Image Watermarking Based on Feature Point Synchronization with the Use of ECC

LIRMM, October 25, 2012

Boris Assanovich
bas@grsu.by
YK State University of Grodno
EMERGE Project

Outline

- Part I: Introduction to Watermarking
  - Watermarking types and generations
  - Synchronization Techniques for Image Watermarking
  - Image Features and Feature Points
  - Types of Feature Point Detectors
  - Corner Harris Detector and its properties
  - Problems of FP matching
- Part II: Error correcting coding and application for Watermarking
  - Historical Pedigree of ECC
  - Concatenated codes
  - Davey & MacKay code
  - Proposed Coding-Decoding scheme
  - Synchronizing Codes (VT& Levinstein)
  - Prototype of our scheme for data embedding
  - Examples of its use
- Conclusions and future research

Part I:

Introduction to Watermarking

Introduction

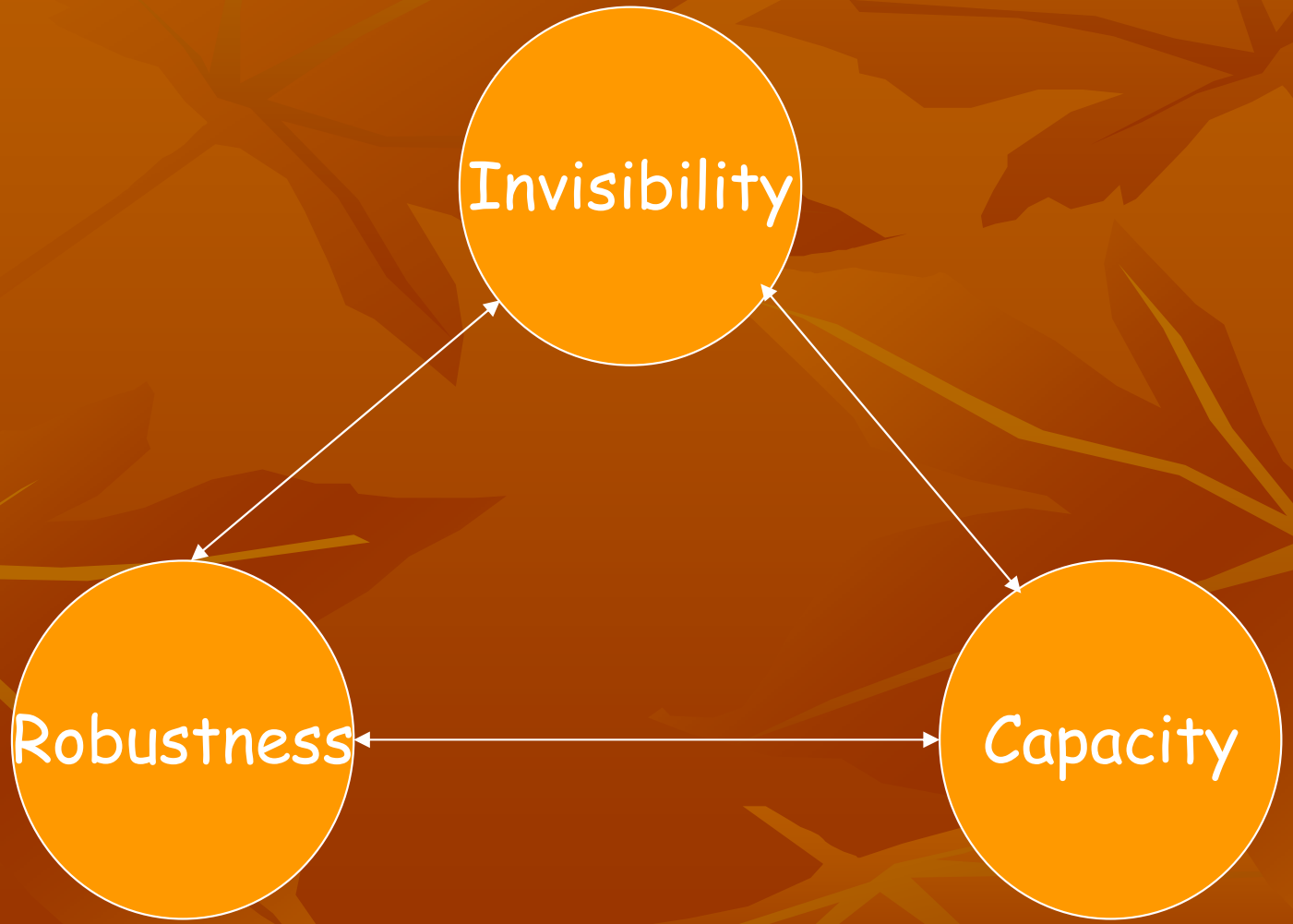
- Watermarking is the area of Information Hiding Technology, containing also another area – Steganography.
- **Steganography** keeps the existence of the information in secret **Watermarking** makes the information imperceptible
- A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal. It is typically used to identify the ownership of it.
- The term "digital watermark" was first coined in 1992 by Andrew Tirkel and Charles Osborne

- http://en.wikipedia.org/wiki/Digital_watermarking
- A.Z.Tirkel, G.A. Rankin, R.M. Van Schyndel, W.J.Ho, N.R.A.Mee, C.F.Osborne. "Electronic Water Mark". DICTA 93, Macquarie University. p.666-673

The Necessity for Data Hiding

- Ownership of digital images, protect copyright
- Data integrity authentication, fraud detection
- Traitor-tracing (fingerprinting video)
- Adding captions to images, additional information, such as subtitles
- Intelligent browsers, automatic copyright information
- Covert communication using images (secret message is hidden in a carrier image)

Watermarking Requirements



Watermarking Types

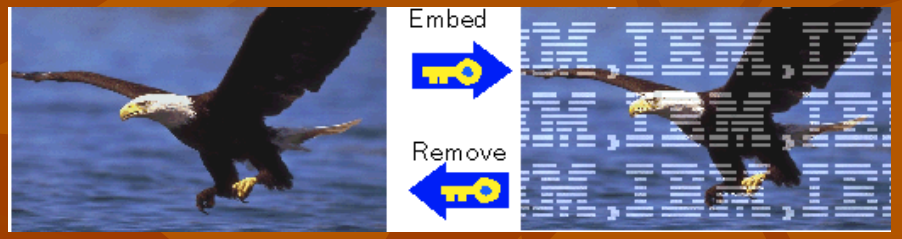
- Visible and invisible: how we see the embedded information
- Fragile and Semi-fragile: test the distorted or broken under slight changes images when they exceed a user-specified threshold
- Domain chosen watermarking: spatial domain, frequency domain, spread domain
- Detection used watermarking: blind, non-blind (data assisted), semi-blind ; oblivious, non-oblivious (watermarking original image is needed or not)

Generations of Watermarking

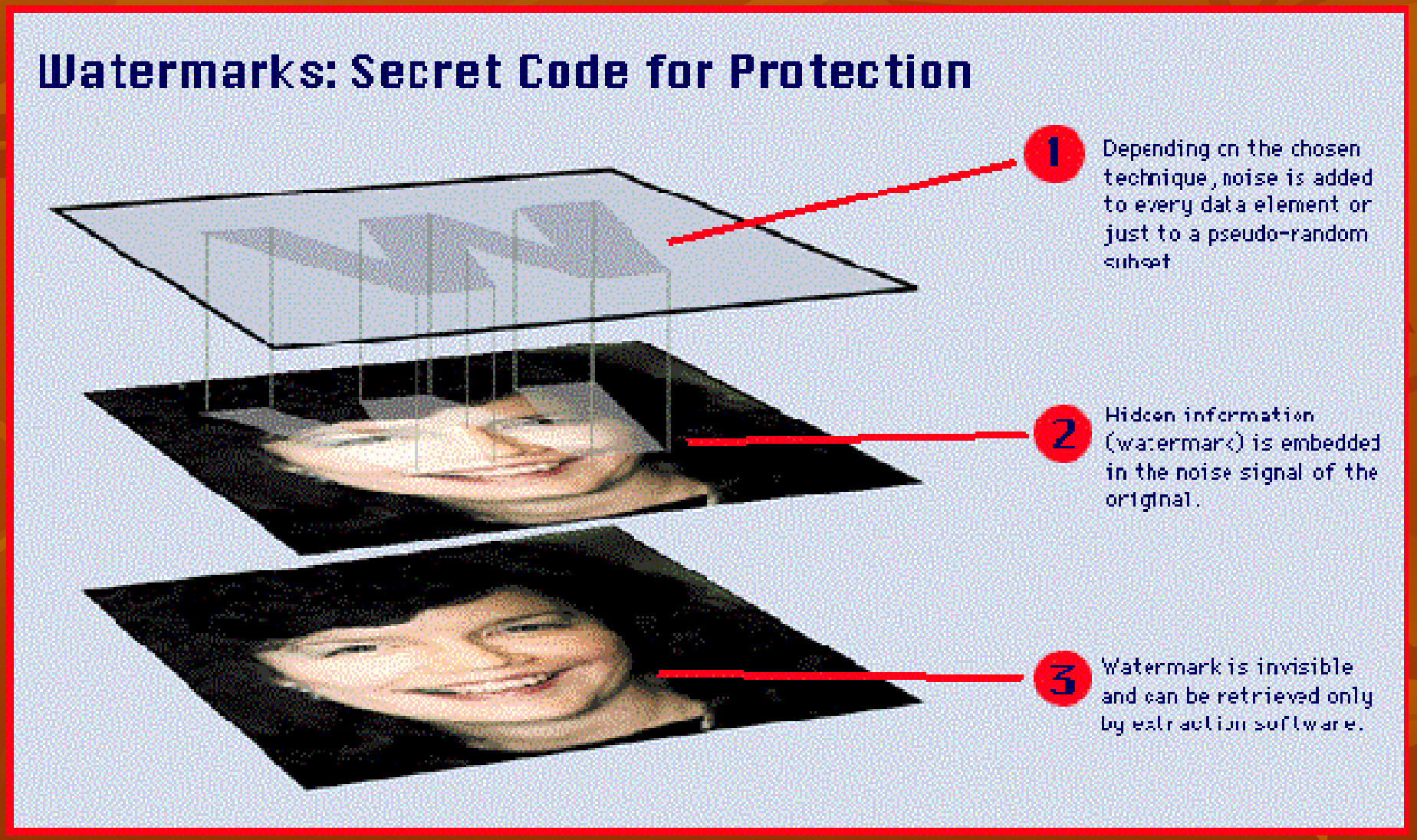
- **First generation watermarking schemes** use pixels or samples or transform coefficients to embed the information. Drawback - watermark is not embedded in the perceptually significant portions of data
- **Second generation watermarking schemes** involve the notion of perceptually significant features which may be abstract or semantically meaningful (Kutter)
- Third generation schemes will give the opportunity to insert robust, high density watermarks in 2D and 3D data in intelligent way

- M. Kutter S. K. Bhattacharjee T. Ebrahimi. Towards Second Generation Watermarking Schemes. Signal Processing Lab. Swiss Federal Institute of Technology, 1999.

Visible Watermark



Invisible Watermark



Synchronization Techniques for Image Watermarking

Any distortion of image can dramatically reduce the ability to detect the watermark leading to lose the synchronization. What is done to prevent it?

- Use periodical sequences
- Use templates insertions
- Use invariant transforms
- Use original image
- Use image content

Brief Comparison of known Synchronizing Methods

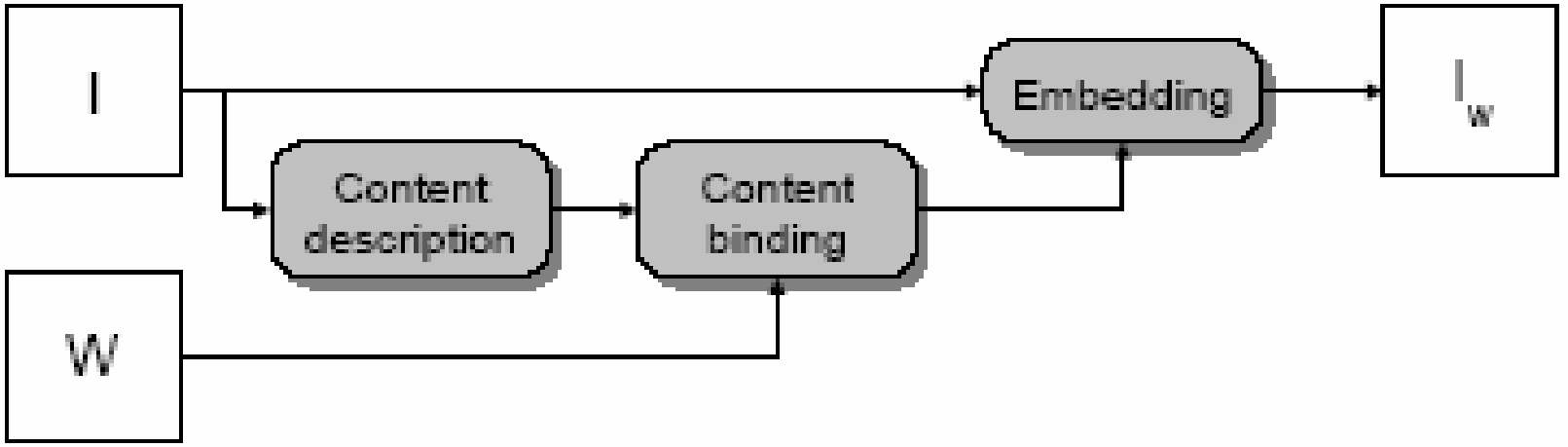
	Local Transformation Robustness	Global Transformation Robustness	Notes
Periodic insertion	No	Yes	-
Template insertion	No	Yes	Can be removed
Invariant transform	No	Yes	-
Non-blind	Yes	Yes	Computational cost

Is it possible to use the natural image content for synchronization?

Content-based Data Embedding

Content based (feature-based) watermarking

- new class of the watermarking techniques which link the watermark with image semantics and not the coordinates
- problem of synchronization could be solved according to invariant reference to image feature characteristics



How the Image Feature Points are Used?

- Feature points could be used for:
  - Motion tracking
  - Image alignment
  - 3D reconstruction
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation

Characteristics of Good Features

- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature has a distinctive description
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image

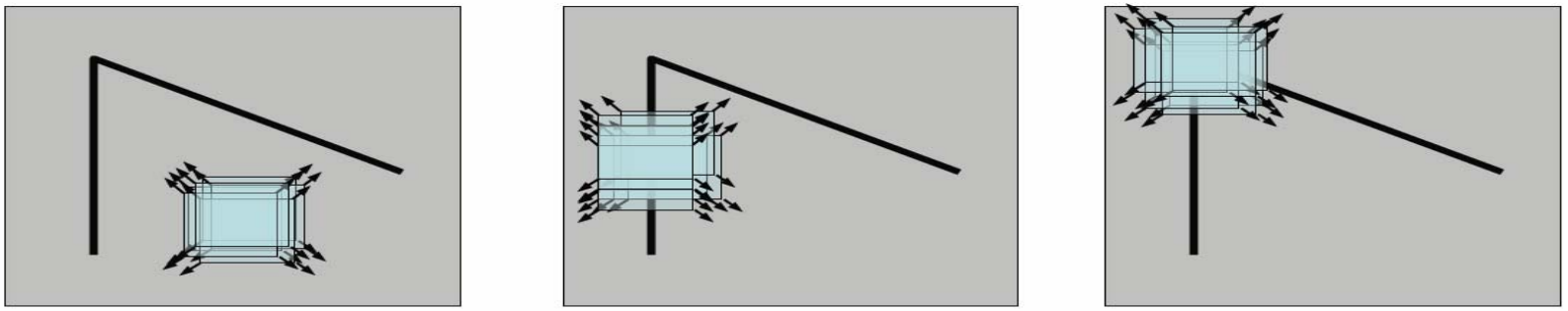
Feature Detectors

Feature detectors	Edge	Corner	Blob
Canny	X		
Sobel	X		
Harris	X	X	
SUSAN	X	X	
Level curvature		X	
FAST		X	X
Laplacian of Gaussian		X	X
Difference of Gaussians		X	X
Determinant of Hessians		X	X

http://en.wikipedia.org/wiki/Feature_detection_(computer_vision)

Corner Detection

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in the intensity



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

C.Harris and M.Stephens. A Combined Corner and Edge Detector. *Proceedings of the 4th Alvey Vision Conference*, 1988

Math for Corner Detector

Change in appearance for the shift [*u,v*]:

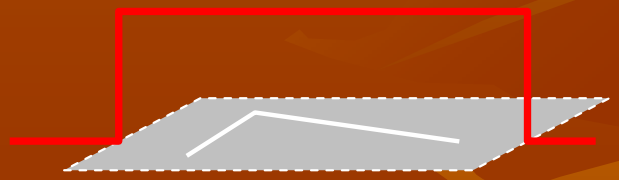
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

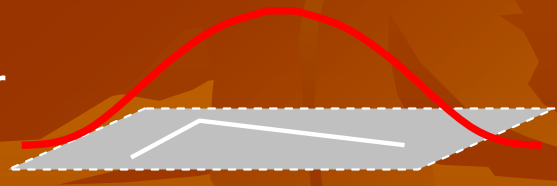
Intensity

Window function *w(x,y)* =



1 in window, 0 outside

or



Gaussian

Math to Use

Second-order Taylor expansion of $E(u,v)$ about (0,0) and quadratic approximation simplifies to:

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a *second moment matrix* computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

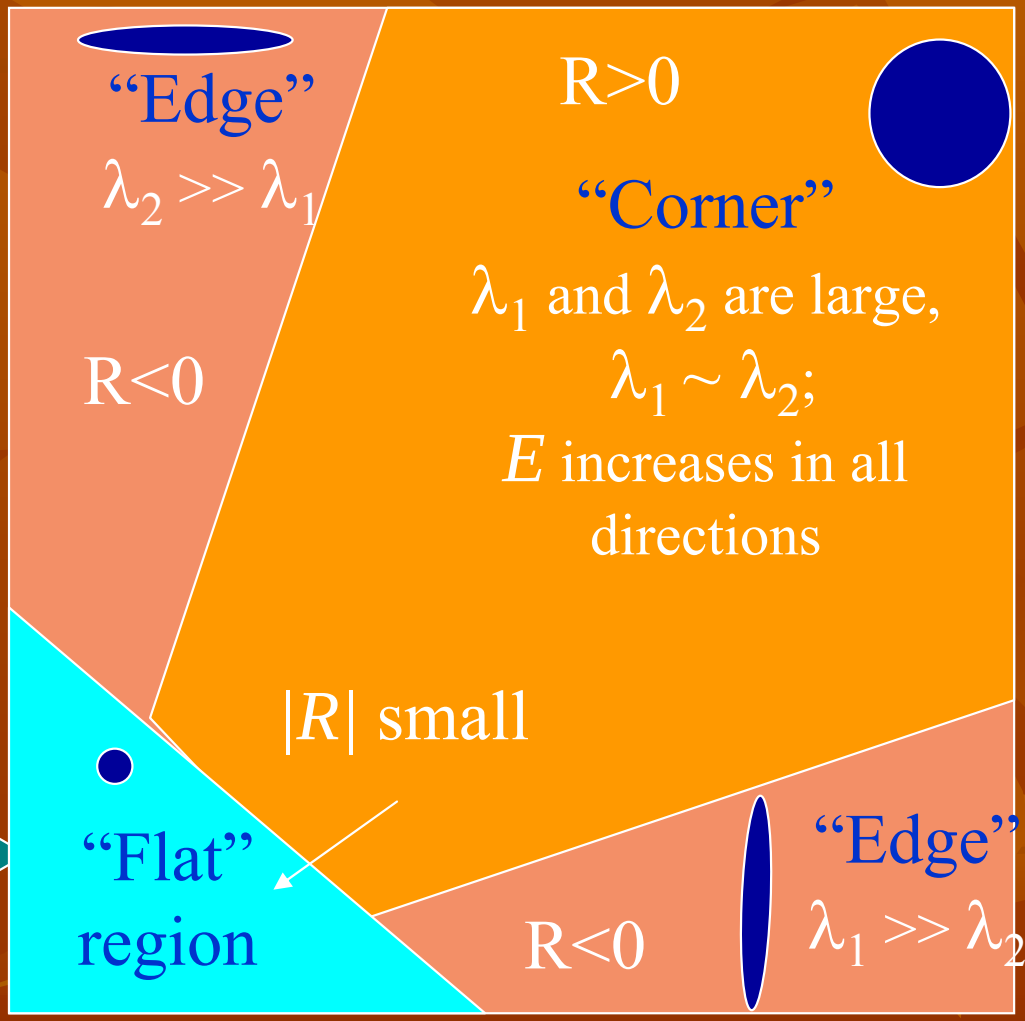
Harris detector: Steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix M in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maximum of response function

Corner Response Function

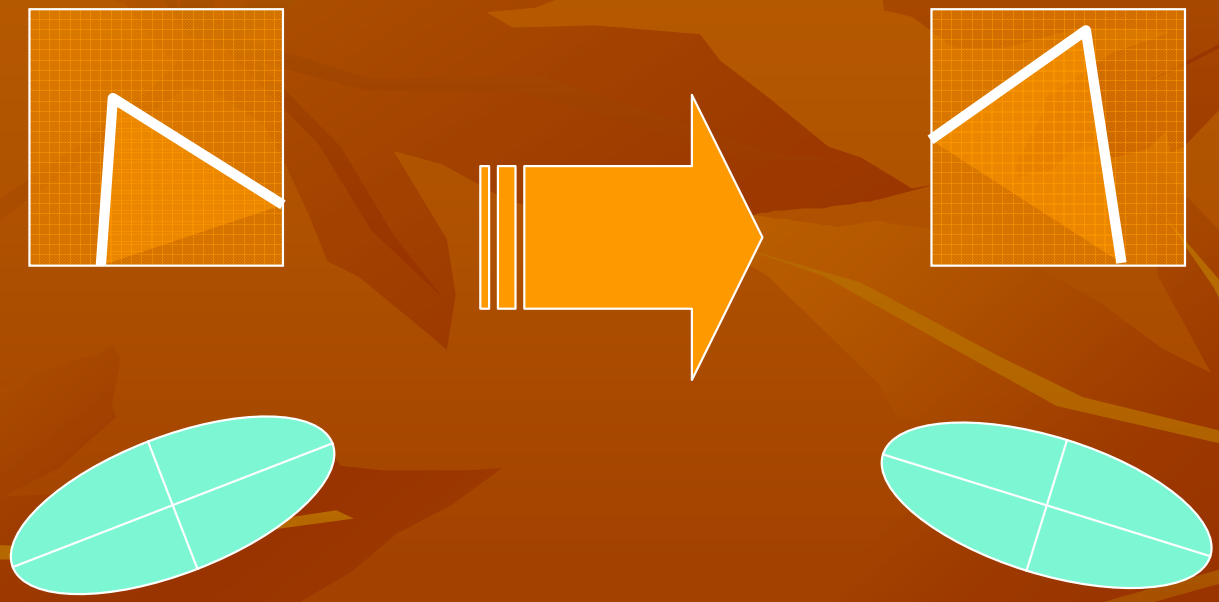
$$R = \det(M) - \alpha \,\mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

Corner response function-
"Cornerness" function



λ_1 and λ_2 are small;
E is almost constant
in all directions

Harris Detector: Invariance Properties (Rotation)

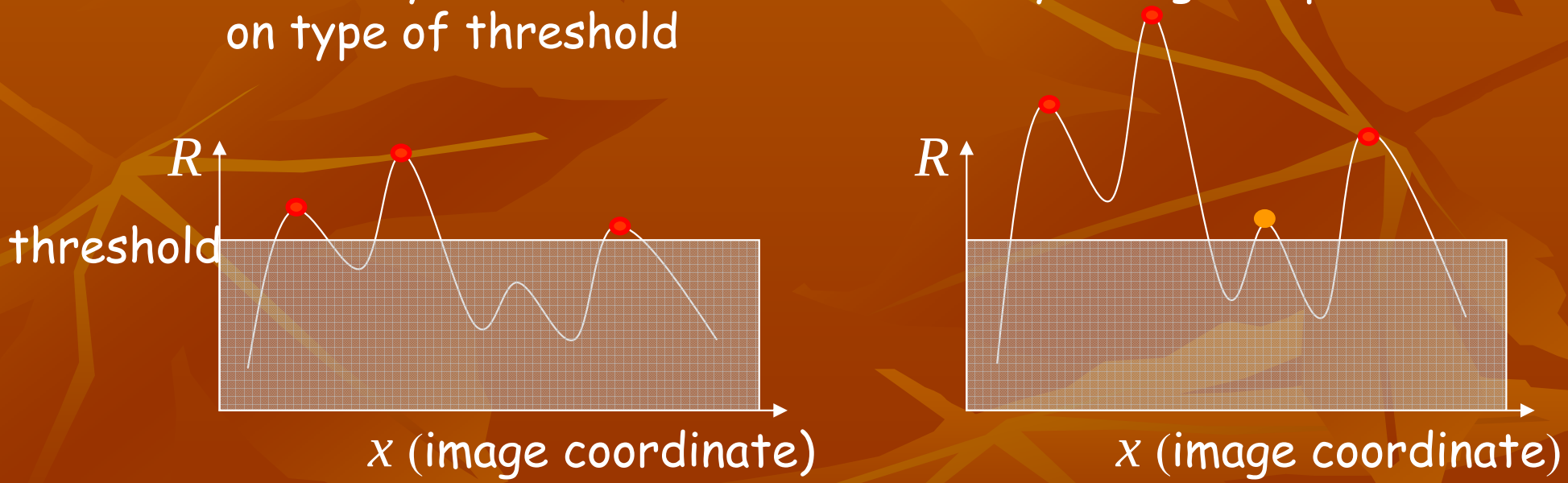


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris Detector: Invariance Properties (Intensity)

- Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
- Intensity scale: $I \rightarrow a\,I$
- Partially invariant to affine intensity change, dependent on type of threshold



Harris Features (in red)

The tops of the horns are detected in both images

http://lear.inrialpes.fr/

Any Problems of Matching FP

http://lear.inrialpes.fr/

213 / 190 detected feature points, some of them deleted or inserted. What to do .. Synchronization? ECC ?

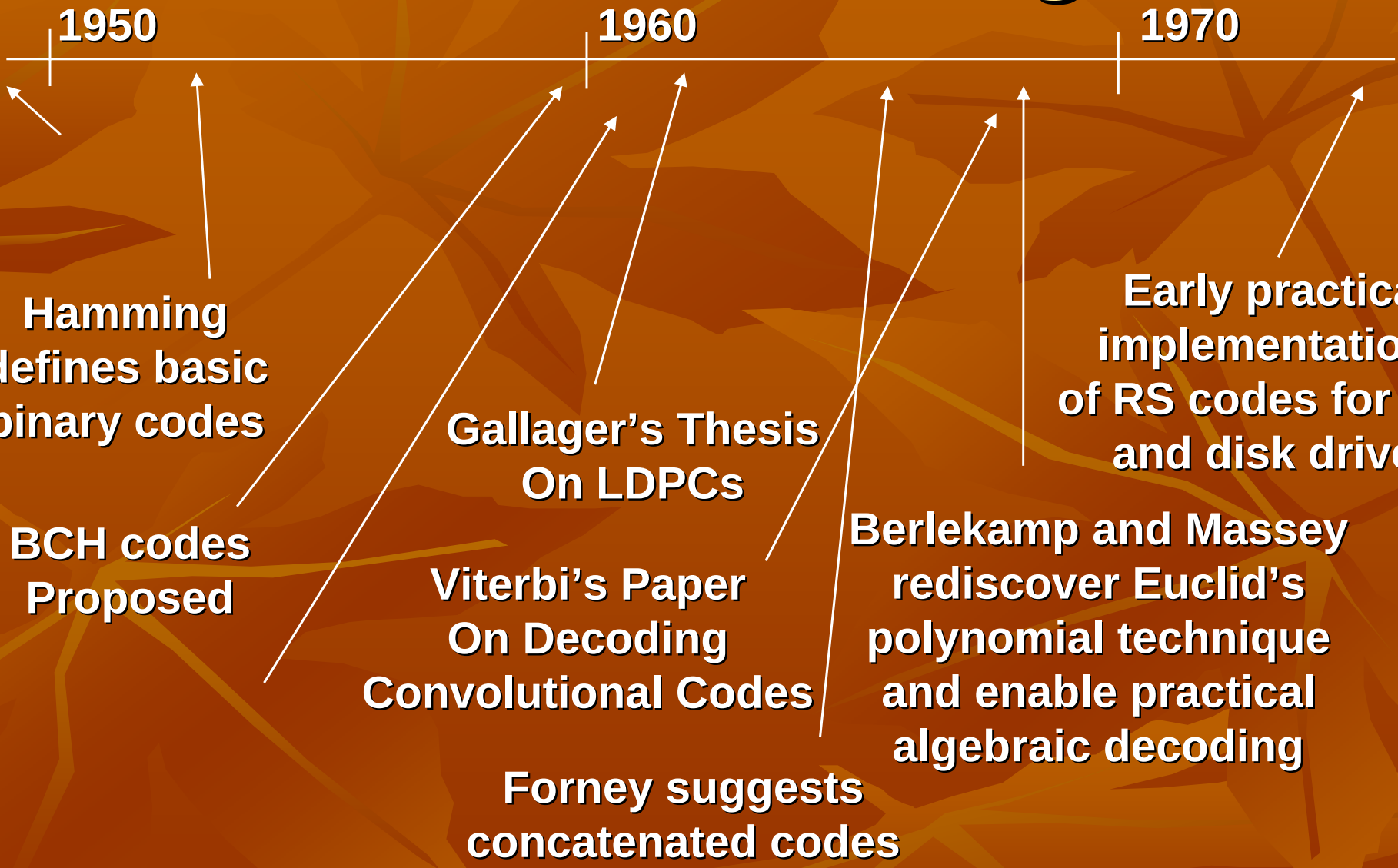
Types of Synchronization Errors

- Additive errors are special case of deletion and insertion in same position of bits of opposite value
- Repetition/duplication error: copies bit
- Bit/peak shift: 01 becomes 10
- Decision 1: Synchronizable codes, Synchronisation with timing, Marker codes
- Decision 2: Convolutional codes, binary and non-binary block EEC

Part II:

Error Correcting Coding and its Applications for Watermarking

EEC Historical Pedigree I



EEC Historical Pedigree II

Ungerboeck's TCM Paper - 1982

RS codes appear in CD players

First integrated Viterbi decoders (late 1980s)

TCM Heavily Adopted into Standards

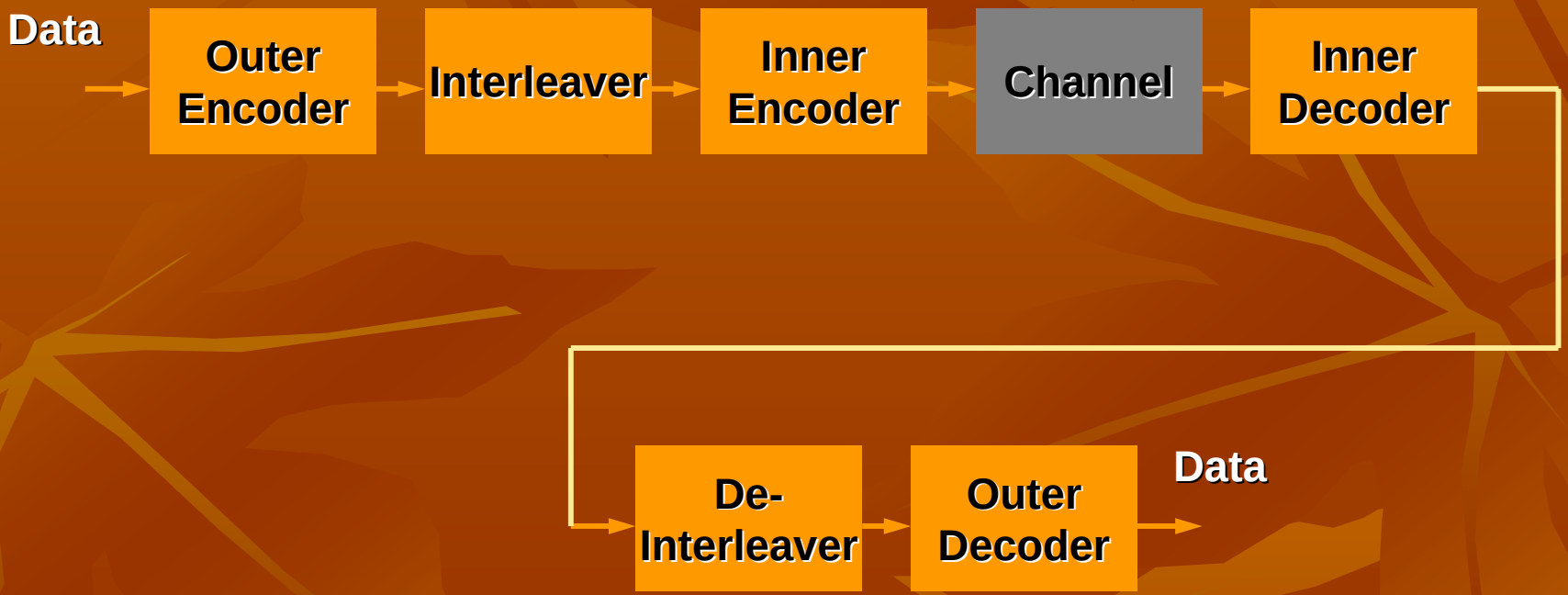
Berrou's Turbo Code Paper - 1993

Turbo Codes Adopted into Standards (DVB-RCS, 3GPP, etc.)

Renewed interest in LDPCs due to TC Research

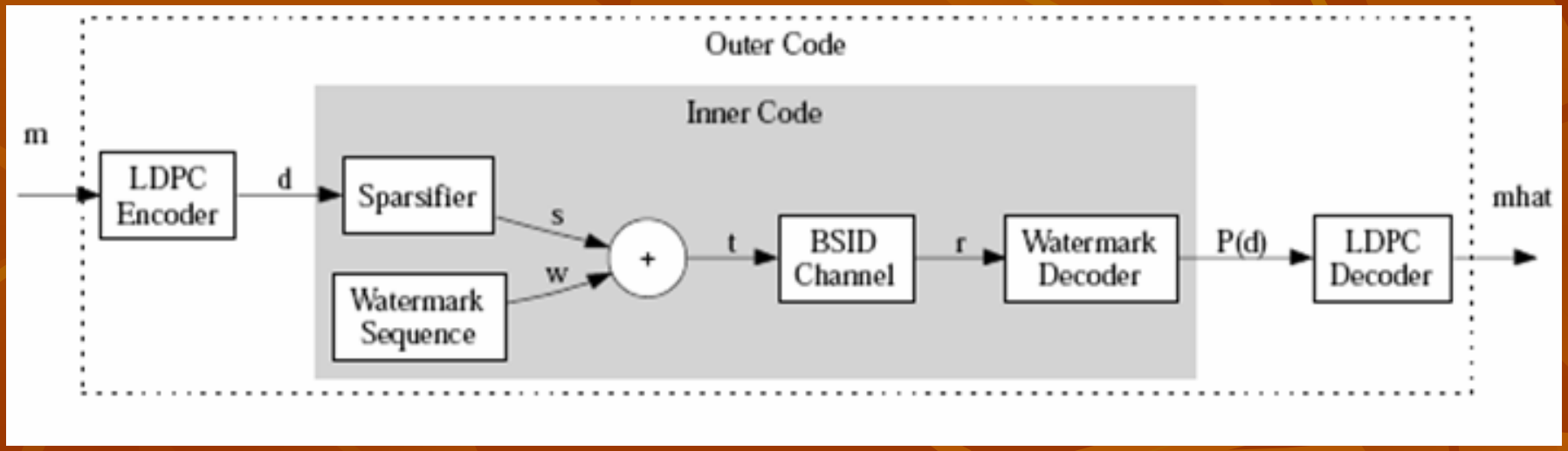
LDPC outrun Turbo Codes For DVB-S2 Standard - 2003

Concatenated Coding-Decoding Scheme



Previous Works in Coding Schemes Design

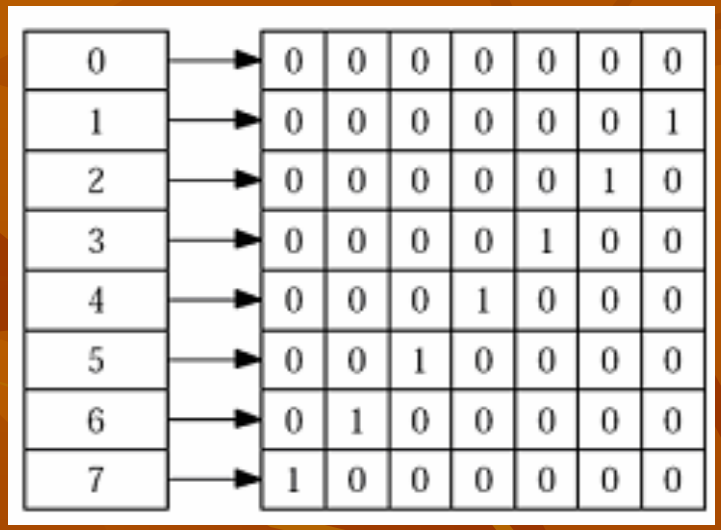
- To make Concatenated coding scheme with 2 types of codes:
- Q-ary outer code that maps symbol to a fixed length binary string
- Binary inner code that provides the synchronization
- Davey & MacKay used a pseudo-random sequence together with a sparse code for binary insertion-deletion channel

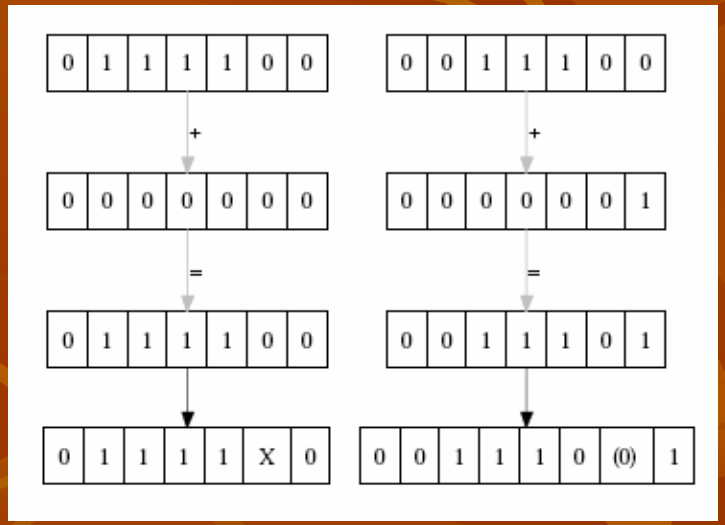


M. C. Davey, D. J. C. MacKay, Reliable communication over channels with insertions, deletions, and substitutions. IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 687–698, 2001.

DM-sparsefier

- Davey & MacKay DM - code presents q-ary symboled in sparse way and combine it with pseudo-random sequence (watermark) taking mod 2;
- Pseudo-random sequence could destroy synchronization code properties;
- Decoding is very expensive and Decoder cannot distinguish between:
  -Channel errors
  -Sparse symbol uncertainty

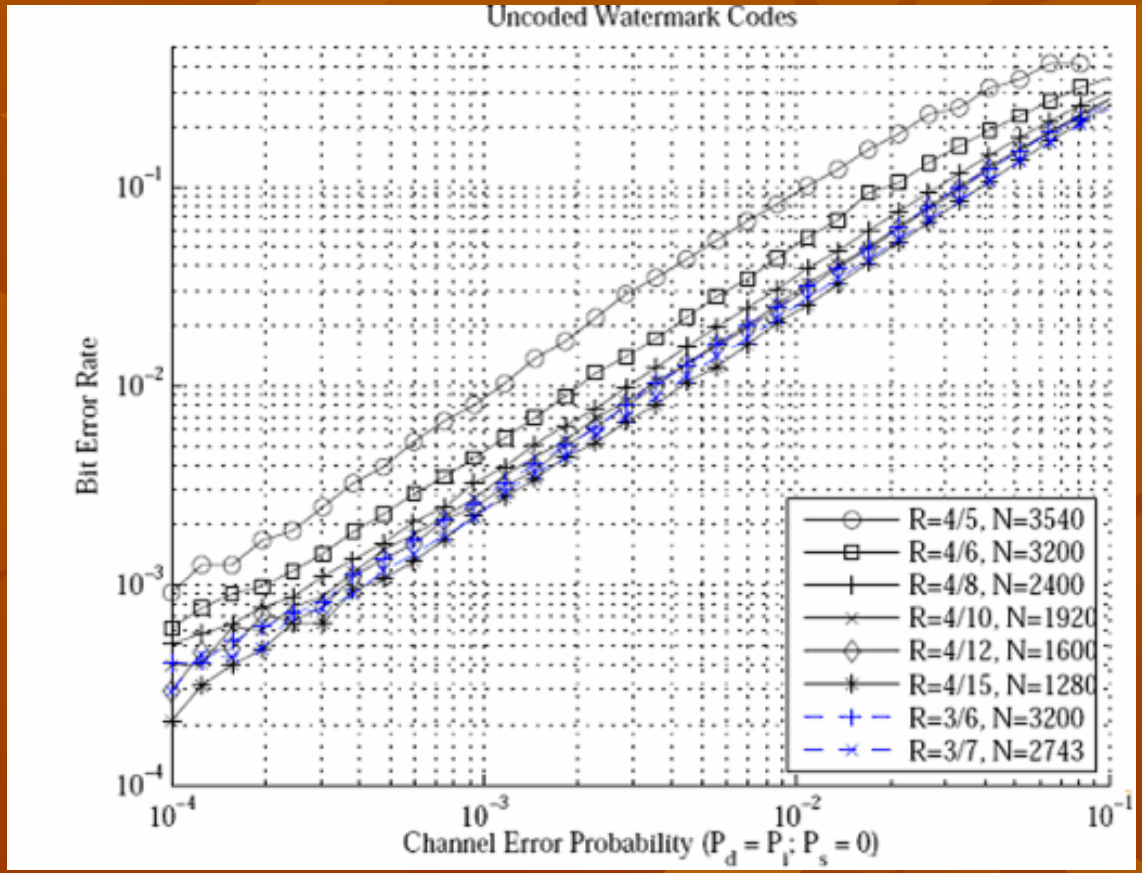




Improvement of DM-scheme

JA Briffa, HG Schaathun (2008):

- Analyzed the inner DM codes, proposed to consider its distance properties;
- Proposed to use turbo-codes instead of LDPC;
- Simulated and showed that 3/7 inner code is optimal and 3/6 is close

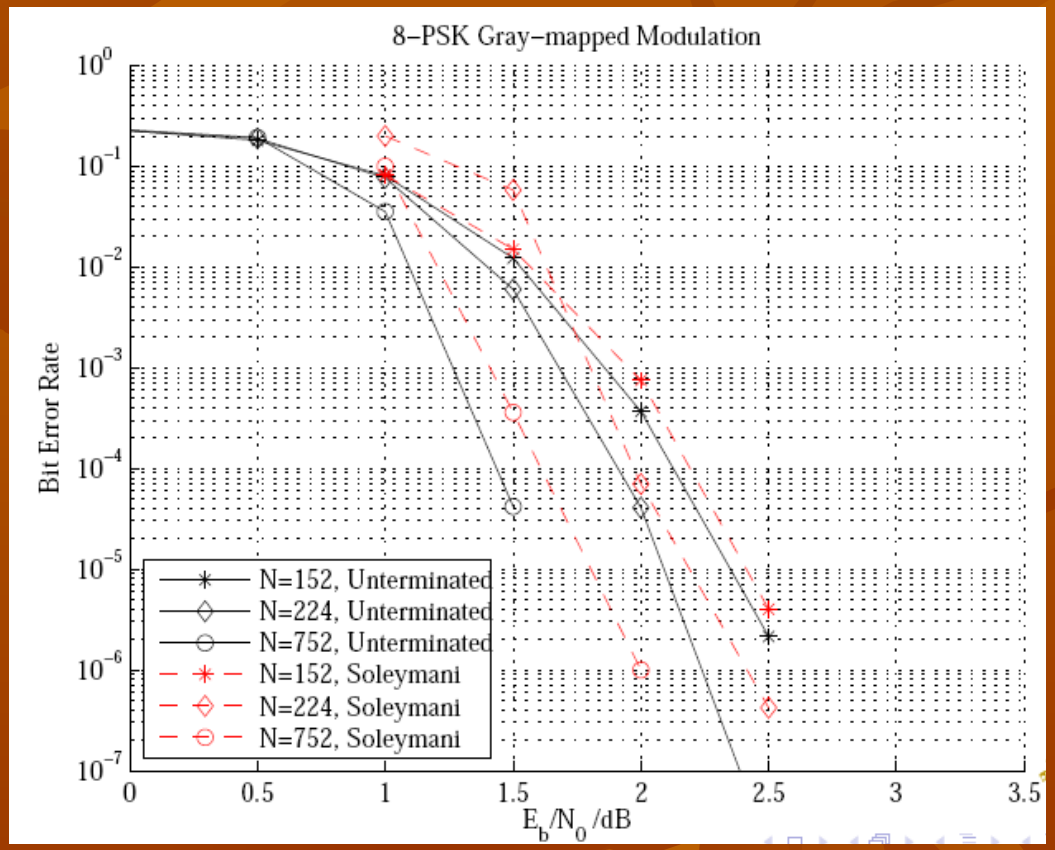


JA Briffa, HG Schaathun Improvement of the Davey-MacKay construction. ISITA 2008.

Error Control Watermarking - Deletion-Insertion Correcting Codes and Robust Watermarking». University College Dublin, February 2008.

From LDPC to Turbo

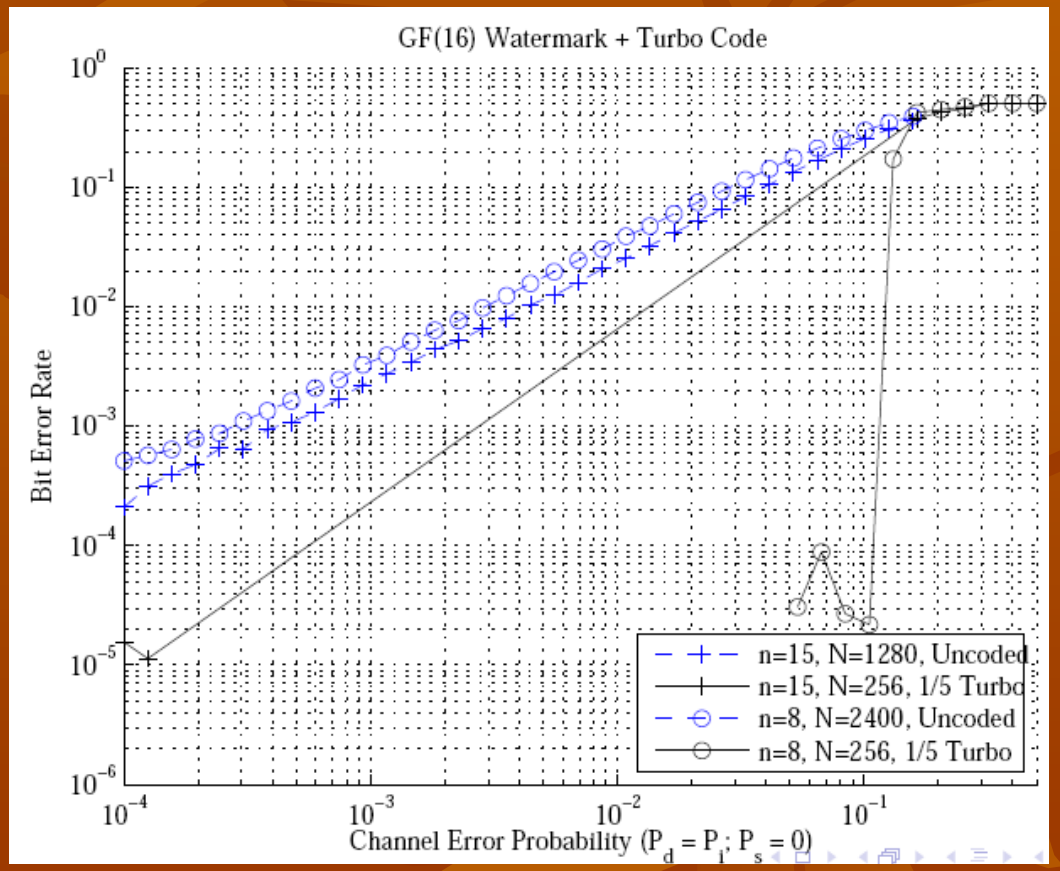
- Nonbinary Turbo codes have been weakly studied during the last decade
- LDPC have shown large complexity of a decoder in DM construction
- Simulation made show the efficiency of turbo codes with the use of PSK-modulation and Gray coding



- M.R. Soleymani, Y. Gao and U. Vilaipornsawai, Turbo Coding for Satellite and Wireless Communications, Kluwer Academic Publishers, MA, USA, 2002.

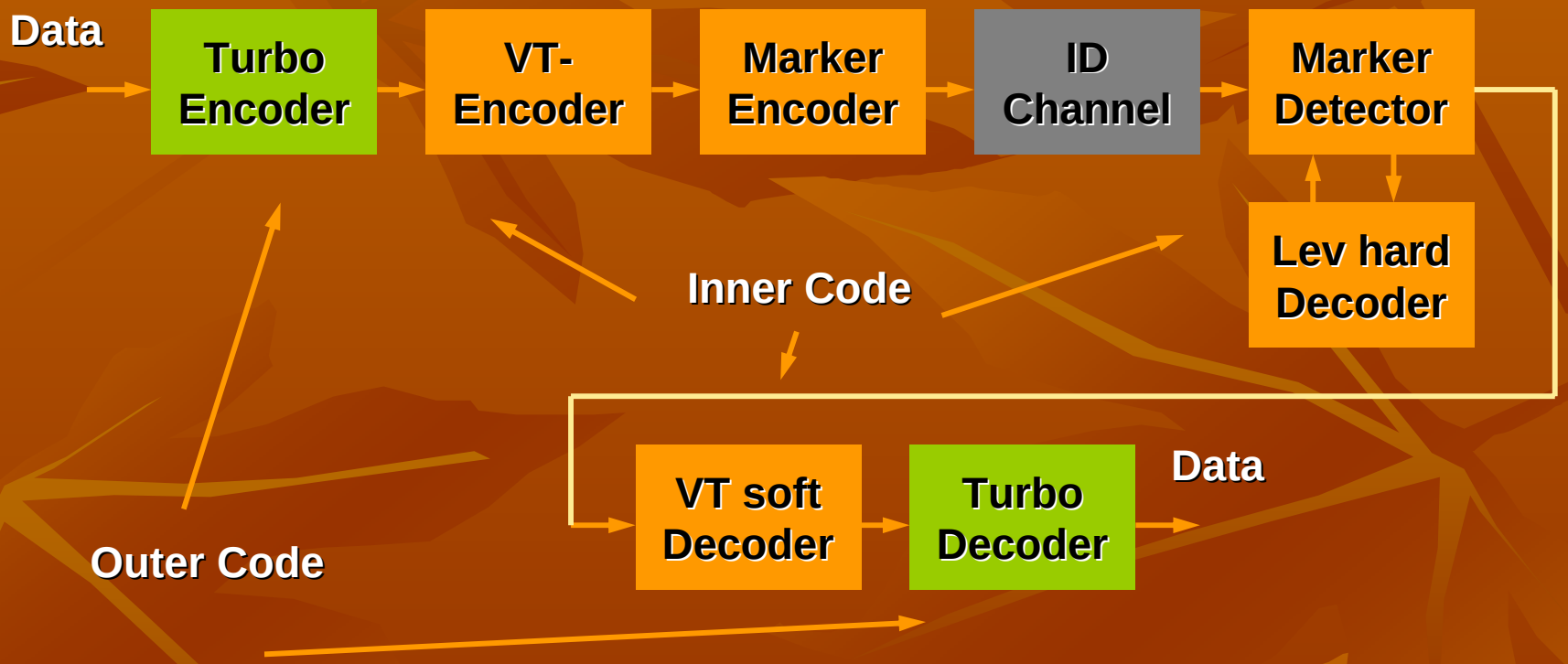
Results of Turbo Code Use

- Briffa et all used turbo codes over GF(16)
- To compare with DM construction focused on low-rate systems
- DM watermark code had rate k/n = 4/15
- Briffa et all used R = Turbo code with rate 1/5 and some inner code k/n = 4/15 obtaining the overall rate R = 4/75



J.A.Briffa, Hans.G.Schaathun, S.Wesemeyer
An improved decoding algorithm for the Davey-MacKay construction. ICC 2010.

Proposed Coding-Decoding Scheme



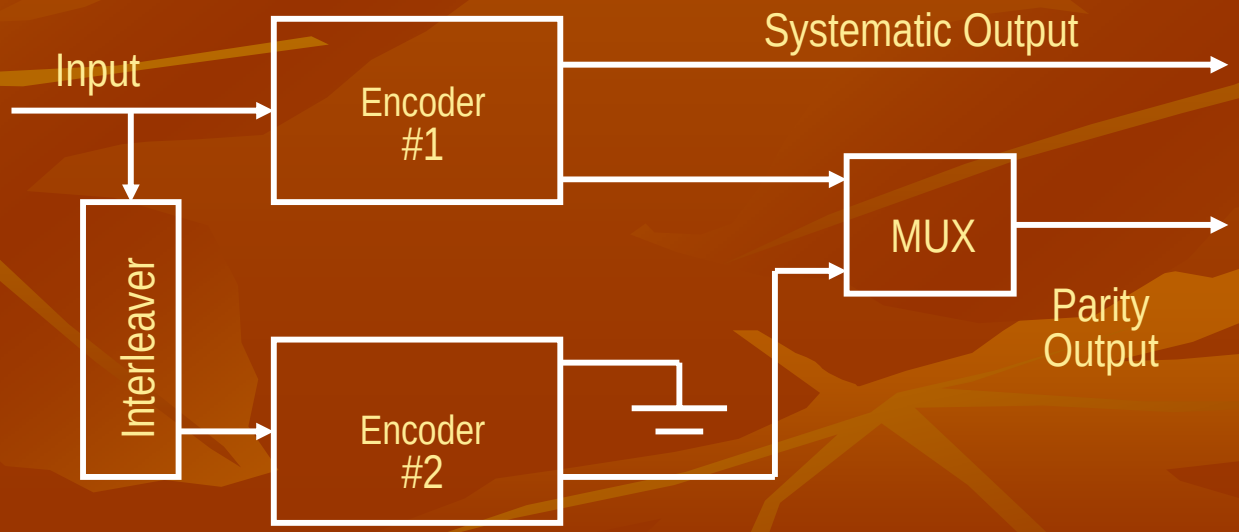
- Outer ECC - Non-binaryTurbo Code
- Inner code the binary Levinstein code with a special marker

Turbo Codes

- History of turbo codes
  - Turbo codes were proposed by Berrou and Glavieux in the 1993 International Conference in Communications
  - Performance within 0.5 dB of the channel capacity limit for BPSK was demonstrated
- Features of turbo codes
  - Parallel concatenated coding
  - Recursive convolutional encoders
  - Pseudo-random interleaving
  - Iterative decoding

Turbo Coding Scheme

- Instead of concatenating in serial, codes can also be concatenated in parallel
- The original turbo code is a parallel concatenation of two *recursive systematic convolutional* (RSC) codes
- Turbo codes possess random-like properties

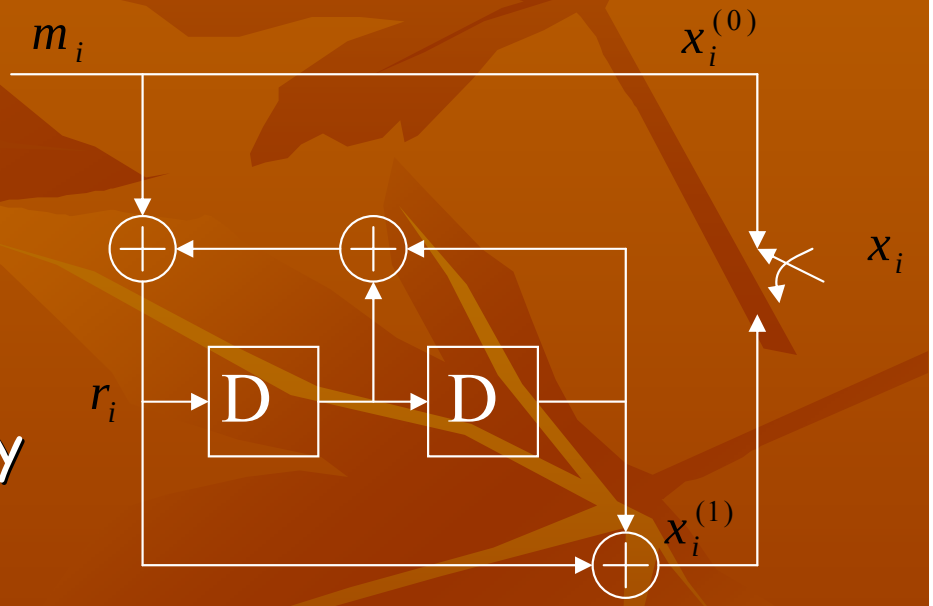


Binary RSC Encoder

- An RSC encoder can be constructed from a standard convolutional encoder by feeding back one of the outputs
- An arbitrary input will cause a high weight output with high probability
- An RSC code will produce low weight outputs with low probability

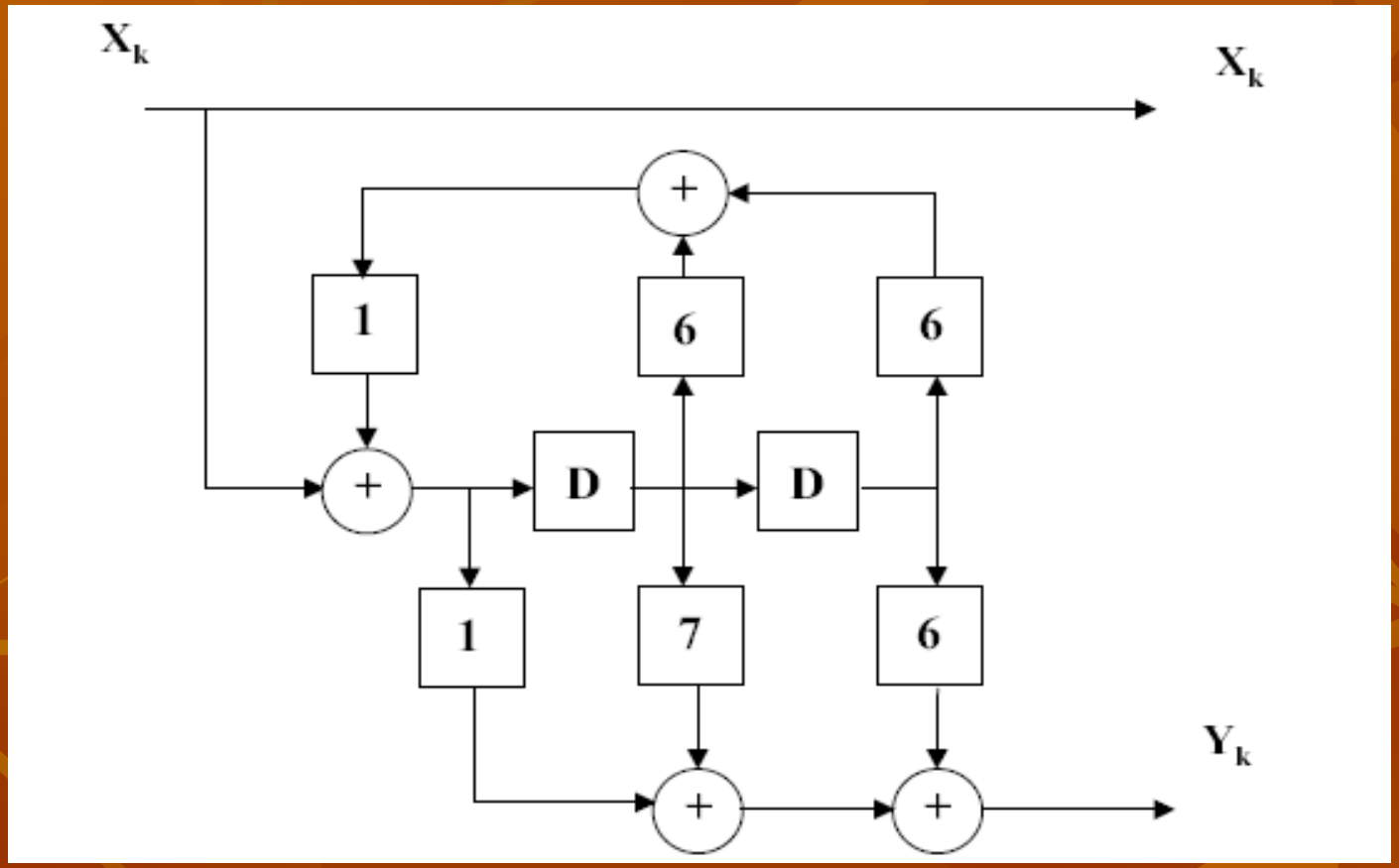
The parallel concatenation of both encoders will produce a "good" code

- Since the interleaving pattern is known, decoding is possible



Constraint Length K= 3

RSC Encoder over GF(2^3)



Generator matrix: g = [1 6 6; 1 7 6];

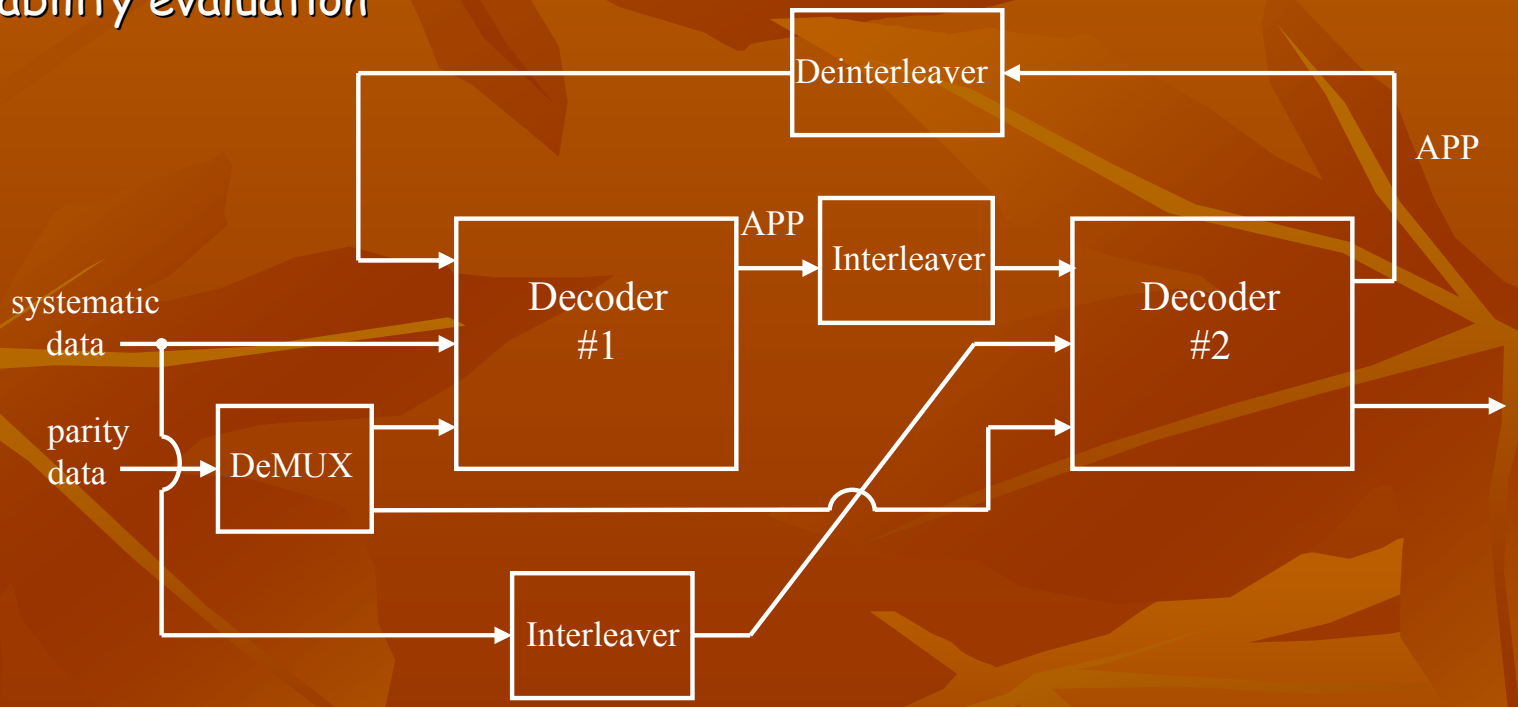
Polynomials: h0 =1 h1 =6, h2 =6; g0 =1, g1 =7, g2 =6.

Outer Code: Nonbinary Code

- 8-ary Codes: over the Field GF(2^3) with elements {0, 1, α, α^2, α^3, α^4, α^5, α^6 }, where α is a root of the primitive polynomial X^3+X+1 give the dictionary *A*:
- $0 \rightarrow 000 \rightarrow 0$; $1 \rightarrow 001 \rightarrow 1$
- $\alpha \rightarrow 010 \rightarrow 2$; $\alpha^2 \rightarrow 100 \rightarrow 4$;
- $\alpha^3 = \alpha + 1 \rightarrow 011 \rightarrow 3$;
- $\alpha^4 = \alpha^2 + \alpha \rightarrow 110 \rightarrow 6$;
- $\alpha^5 = \alpha^2 + \alpha + 1 \rightarrow 111 \rightarrow 7$;
- $\alpha^6 = \alpha^2 + 1 \rightarrow 101 \rightarrow 5$

Decoder Scheme

- Decoder includes 2 decoders and interleavers to estimate the *a posteriori probability* (APP) of not correlated data element (bit or symbol)
- The APP's are used as *a priori* information by the other decoder
- Performance generally improved from iteration to iteration with more precise probability evaluation



Decoding of Turbo Code

- Normal decoding algorithms (Viterbi algorithm) find the most likely sequence of bits that was transmitted
- In a turbo decoder, want to find the likelihood of each bit (symbol). This serves as the *a posteriori* probability or the reliability of each bit (symbol), to use as input to the next decoder
  - Optimal MAP (Maximum A-Posteriori) – BCJR (Bahl, Cocke, Jelinek, Raviv)
  - Simpler - SOVA (Soft Output Viterbi Algorithm) – lose roughly .7 dB coding gain

MAP Decoding and Information Exchange

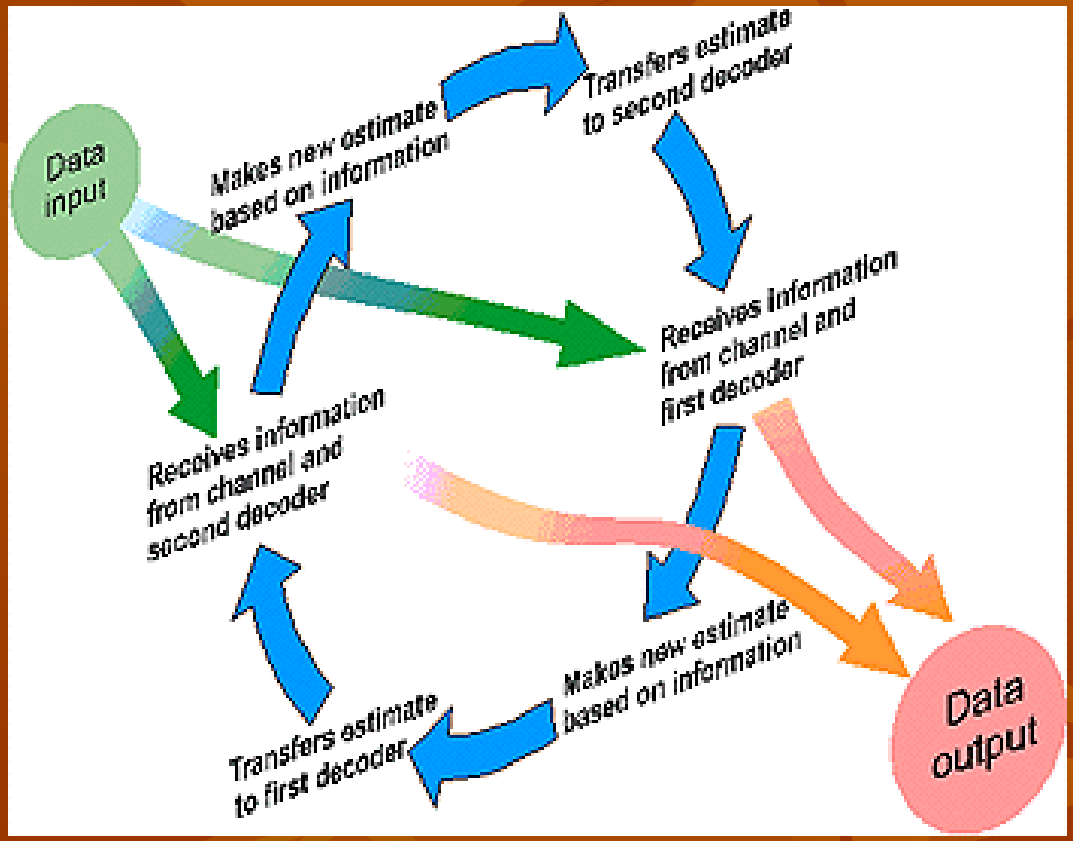
LLRs for L(*u*) for binary and non binary case and **APP** P(/) are defined with the help of BCRJ algorithm:

$$L(u) = \ln \frac{P(u=+1)}{P(u=-1)}$$

$$L(u) = \ln \frac{P(u=A_i)}{P(u=A_j)}$$

$$P(u = A_i \mid r) = \frac{p(u = D', r)}{P(r)} =$$

$$= \frac{\sum_{(s',s)\in S} p(s_i = s', s_{i+1} = s, r)}{\sum p(r_j)}$$



- Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm (1974)
- http://www.xenotran.com/turbo_tech_error_turbo.html)

Our Scheme Decoding Example (Outer Code)

x=[2 0 1 6]

r=

[2 0 1 6

2 2 5 7

6 6 6 7]

The outer code correct the errors and erasures

The inner code must take care of synchronization

LLR-values and Apost probabilities

(1st and 2nd iterations

1,7	7,14	10	3	0,29	20,9	7,96	6,376
11,8	0,18	11,8	0	5,53	0	22,8	0
21,9	0	1,86	0,5	27,3	0	1,01	0
24,4	2,84	6,14	0,4	18,9	3,17	12,1	0
14,2	0,95	0,13	1,9	14,1	0	0	1,886
0	5,73	0,83	5,1	0	4,91	0	6,407
0,42	6,92	2,13	12	0	12,5	2,12	25,35
2,48	4,33	0	12	0	3,15	0	12,03

0,02	0,25	0,3	0,1	0	0,47	0,17	0,123
0,15	0,01	0,38	0	0,08	0	0,5	0
0,28	0	0,06	0	0,41	0	0,02	0
0,32	0,1	0,19	0	0,29	0,07	0,26	0
0,18	0,03	0	0,1	0,21	0	0	0,038
0	0,2	0,03	0,1	0	0,11	0	0,123
0,01	0,25	0,06	0,3	0	0,28	0,05	0,487
0,03	0,15	0	0,3	0	0,07	0	0,231

Synchronizing Codes

- 60-s Levenshtein and V&T – small codes, correcting one error
- 1999 Schulman & Zuckerman asymtotically good codes
- 2001 Davey & MacKay – Watermark Codes
- Their construction are not ideally suited for application in watermarking
- Channel model is one-dimensional
- 2011 Paluncic nonbinary insertion/deletion codes

Varshamov Tenengoltz Levenshtein codes

- 1965 : Varshamov Tenengoltz construction
- 1965 : Levenshtein proposed algorithm to correct insertion-deletion error

$$\sum_{i=1}^{n} ix_i \equiv a \mod m$$

$$a = 0,1,\ldots, m-1$$

[1] Error correcting code for asymetric errors. Automatic, Telemechanic. SSSR, Feb. 1965.

[2] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady Akademii Nauk SSSR, August 1965.

VT-code for n=4, a=0

$i = 4\ 3\ 2\ 1$	$\Sigma i x_i$	$\Sigma i x_i \equiv a \bmod(n+1)$
$x = 00\ 0\ 0$	0	0
$x = 00\ 0\ 1$	1	1
$x = 00\ 1\ 0$	2	2
$x = 00\ 1\ 1$	3	3
$x = 01\ 0\ 0$	4	4
$x = 01\ 0\ 1$	4	4
$x = 01\ 1\ 0$	5	0
$x = 01\ 1\ 1$	6	1
$x = 10\ 0\ 0$	4	4
$x = 10\ 0\ 1$	5	0

Cardinality of a code set ?

Hamming Distance Properties of Levenshtein Codes

- ***Proposition 1 :*** *A* Levenshtein code has only one code word of either weight $w = 0$ or weight $w = 1$.

Proposition 2 : In a Levenshtein code there is a minimum Hamming distance, $d_{min} \geq 2$ between any two code words.

Proposition 3 : Code words in a Levenshtein code have a $d_{min} \geq 4$ if they have the same weight.

- The proof of propositions is straight forward when considering the resulting subwords after s deletions.

Paluncic, F. ; Ferreira, H.C. ; Clarke, W.A., Abdel-Ghaffar, K.A.S. On Helberg's Generalization of the Levenshtein Code for Multiple Deletion/Insertion Error Correction. IEEE Information Theory. 2012

Our Inner Code: VT_6 code

- VT-codes with codewords of length k=6 according to eqn. sum($i*x_i$) mod 7=0 give dictionary **B** from VT_6 codes:

- **000→ [0 0 1 0 1 1];**
- **001 → [0 0 1 1 0 0];**
- **010 →[0 1 0 0 1 0];**
- **011 →[0 1 1 1 1 0];**
- **100 →[1 0 1 1 0 1];**
- **101 →[1 1 0 0 1 1];**
- **110 →[1 1 0 1 0 0];**
- **111 →[1 1 1 1 1 1];**
- **s → [1 0 0 0 0 1] – can be used for block synchronization**

- Special pattern s→ "000" used as suffix for the codeword to perform codeword synchronization

Example of Inner Code Decoding

- Marker helps to detect the codeword boundaries
- Iterative detection of the codewords performed with the use of so-called weighted Hamming cumulative distance
- For obtained codewords or their parts the LLR values of symbols is calculated and transferred to the outer turbo-decoder

5	6	6	6	5	6	5	6
0	1	1	0	0	1	1	0
6	5	6	6	6	6	4	6
0	0	1	0	0	0	2	0
7	16	25	34	43	52	61	70
6	15	24	33	40	50	58	67
7	15	24	33	42	51	58	67

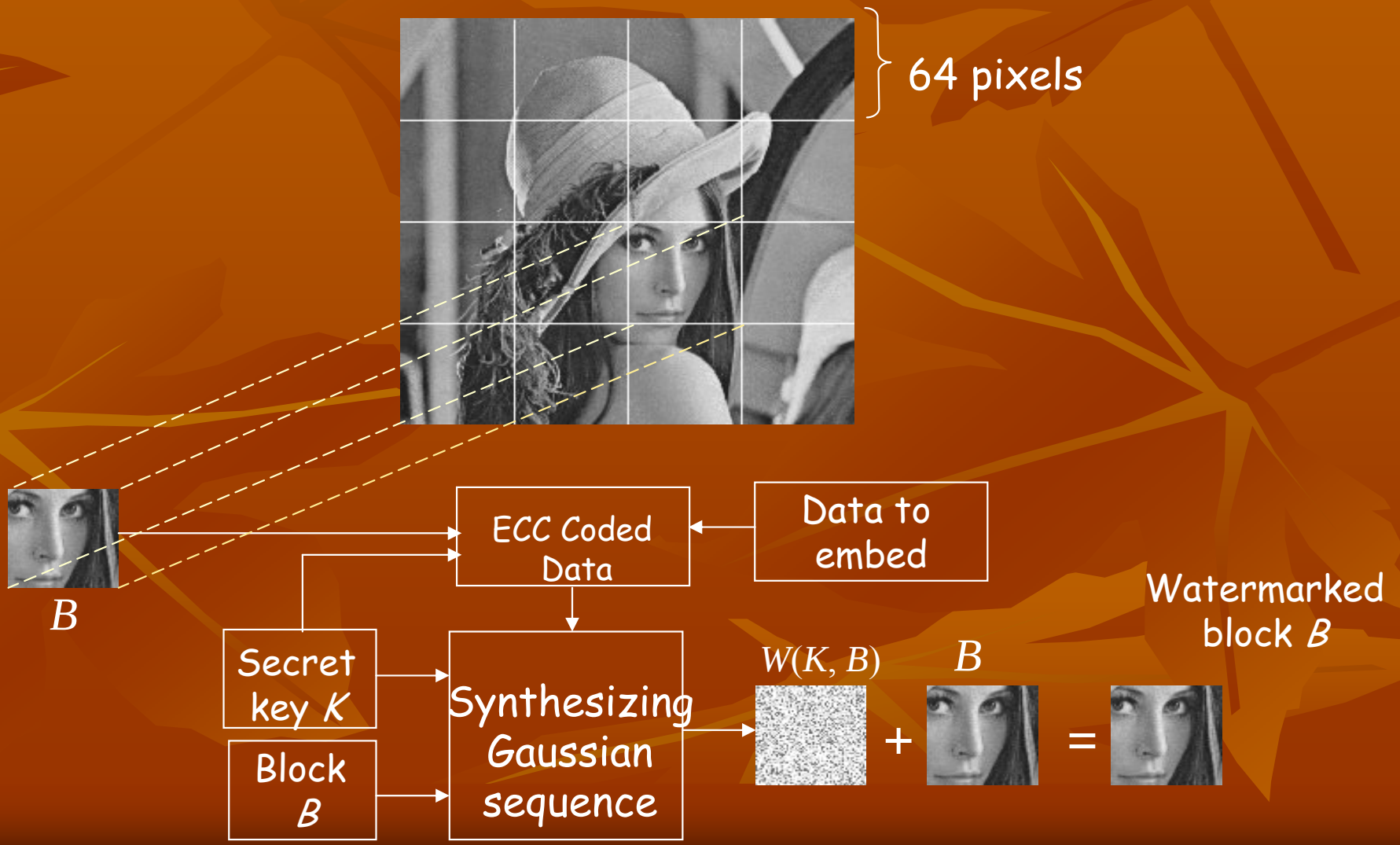
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	1	1
0	0	1	0	1	1
0	0	1	1	0	0
0	1	1	1	1	0
1	1	1	0	0	0
1	1	0	0	1	1

$$C_h = \sum_{c_i \in B, b_i \in r} \alpha_i d(b_i, c_i)$$

Step-by-step Data Processing for Watermark Embedding

- Get watermark data and convert it into one-dimensional binary sequence
- Divide sequence into segments and transform them into the elements of $GF(2^3)$
- Use outer turbo block code of rate 1/3 over $GF(2^3)$ to encode the input elements
- Convert turbo encoded elements into 3-bit words to perform Grei-coded 8PSK and encode them by VT_6 code with rate 1/2
- Add to 6-bit codewords a 3-bit marker code and obtain the data to be embedded containing the watermark
- Choose the image feature points invariant to known transformations and modulate image features in the predefined domain around FP

Prototype of Scheme for Data Embedding

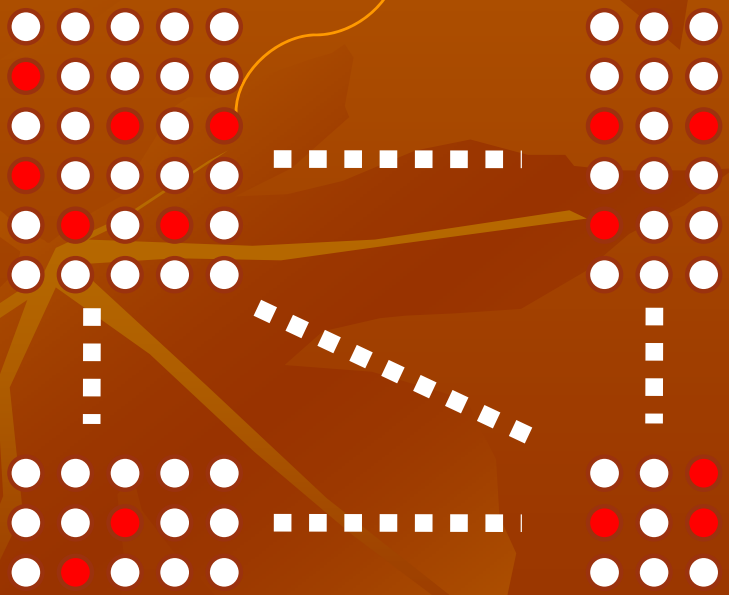


Selective Watermark embedding

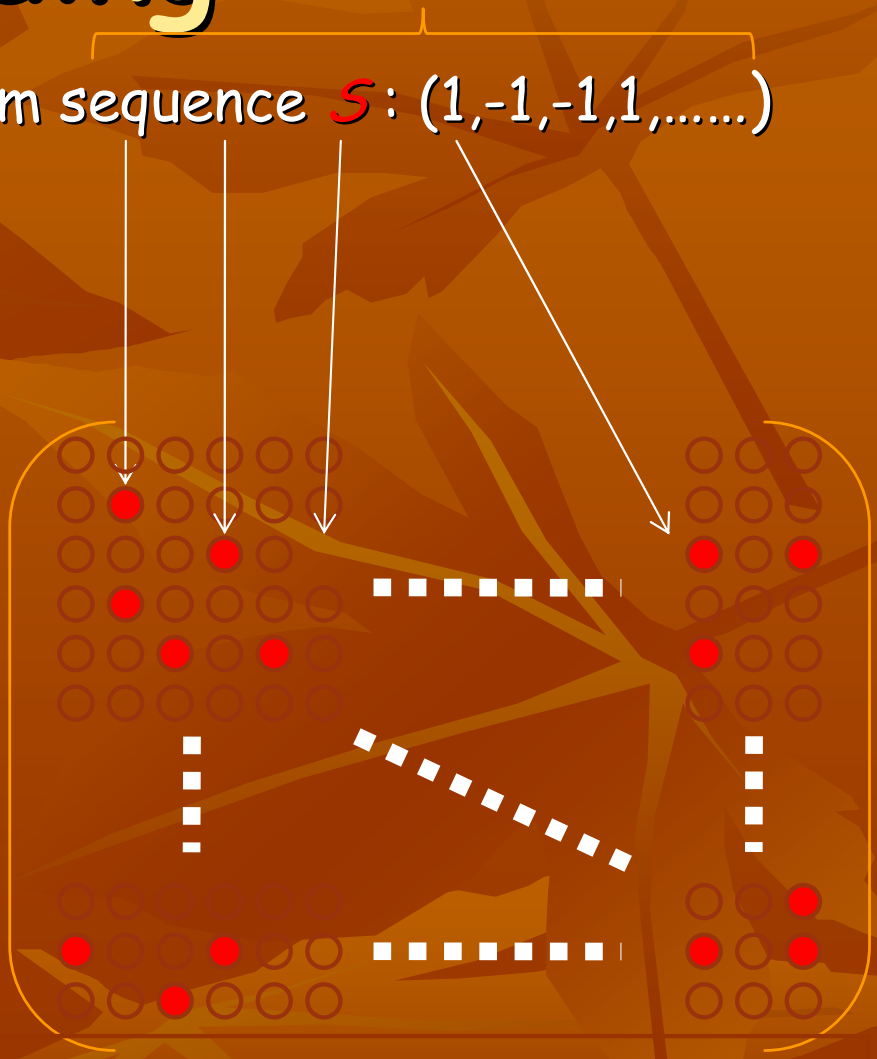
Watermark *W*: Random sequence *S*: (1,-1,-1,1,.....)

Selected area - block 8x8 around feature point

Red is feature point



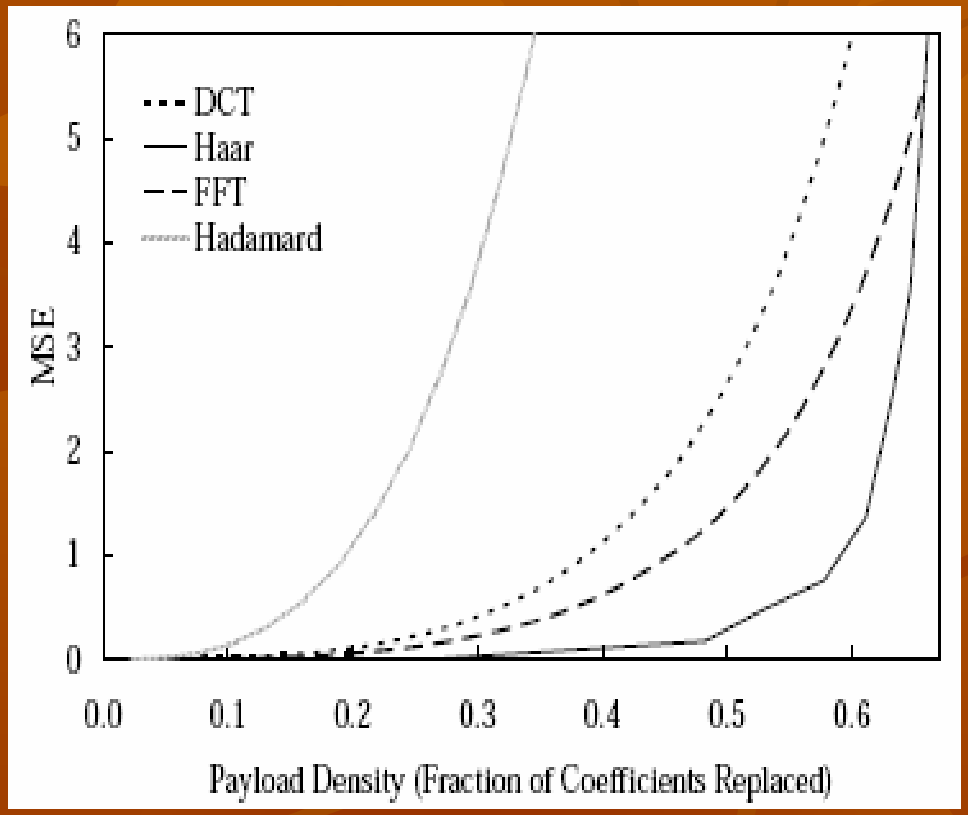
Original image



Watermarked image

FTT against DCT

The embedding data achieves the highest payload density with the lowest induced MSE when data is embedded in suitable transform. The experiments show that FFT outperforms DCT



Sos S. Agaian and Eric A. Silva, "*The Best Transform in the Replacement Coefficients and the Size of the Payload Relationship Sense*", IS&T Archiving Conference, April, San Antonio, pp. 199-203, 2004.

Arnold Transform

The security issues considered by the use of data permutation exploiting Arnold's mapping

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, x, y \in \{0,1,2,\cdots,N-1\}$$

Arnold's cat map named after Vladimir Arnold, who demonstrated its effects in the 1960s using an image of a cat

http://en.wikipedia.org/wiki/Arnold's_cat_map

Feature Points on Lena image

$$f(m,n) = \bigcup_{k=1}^{K} f_k(m',n'),$$

$$1 \le m', n' \le 8$$

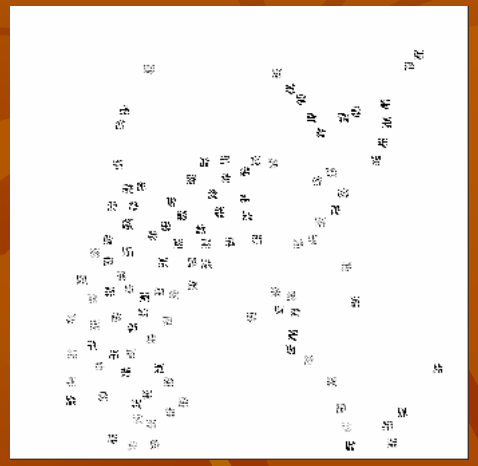
$$F_k(u',v') = DFT\{f_k(m',n')\}$$

$$1 \le u', v' \le 8$$

$$abs\_B(i,j) = abs\_B(i,j) \times$$

$$\times (1 + t * pn\_seq(l))$$

$$f_k'(m',n') = IDFT\{f_k(m',n')\}$$



Conclusions

- Harris FP detector has low complexity and invariant characteristics to several image transforms and could be used in our WM scheme
- Nonbinary TC over GF(8) have good performance and are to be exploited in concatenated coding-decoding scheme as the outer codes for erasures correction
- Inner VT code contain redundancy to eliminate additional errors and must be carefully designed to control synchronization of the codewords
- Concatenated coding-decoding scheme has soft-in soft-out iterative decoding properties and give a flexible implementation
- The use of DFT domain for FP-based watermarking provide selective data embedding and user defined image degradation preserving required secrecy

Future Research

- Make simulations of proposed scheme to study the robustness to noise and geometric attacks (scaling, rotation, etc)
- Test the quality of watermark detector based on DFT and check the false alarm probability
- Make the analysis of the other appropriate modulation techniques for watermark embedding
- Study the scale-invariant FP detectors, including Harris-Laplace, SIFT and the others
- Test the BER and SER of proposed error correcting technique and study the joint iterative soft decoding of inner and outer codes, including nonbinary codes
- Study the other methods of content-based synchronization for image watermarking

Thank you !!!

Questions?