Watermarking authentication based on the orthogonality of pseudo-random binary sequences

Manuel Graña,
Grupo de Inteligencia Computacional
UPV/EHU, www.ehu.es/ccwintco
Contents

• Introduction
• Algorithm features
• Watermark insertion and removal
• Empirical results
• Conclusions
Introduction

• Watermarking consists of the insertion of information (the watermark) inside the image.
• The watermark is desired to be invisible and robust.
  – It does not introduce perceptual changes in the image
  – It is not easy to remove, and
  – It can be recovered after the so-called attacks: lossy compression, cropping, smoothing, adding noise, etc.
Introduction

• Our watermarking procedure works on the coefficients of the Haar DWT.
• Insertion of the watermark
  – addition of pseudo-random binary sequences generated for each bit in the watermark to DWT coefficients selected according to their magnitude.
• The watermark extraction
  – testing the correlation of the pseudo-random binary sequences generated for the watermark bits with the selected DWT coefficients.
Algorithm features

- The watermark is a binary image,
- Each pixel in the watermark image is associated with a pair of pseudo-random binary number \{-1,1\} sequences.
Algorithm features

• The watermark insertion is performed on the difference coefficients:

\[(LH_n, HL_n, HH_n; n = 1, 2)\]
Algorithm features

• The watermark extraction is performed at each pixel independently,
  – through the regeneration of their associated pseudo-random binary \{-1, 1\} sequences.
  – we compare the correlation among the DWT selected coefficients and its associated pseudo-random binary \{-1,1\} sequences
Algorithm features

• For watermark extraction we require the knowledge of
  – the random number seed (the key in the figures below),
  – the position of the DWT coefficients affected by the watermark and
  – the watermark itself.
Algorithm features

• a key fact for our approach to work is that the pseudo-random binary sequences are (almost) orthogonal
DWT coefficient selection

1. All the \((LH_n, HL_n, HH_n; n = 1, 2)\) are sequenced into a vector \(Z\).

2. We initialize the selection threshold as the maximum absolute value of the coefficients.

3. We select the coefficients according to the threshold \(P = \{ j \mid Z(j) > \text{Threshold} \}\).

4. If \(|P| < n_r\) then decrease threshold and go to 3,

5. Save the values of the DWT coefficients \(C = \{ Z(j), j \in P \}\).
Watermark insertion

- Pseudo random sequence
  \[ \{ w_0(i) \in \{-1, 1\}, w_1(i) \in \{-1, 1\}, i = 1, \ldots, n_r \}. \]

- Modification of the selected DWT coefs
  \[
  C(i) = \begin{cases} 
  C'(i) + kw_0(i) & \text{black pixel} \\
  C'(i) + kw_1(i) & \text{white pixel}
  \end{cases}
  \]
Fig. 4: Watermark insertion

Wordcomp'09, Las Vegas, July 16, 2009
Watermark extraction

• We regenerate the pseudo-random sequences
• Recovery is performed computing the correlation

\[ W(p) = \begin{cases} 
0 & \text{corr}(C^*, C^0) > \text{corr}(C^*, C^1) \\
1 & \text{else}
\end{cases} \]
Fig. 5: Watermark extraction
Computational results
Fig. 9: Lena watermarked with $k = 0.3$. Correlation of the recovered watermark with the true one after JPEG lossy compression attacks
Fig. 10: Lena watermarked with $k = 0.5$. Correlation of the recovered watermark with the true one after JPEG lossy compression attacks
Fig. 11: Lena watermarked with $k = 0.3$. Correlation of the recovered watermark with the true one after cropping attacks.
Fig. 12: Lena watermarked with $k = 0.5$. Correlation of the recovered watermark with the true one after cropping attacks
Fig. 13: Lena watermarked with $k = 0.3$. Correlation of the recovered watermark with the true one after gaussian additive attacks.
Fig. 14: Lena watermarked with $k = 0.5$. Correlation of the recovered watermark with the true one after gaussian additive attacks.
Conclusions

• We present in this paper a watermak authentication procedure based on the DWT.
• The procedure tries to ascertain if the image contains a certain logo or binary image, given the original image and the watermarked image.
• The algorithm is based on the orthogonality of pseudo-random binary number sequences, so that storing information over a mark pixel does not interfere with others stored previously or in the future.
Conclusions

• We have tested is robusted with some encouraging success for the case of lossy compression and cropping, however the algorithm fails heavily when the attack consist of Gaussian noise addition.
• We need to do further computational experiments to test whole approach, also some improvements of the algorithm to correct the discovered problems when additive noise corrupts the images.