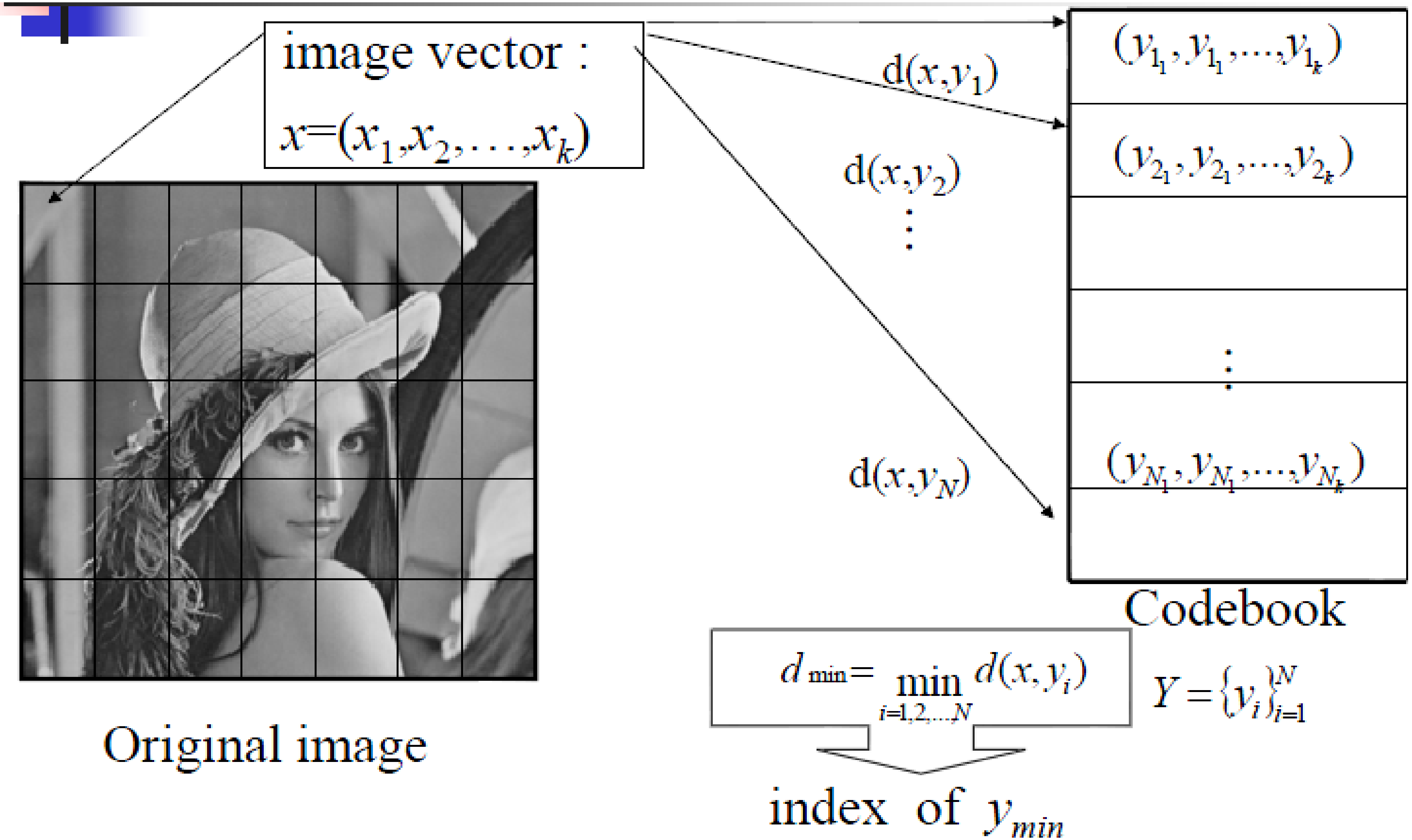


Terms

- Codebook
  - In cryptography, a codebook is a document used for implementing a code
  - A codebook contains a lookup table for encoding and decoding; each word or phrase has one or more strings which replace it
- Codeword / Codevector
  - A codeword is an element of a code

VQ for Image Compression

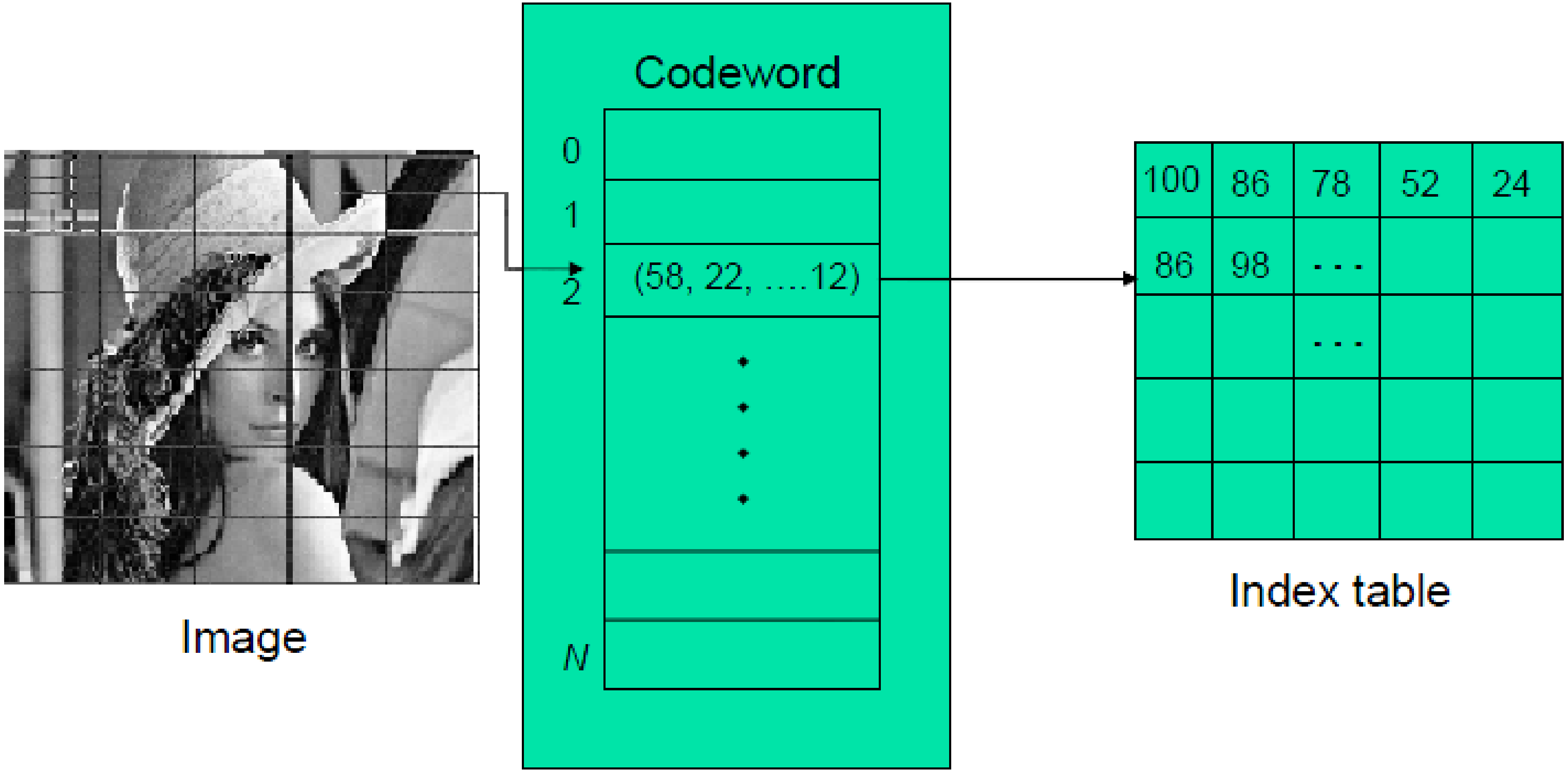
Encoder



VQ for Image Compression

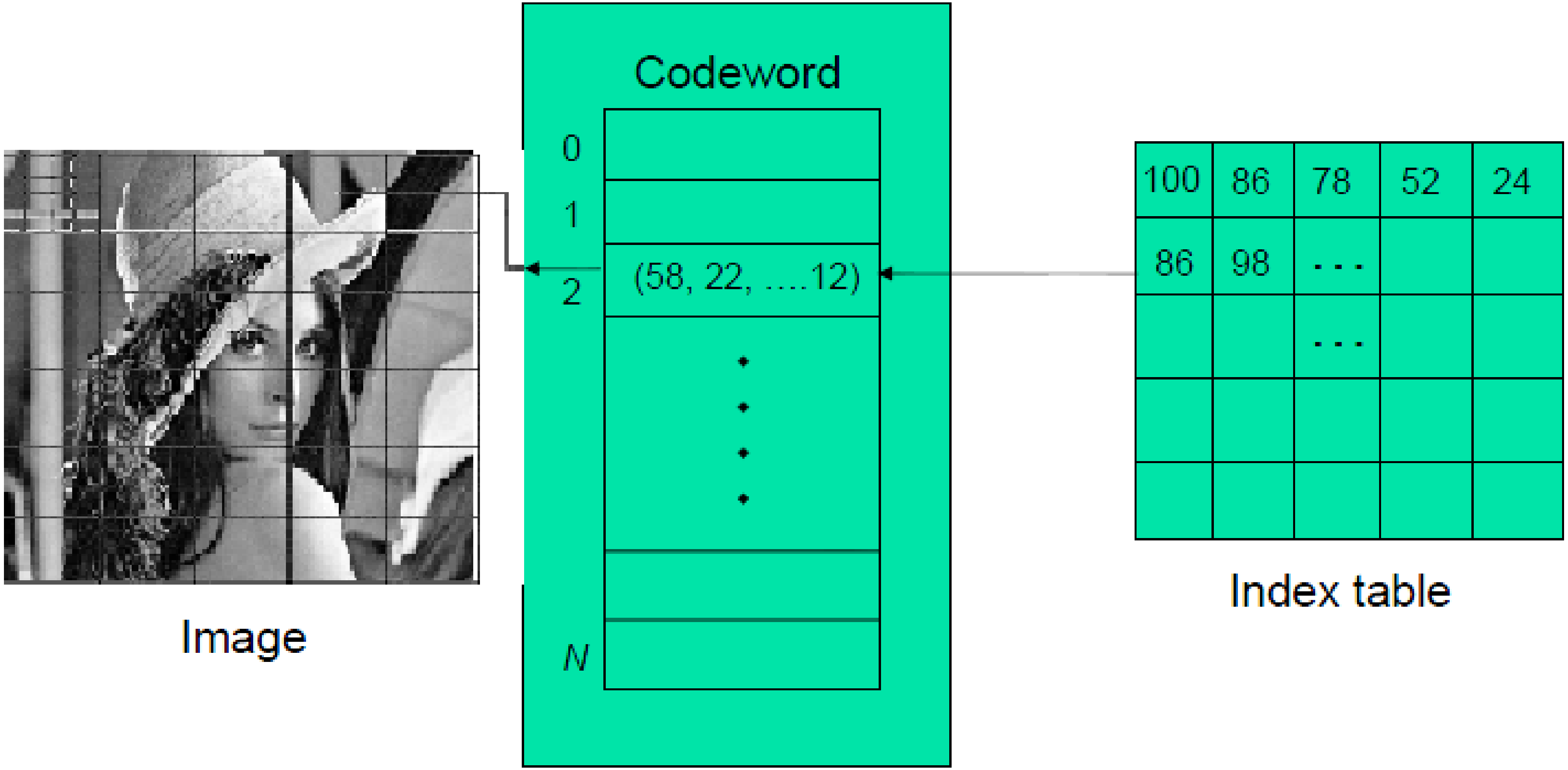
Encoder

$d(x, yi) = \|x - yi\|^2 = \sum_{j=1}^{k}(x_j - yij)^2$



VQ for Image Compression

Decoder



Codebook generation

- In the earlier days, the design of a vector quantizer (VQ) is considered to be a challenging problem due to the need for multi-dimensional integration

- Linde-Buzo-Grey (LBG) algorithm

  In 1980, Linde, Buzo, and Gray (LBG) proposed a VQ design algorithm based on a training sequence

- The use of a training sequence bypasses the need for multi-dimensional integration

LBG Algorithm

- Determine the number of codevectors N
- Select N codevectors at random to be the initial codebook
- Using the average distortion measure cluster the vectors around each codevector
- Compute the new set of codevectors (codebook)
- Repeat Steps 2 and 3 until the either the representative codevectors do not change

Design

- A training sequence consisting of M source vectors:

$$T = \{x_1, x_2, \dots, xm\}$$

source vectors are k–dimensional: $\mathrm{xm} = (\mathrm{x_{m1}}, \mathrm{x}_{m2}, \dots, xmk),\ m = 1,2,\dots,M.$

- N is the number of codevectors and codebook is:

$$Y = \{y_1, y_2, \dots, yN\}$$

Each codevector is k-dimensional: $\mathrm{y_n} = (\mathrm{y_{n1}}, \mathrm{y}_{n2}, \dots, yn_k),\ n = 1,2,\dots,N$

- Let S_n be the encoding region associated with codevector:

$$S = \{s_1, s_2, \dots, sN\}$$

- The average distortion is given by:

$$D_{ave} = \frac{1}{M}\sum_{m=1}^{M} \|x_m - ym\|^2$$

The design problem is: Given T and N, find Y and S such that D_{ave} is minimized.

Optimality Criteria

- Y are a solution to the about VQ, then it must satisfied the following two criteria.
- Nearest Neighbor condition:

$$S_n = \{x : \|x - yi\|^2 \leq \|x - yi_{\prime}\|^2 \quad \forall i' = 1, 2, \dots, N\}$$

  Encoding region should consists of all vectors that are closer to than any of the other codevectors.

- Centroid Condition:

$$y_i = \frac{\sum_{x_m \in Sn} x_m}{\sum_{x_m \in Sn} 1} \qquad (i = 1, 2, \dots, N)$$

  Codevector y_i should be average of all those training vectors that are in encoding region .

LBG Algorithm

This initial codebook is obtained by the splitting method.

In this method, an initial codevector is set as the average of the entire training sequence.

1. Given T. Fixed ε>0 to be a "small" number
2. Let N=1 and $y_1^{\ *} = \frac{1}{M}\sum_{m=1}^{M} x_m$, calculate $D_{ave} = \frac{1}{M}\sum_{m=1}^{M} ||x_m - ym||^2$
3. Splitting: For i=1,2,…, N, set

   $y_i^{(0)} = (1+\varepsilon)y_i^{\ *}$, $y_{N+i}^{(0)} = (1-\varepsilon)y_i^{\ *}$ Set N=2N

4. Iteration:

LBG Algorithm

4. Iteration:

   i. For m=1,2,…M find the min($\|x_m - ym\|^2$)

      Let n* be the index which achieves the minimum. Set $Q(x_m) = cn *^{\,i}$

   ii. For n=1,2,…N, update the codevectors (centroid)

$$y_n^{(i\,+\,1)} = \frac{\sum_{Q(x_m)=cni} x_m}{\sum_{Q(x_m)=cni} 1}$$

   i. Set i=i+1.

   ii. Calculate

$$D_{ave} = \frac{1}{M}\sum_{m=1}^{M}\|x_m - ym\|^2$$

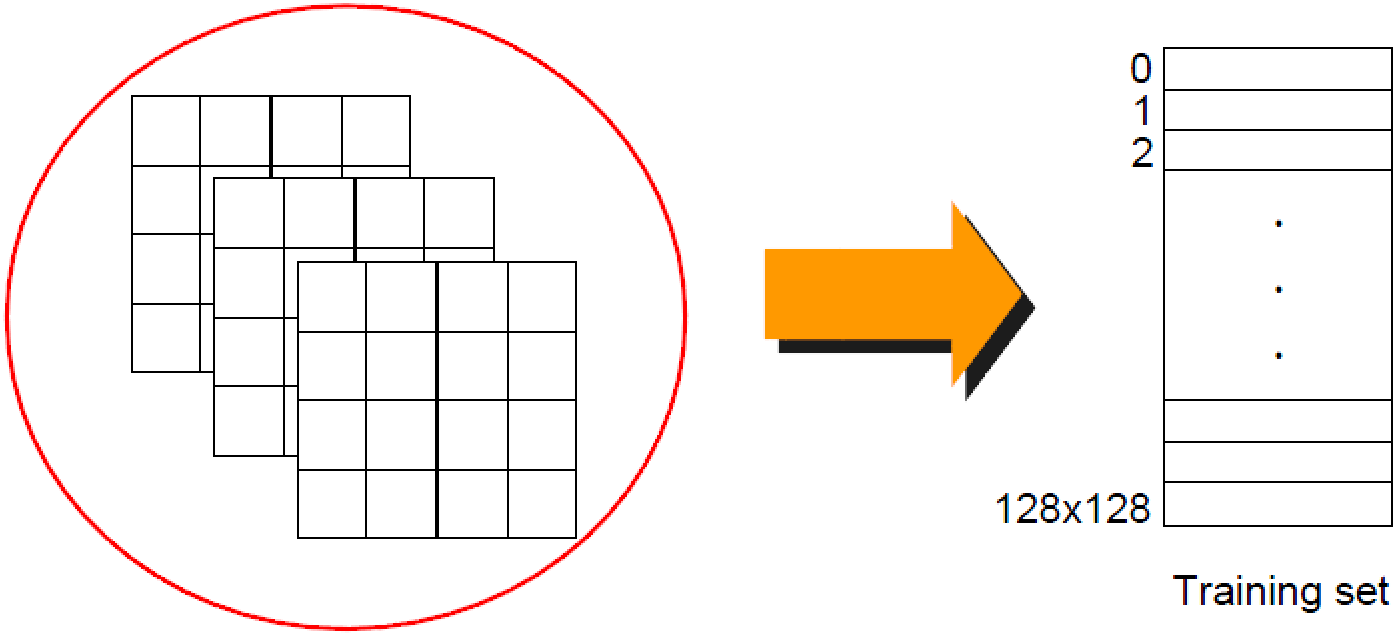
   i. If(D_{ave}(i-1)-Dave(i)/Dave(i-1)> ε, go to Step (i)

   ii. Set $D_{ave}' = D_{ave}^{i.}$ For n=1, 2, ..., N, set $y_n' = y_n^{(i)}$ as the final codevectors.

   iii. Repeat step 3 and 4 until the desired number of codevectors is obtained.

Codebook Training

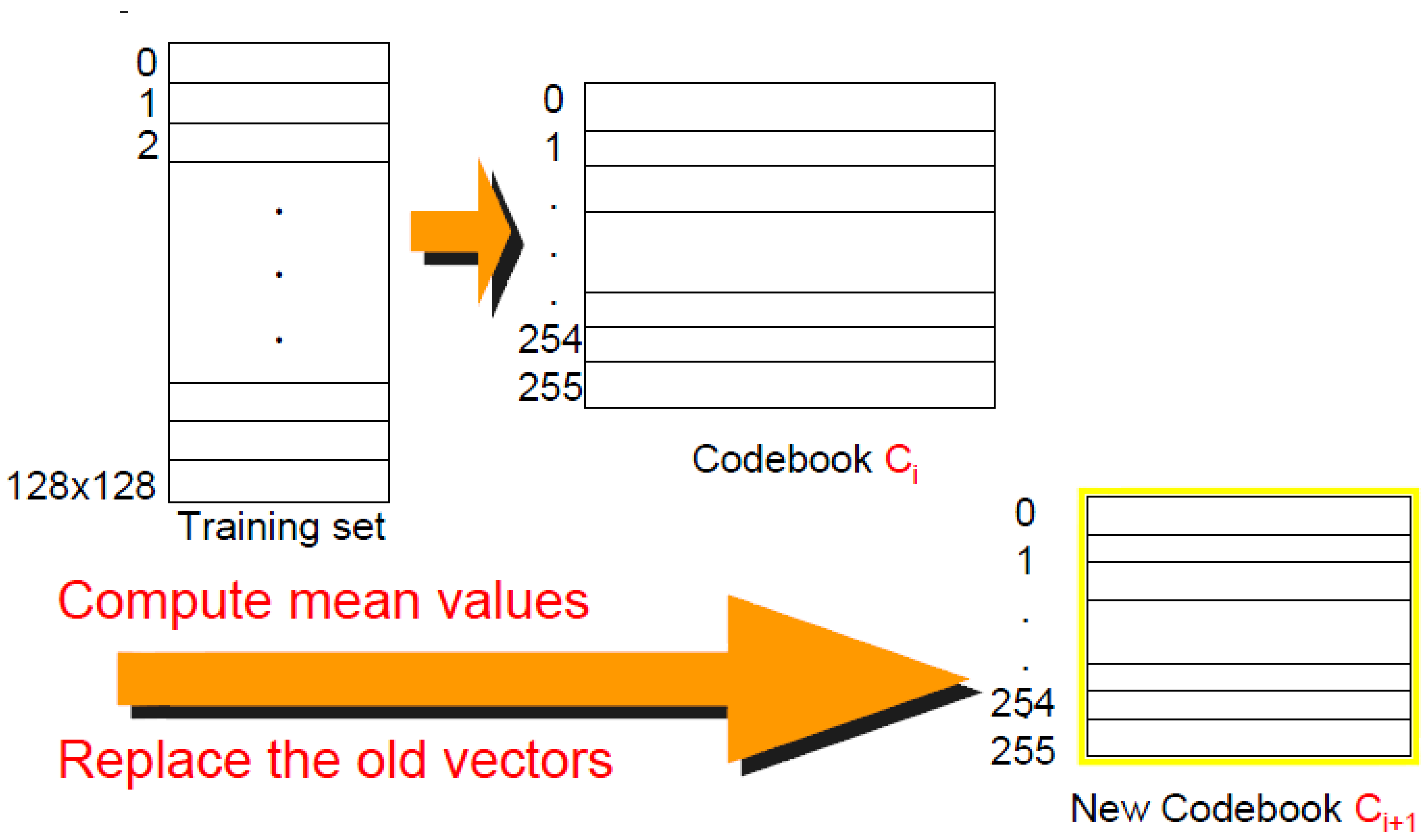
- Image block size is 4*4
- 128*128 vectors for 512*512 image
- Convenience
- Without loss of generality



Training image

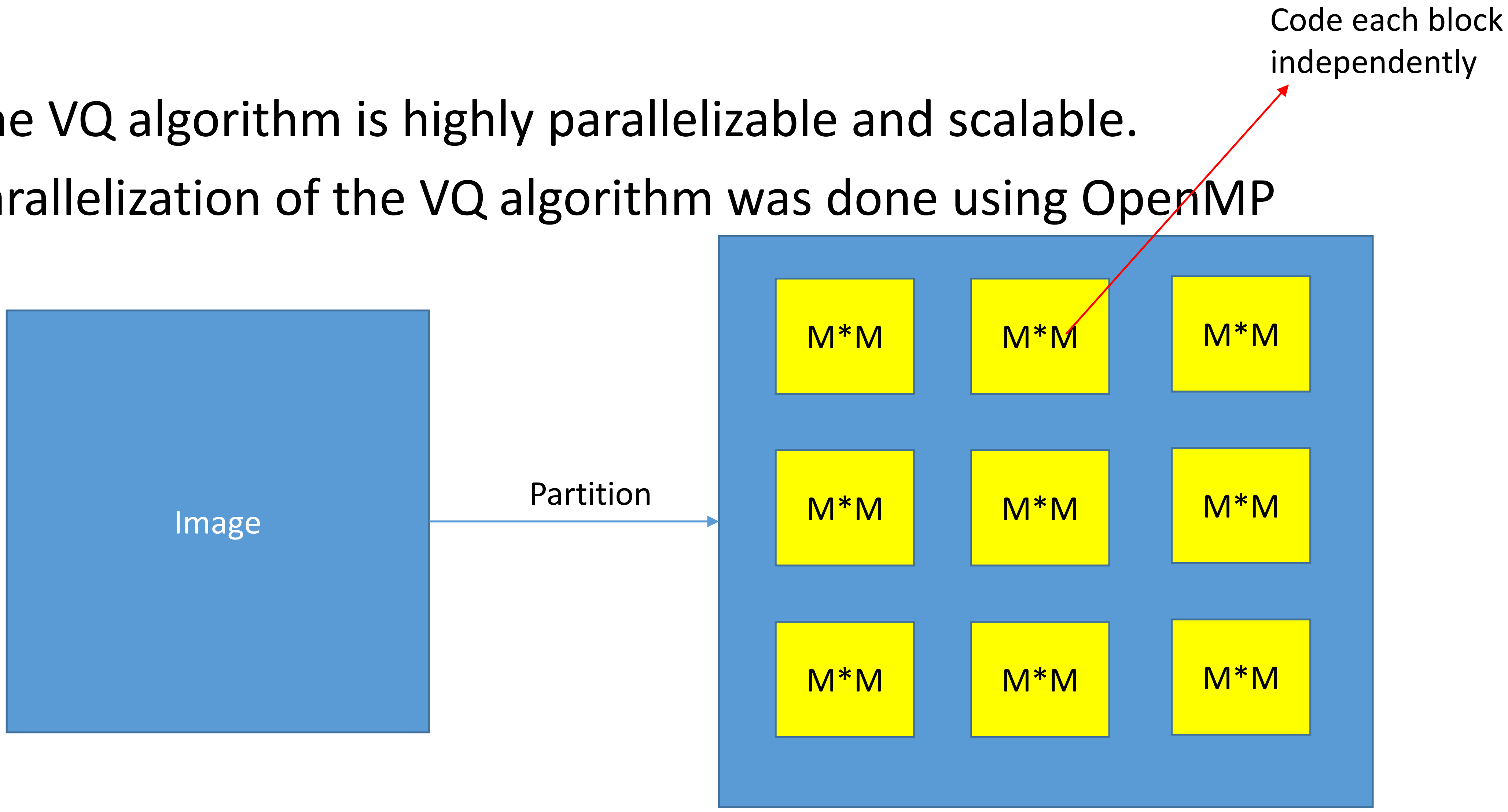
Training set

Codebook Training

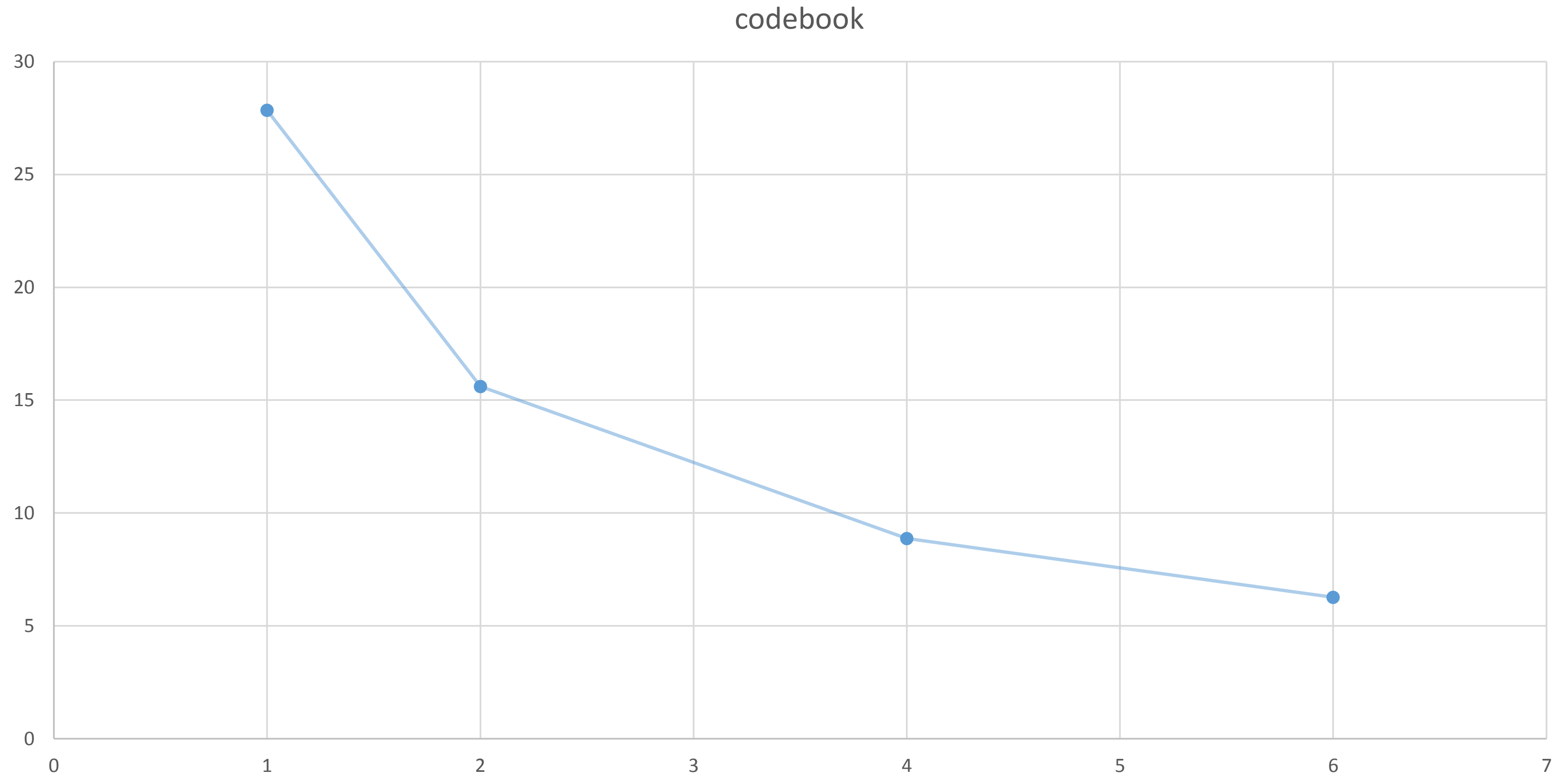


Parallelization of Vector Quantization

- The VQ algorithm is highly parallelizable and scalable.
- Parallelization of the VQ algorithm was done using OpenMP



Result for 512*512 8bit graycolor, 256 codevectors, 4*4 block



Bit rate

- Number of codevectors : N_C
- Input vector dimension: N
- $(\log_2 N_C)/N$ bits/pixel
  - Example: 4×4 blocks, N_C =256, $\log_2 N_C$ =8
  - bit rate=7/(4×4)
- Two process in VQ
  - Codebook generation
  - Speedup search

Result

Original

0.5 bpp