

## A Reversible Data Embedding Scheme Based on Search Order Coding for VQ Index Tables

Authors:

Peyman Rahmani, Gholamhossein Dastghaibyfar, Ehsan Rahmani

Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran

1



## Outlines

- Introduction
- Vector quantization
- Search order coding
- Proposed method
  - Data embedding procedure
  - Data extraction procedure
- Experimental results
- Conclusion

2

## Introduction

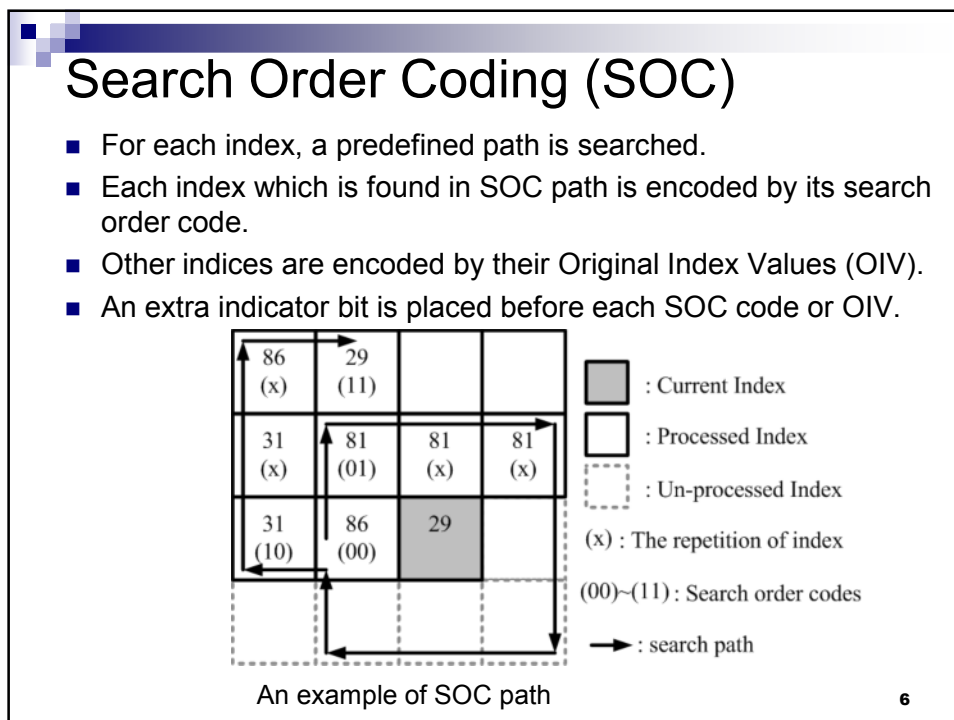
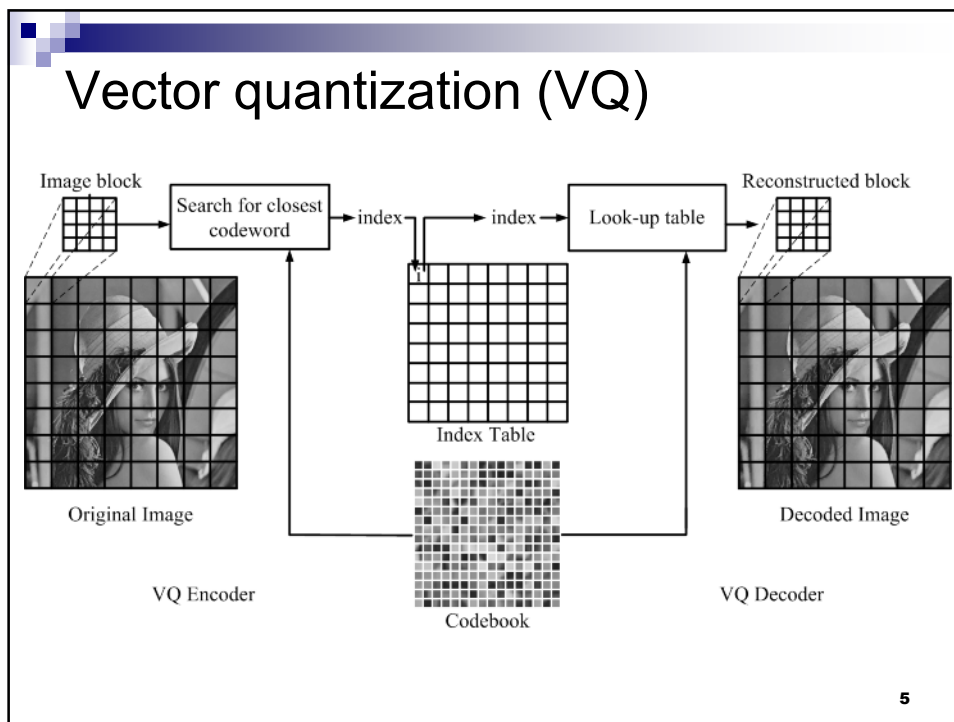
- **Data Hiding**  
Hide secret data into cover media to provide secret and secure transmission
- **Cover objects**  
images, videos, audios, ...
- **Various domain for data hiding in images**
  - Spatial domain
  - Frequency domain
  - Other compression types including vector quantization (VQ)
- **Categorization of data hiding schemes from aspect of reversibility**
  - Irreversible
  - Reversible

3

## Introduction (cont.)

- **VQ based reversible data hiding techniques:**
  - **Non-legitimate output**
    - Output can't be decoded by standard decoders so it may raise suspicion.
    - Such schemes are not suitable for steganographic applications.
  - **Legitimate output**
    - An index replacement strategy is used to embed data.
    - Such schemes generate VQ codes with some artifacts in the image blocks and also some spots may appear in their resulting stego image.

4



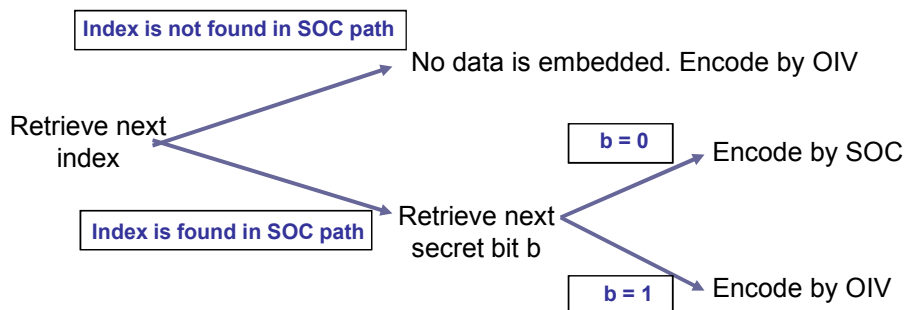
## Proposed method (1/3)

- Proposed scheme embed data into SOC code with reversibility.
- Output code is legitimate SOC code, and embedded code can be decoded by using standard SOC decoder.
- Instead of changing indices, the method to encode each index is used to embed data.
- Data is embedded into indices which can be represented by SOC code or OIV.
- The embedding procedure doesn't degrade image quality, but compression performance is affected.

7

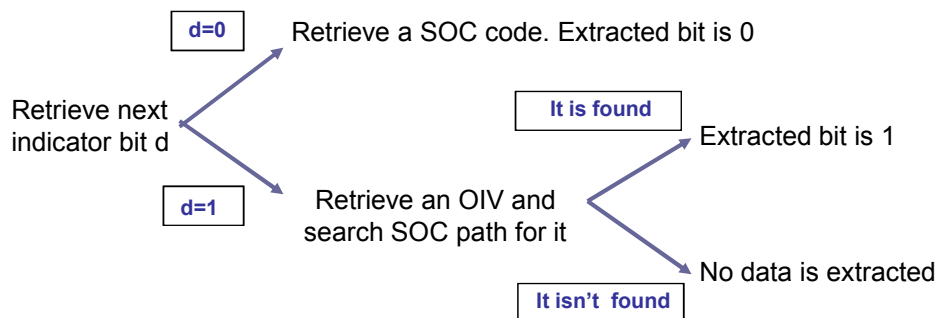
## Proposed method (2/3)

- Data Embedding Procedure:



## Proposed method (3/3)

### Data Extraction Procedure:



## Example (Data Embedding) (1/3)

### Secret bits: ...0 ...1

#### Index table

20	35	75	80	20	18	27	118
01	10	11					
66	227	75	83	80	75	27	120
00							
66	31	31	75	31	20	31	118
15	31	38	86	20	27	120	118

Embedded code:

1 11100011

## Example (Data Embedding) (2/3)

- Secret bits: ...0...1

Index table

20	35 <b>01</b>	75 <b>11</b>	80	20	18	27	118
66	227 <b>00</b>	<b>75</b>	83	80	75	27	120
66	31	31	75	31	20	31	118
15	31	38	86	20	27	120	118

Embedded code:

1 11100011 **0** 10

11

## Example (Data Embedding) (3/3)

- Secret bits: ...0...1

Index table

20	35	75	80	20	18	27	118
66	227	75	83 <b>01</b>	80 <b>10</b>	75 <b>x</b>	27	120
66	31	31 <b>11</b>	75 <b>00</b>	<b>31</b>	20	31	118
15	31	38	86	20	27	120	118

Embedded code:

1 11100011 **0** 10 ... **1** 00011111

12

### Example (Data Extraction) (1/3)

Input code: ... **1** 11100011 **0** 10 ... 1 00011111

Index table

20	35	75	80	20	18	27	118
01	10	11					
66	227						
00	00						

SOC code: 1 11100011

Secret bits:

13

### Example (Data Extraction) (2/3)

Input code: ... 1 11100011 **0** 10 ... 1 00011111

Index table

20	35	75	80	20	18	27	118
	10	10					
66	227	75					
	00						

SOC code: 1 11100011 0 10

Secret bits: 0

14

## Example (Data Extraction) (3/3)

Input code: ...1 11100011 0 10 ... **1** 00011111

Index table

20	35	75	80	20	18	27	118
66	227	75	83	80	75	27	120
66	31	31	75	31			

Red annotations: '31' above the top-right cell, '01' and '10' in the second row, '11' and '00' in the third row, 'x' in the second row, and arrows pointing from the '31' in the second row to the '31' in the third row and to the '0 11' in the SOC code below.

SOC code: 1 11100011 0 10 ... 0 11

Secret bits: 0 ... 1

15

## Experimental results

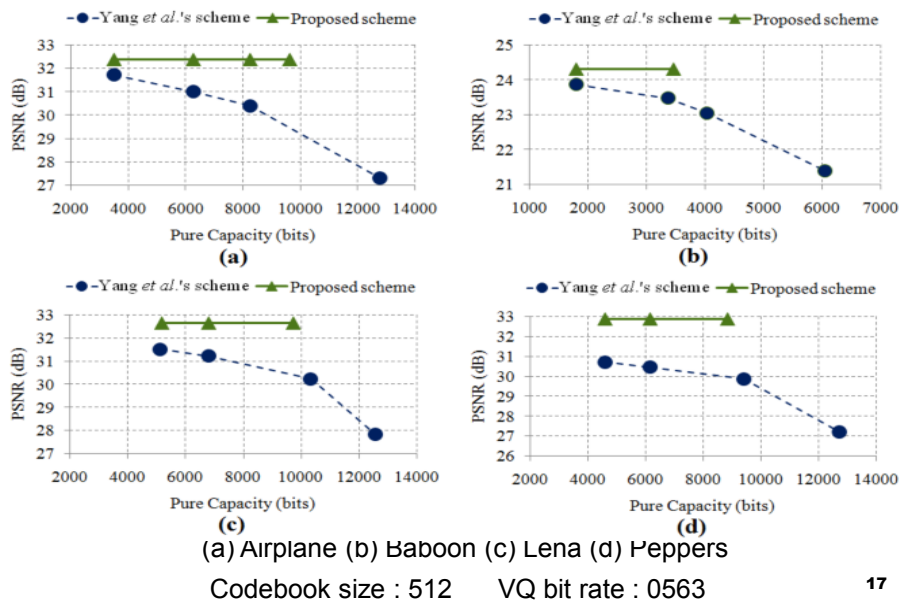
Images	VQ	$m = 2$			$m = 3$		
	PSNR	SOC rate (bpp)	Capacity (bits)	Rate (bpp)	SOC rate (bpp)	Capacity (bits)	Rate (bpp)
Airplane	32.38	0.405	8,223	0.515	0.404	9,642	0.514
Baboon	24.31	0.564	2,278	0.595	0.546	3,456	0.587
Boat	29.70	0.394	8,644	0.508	0.405	9,606	0.513
Lena	32.63	0.398	8,515	0.509	0.402	9,734	0.515
Peppers	32.85	0.422	7,619	0.524	0.422	8,859	0.524
Sailboat	28.93	0.435	7,097	0.529	0.434	8,356	0.529
Tiffany	30.47	0.356	10,082	0.490	0.367	11,268	0.486
Toys	32.68	0.340	10,675	0.483	0.348	12,123	0.486
Zelda	30.06	0.422	7,609	0.525	0.417	9,067	0.521

Codebook size : 512      VQ bit rate : 0563  
 $m$ : no. of bits used to represent each SOC code

16



## Experimental results (comparison)



## Conclusion

- Proposed scheme generate legitimate SOC code as embedded code.
- Instead of changing indices, the method to encode each index is used to embed data.
- Proposed scheme doesn't degrade quality of stego-image.
- The proposed scheme can be used beside other reversible VQ-based data hiding schemes to further improve capacity of these schemes and embed side information of them.

18



Thanks for your attention!!