UNIVERSITÀ DEGLI STUDI DI CATANIA

DIPARTIMENTO DI MATEMATICA E INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

DIGITAL WATERMARKING CON DWT e SVD

PROGETTO DI MULTIMEDIA

Andrea Sequenzia
Matricola: W82000160

ANNO ACCADEMICO 2018/2019

Indice

1 Introduzione **3**

2 Definizione Watermarking **5**

2.1 Processo di Encoding 5

2.2 Processo di Decoding 6

2.3 Tipologie 6

3 Algoritmo robusto DWT-SVD **9**

3.1 Discrete Wavelet Transform 9

3.2 Singular Value Decomposition 10

3.3 Embedding 11

3.4 Extraction 11

4 Implementazione **12**

5 Esperimenti **18**

6 Conclusioni **18**

1 Introduzione

Il crescente successo di internet, della distribuzione digitale dei dati ed i progressi delle tecnologie di trasmissione dell'informazione hanno permesso di creare, replicare, trasmettere e distribuire contenuti digitali in modo semplice e veloce. La protezione e l'applicazione del diritto di proprietà intellettuale per i media digitali sono diventate una questione importante. La crittografia ha realizzato sistemi e protocolli in grado di garantire riservatezza, autenticità ed integrità delle comunicazioni e delle informazioni digitali ma non fornisce un modo per esaminare i dati dopo che essi sono stati distribuiti, inoltre i dati multimediali non contengono nessuna informazione sul proprietario originale e quindi nessuna protezione da copie non autorizzate, contraffazioni ecc... Per risolvere molti di questi problemi è nato il **digital watermarking**. In generale, un watermark è un'informazione impercettibile incorporata all'interno dei dati in modo che essa non possa essere rimossa. Può essere incorporato in qualsiasi tipologia di dato multimediale come audio, video e immagini ma in questa relazione verrà affrontata solo nel caso di immagini digitali. È difficile determinare quando si è iniziato a parlare di digital watermarking. Nel 1979, Szepanski descrive un machine-detectable pattern che può essere inserito all'interno di un documento per scopi di anti-contraffazione. Nove anni dopo, Holt *et al.* descrive un metodo per incapsulare un codice di identificazione in un audio digitale. Sono Komatsu e Tominaga che nel 1988 usano per la prima volta il termine digital watermarking [1] ma è negli anni 90 che il termine diventa popolare. Intorno al 1995 l'interesse verso il digital watermarking crebbe di molto come si può notare nella figura 1 che mostra il numero di articoli di ricerca riguardo a questo argomento. Nel 1996 viene tenuto il primo Information Hiding Workshop (IHW), dove uno degli argomenti primari era proprio il watermarking. In questo periodo molte aziende hanno considerato di inserire il digital watermarking all'interno dei propri standard. Alla fine degli anni 90 furono create diverse società che creavano software di digital watermarking, come ad esempio Digimarc che ha i propri embendder e detector all'interno di Adobe Photoshop [2]. L'antenato del watermarking è la steganografia, questo termine si riferisce all'arte di nascondere la comunicazione tra due interlocutori. Il messaggio è incorporato

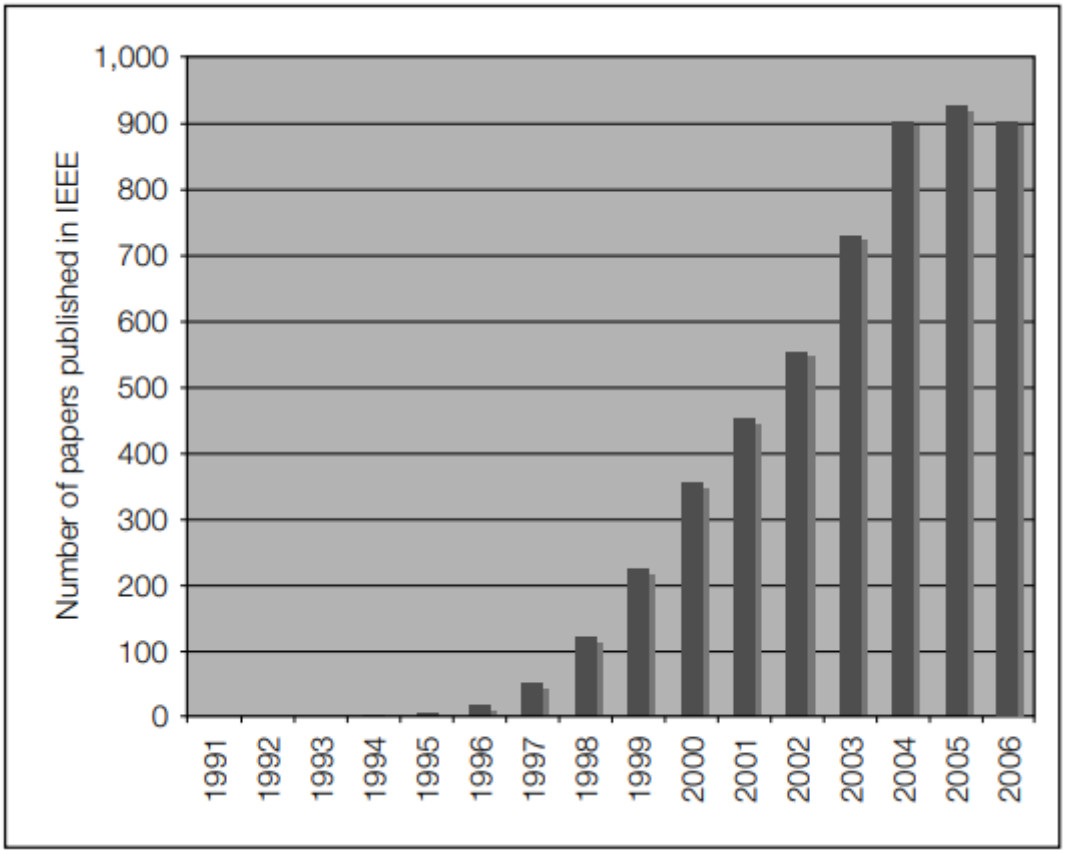

Figura 1: Numero annuale di papers pubblicati su watermarking e stenografia dall'IEEE

all'interno di un altro oggetto chiamato **cover work** e ne modifica certe proprietà [3]. La differenza fondamentale è che per la steganografia il documento è soltanto una maschera, quindi senza valore, mentre quello che vale è il messaggio nascosto al suo interno. Tale corrispondenza è invertita nel digital watermark in cui il messaggio nascosto non ha valore, visto che funge solo da protezione, mentre il documento è il vero portatore di valore. Principalmente il watermarking ha i seguenti scenari applicativi:

- Identificazione del proprietario
- Prova della proprietà
- Tracciare la distribuzione delle copie
- Content management nei social networks [4]

2 Definizione Watermarking

Il watermarking è il processo che incorpora (fase di **encoding**) dei dati chiamati **watermark** o **digital signature** in modo che tali dati possano essere rilevati o estratti in seguito (fase di **decoding**). In generale, ogni processo di watermarking è formato da tre parti:

- Il watermark
- L'encoder che esegue l'algoritmo di inserimento
- Il decoder ed il comparator, contengono gli algoritmi di detection, estrazione e verifica

2.1 Processo di Encoding

Sia I un'immagine, un watermark $S = s_1, s_2, ...$ e $\hat{I}$ l'immagine con il watermark incorporato. La funzione E, chiamata funzione di encoding, prende in input l'immagine I e il watermark S

per generare la nuova immagine $\hat{I}$. Formalmente:

$$E(I, S) = \hat{I}$$

In alcuni metodi viene anche proposto l'aggiunta di una chiave K per rendere più sicuro l'inserimento. Lo schema viene riformulato nel seguente modo:

$$E(I, K, S) = \hat{I}$$

2.2 Processo di Decoding

La funzione di decoding D prende in input un'immagine J che può essere una qualsiasi immagine o l'immagine marchiata (probabilmente anche corrotta) e restituisce il watermark S'. Se la funzione necessita dell'immagine originale I e del watermark originale S allora si parlerà di **private watermarking** (o **non-blind**), al contrario del **public watermark** (o **blind**) che non necessita di informazioni sui dati originali [5]. Formalmente:

$$D(J, I) = S'$$

Il watermark S' estratto viene confrontato con l'originale tramite una funzione comparator C_δ. Se questa funzione restituisce 1 allora ci sarà corrispondenza altrimenti no.

$$C_\delta(S, S') = \begin{cases} 1, c \leq \delta \\ 0, altrimenti \end{cases}$$

Dove C è il correlatore, $x = C_\delta(S', S)$, c è la correlazione tra due watermark e δ è una certa soglia [6].

2.3 Tipologie

I watermarks possono essere applicati nel **dominio spaziale** o nel **dominio delle frequenze**. È stato dimostrato che i watermarks applicati nel dominio delle frequenze sono più robusti, le principali tecniche sono riassunte nella figura 2.

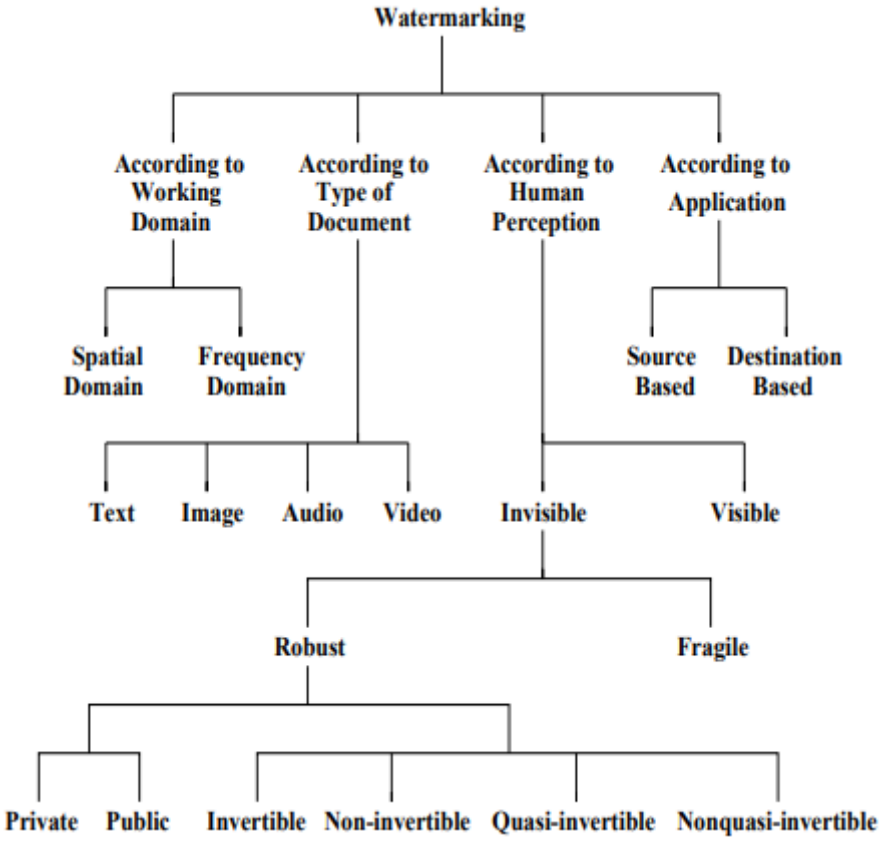

Figura 2: Principali tecniche per il watermarking

Figura 3: Un esempio di visible watermark

Sulla base della percezione umana, queste tecniche possono essere divise in quattro categorie differenti:

- Visible watermark
- Invisible-Robust watermark
- Invisible-Fragile watermark
- Dual watermark

I **visible watermark** sono semplicemente immagini semitrasparenti sovrapposte all'immagine originale. Questo tipo di watermarking viene usato sopratutto quando le immagini vengono rese pubbliche su internet ed il proprietario non vuole che le immagini vengano usate ad uso commerciale senza il pagamento delle royalties, oppure viene utilizzato per indicare il proprietario originale così da incoraggiare l'osservatore a tutelare chi detiene i diritti del prodotto. Gli **invisible-robust watermark** sono incorporati all'interno dell'immagine in modo che l'alterazione fatta all'immagine non sia percettivamente notata e possono essere recuperati solo tramite il proprio algoritmo di decodifica. Questo tipo di watermarking è utilizzato per individuare se l'immagine è stata rubata, in quanto chi l'ha distribuita illegalmente non riesce a vedere che all'interno di essa si trova un watermark che attesta che quell'immagine è protetta da copyright, ma anche per dimostrare la proprietà della stessa da parte del produttore originale. Gli **invisible-fragile watermark** sono incorporati all'interno dell'immagine in modo che qualsiasi alterazione fatta ad essa altererebbe o distruggerebbe il watermark. Uno scenario in cui viene utilizzato questo tipo di watermark è ad esempio quello di una camera affidabile. Un'agenzia di news potrebbe voler verificare se le immagini catturate siano vere oppure sono frutto di una falsificazione della scena, quindi al momento della cattura del filmato, un watermark invisible-fragile viene aggiunto e quindi si può verificare se esso è presente ancora successivamente. Infine i **dual watermark** sono una combinazione tra tra visible e invisible watermark [7]. In questo metodo viene incorporato un watermark invisible così da servire come riserva se quello

visible viene rimosso.

3 Algoritmo robusto DWT-SVD

In questa sezione verrà descritto l'algoritmo per implementare il processo di watermarking non-blind proposto nell'articolo *"Robust DWT-SVD domain image watermarking: embedding data in all frequencies"* [8]. In generale, questo metodo scompone l'immagine in quattro sottobande attraverso la DWT, applica SVD sia ad ogni sottobanda che ai dati che rappresentano il watermark, incorpora i dati del watermark in ogni sottobanda e infine attraverso la trasformata inversa, restituisce l'immagine marchiata.

3.1 Discrete Wavelet Transform

Nella discrete wavelet trasform, ogni livello di decomposizione produce quattro sottobande denotate come **LL** che è un'approssimazione dell'immagine originale, **HL** che contiene i dettagli orizzontali, **LH** che contiene i dettagli verticali e **HH** che contiene i dettagli diagonali. Alcuni articoli [9] hanno dimostrato che incorporare un watermark sia nelle alte che nelle basse frequenze può risultare un metodo molto resistente a vari tipi di attacco. In particolare, incapsulare dati nelle basse frequenze aumenta la robustezza nei confronti di attacchi che hanno caratteristiche low pass come compressione lossy o distorsioni geometriche ma rende il watermark più sensibile alla modifica dell'istogramma dell'immagine come la modifica del contrasto o della luminosità, correzione della gamma o equalizzazione dell'istogramma. Usare le alte e le medie frequenze, invece, rende più vulnerabili ai low-pass ma molto robusti rispetto all'aggiunta di rumori.

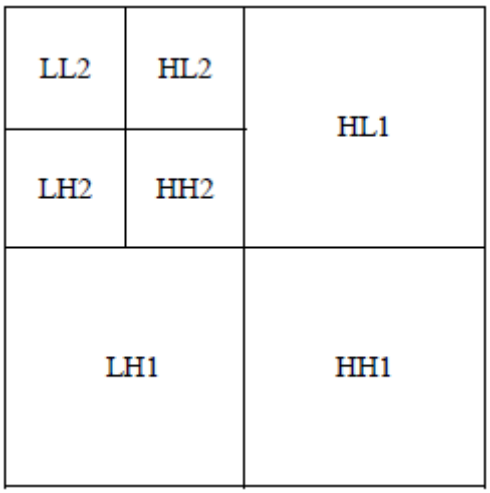

Figura 4: Decomposizione DWT a due livelli

3.2 Singular Value Decomposition

La singular value decomposition per le matrici quadrate è stata sviluppata indipendentemente da Beltrami nel 1873 e da Jordan nel 1874 ed è stata estesa per le matrici rettangolari da Eckart e Young nel 1930. Fino agli anni '60 non ha avuto applicazioni computazionali fino a quando Gene Golub ne ha dimostrato l'utilità in svariate applicazioni [10]. Adesso è usato in svariati campi dell'image processing come la compressione o appunto il watermarking. Formalmente, in algebra lineare, la decomposizione ai valori singolari è una particolare fattorizzazione di una matrice basata sull'uso di autovalori e autovettori. Ogni matrice A può essere decomposta come prodotto di tre matrici $A = U\Sigma V^T$, dove U e V sono matrici unitarie [1] e $\Sigma = diag(\lambda_1, \lambda_2, ...)$. I valori nella diagonale di Σ vengono detti **valori singolari** di A, ognuna delle colonne di U è detta **vettore singolare sinistro** e ognuna delle colonne di V è detta **vettore singolare destro**. Questa decomposizione può essere riscritta come:

$$A = \lambda_1 U_1 V_1^T + \lambda_1 U_2 V_2^T + ... + \lambda_r U_r V_r^T$$

[1] $U^T U = I$ e $V^T V = I$

dove r è il rango della matrice A. In un articolo [11] sono stati analizzati gli effetti delle ordinarie distorsioni geometriche nei valori singolari in un'immagine e grazie alle proprietà emerse da questa analisi, SVD è un ottimo strumento per la costruzione di metodi watermarking.

3.3 Embedding

Si assume che il watermarking sia nxn e l'immagine dove applicarlo sia $2nx2n$. I passi dell'algoritmo di embedding sono i seguenti:

1. Si applica la DWT sulla cover image A ottenendo 4 sottobande, si seleziona la sottobanda LL e si riapplica di nuovo la DWT ottenendo altre quattro sottobande.
2. Si applica SVD alle ultime 4 sottobande ottenute: $A^k = {U_a}^k {\Sigma_a}^k {V_a}^{kT}$ dove $k = 1, 2, 3, 4$ indica le 4 sottobande LL, HL, LH e HH, ${\lambda_i}^k, i = 1, ...n$ sono i valori singolari di ${\Sigma_a}^k$.
3. Si applica SVD al watermark: $U_W \Sigma_W {V_W}^T$ dove $\lambda_{wi}, i = 1, ...n$ sono i valori singolari di Σ_W.
4. Si modificano i valori singolari dell'immagine originale in ogni sottobanda con i valori singolari del watermark nel seguente modo: $\lambda_i^{*k} = {\lambda_i}^k + \alpha\lambda_{wi}, i = 1, ..., n$ e $k = 1, 2, 3, 4$.
5. Adesso si hanno 4 insiemi di coefficienti DWT: $A^{*k} = {U_a}^k \Sigma_a^{*k} {V_a}^{kT}, k = 1, 2, 3, 4$.
6. Infine si applica la trasformata inversa usando i coefficienti modificati, ciò produce la nuova immagine con all'interno il watermark.

3.4 Extraction

L'algoritmo di estrazione si suddivide nei seguenti passi:

1. Usando DWT, si decompone per due volte l'immagine di input A^* in quattro sottobande: LL, HL, LH e HH.

2. Si applica SVD ad ogni sottobanda ottenendo: $A^{*k} = {U_a}^k \Sigma_a^{*k} {V_a^k}^T, k = 1, 2, 3, 4$ dove k indica le sottobande

3. Si estraggono i valori singolari da ogni sottobanda: $\lambda_{wi}^k = (\lambda_i^{*k} - {\lambda_i}^k)/\alpha_k, i = 1, ..., n$ e $k = 1, 2, 3, 4$

4. Si ricostruiscono i quattro watermark usando i vettori singolari: $W^k = U_W {\Sigma_W}^k {V_W}^T, k = 1, 2, 3, 4$.

4 Implementazione

Per l'implementazione dell'algoritmo è stato usato **MATLAB**. Si inizierà a descrivere lo script *embed.h* che implementa la fase di embedding. Come prima cosa vengono caricati l'immagine in cui inserire il watermark (cover image) e il watermark (che viene convertito in scala di grigio).

```
% Caricamento cover image
I = imread('image.png');
% Caricamento watermark image
I_w = imread('watermark.png');
I_w = I_w(:,:,1);
```

Successivamente viene fatto il preprocessing sulla cover image, in particolare viene estratta una finestra nxn dove n deve essere divisibile per 4, questo perché durante la trasformata, le matrici ottenute vengono dimezzate e potrebbero sorgere problemi di matching tra matrici successivamente.

```
% Preprocessing cover image
original_I = I(:,:,1);
I = I(:,:,1);
```

```
size_I = size(I);
if (size_I(1) == size_I)
    % Caso immagine quadrata
    n = size_I(1);
elseif (size_I(2) > size_I(1))
    % Caso immagine  larga
    n = size_I(1);
else
    % Caso immagine alta
    n = size_I(2);
end
% Ricerca finestra divisibile per 4
while (mod(n,4)~=0)
    n = n - 1;
end
I = I(1:n,1:n,1);
```

Adesso si può eseguire l'algoritmo per l'embedding, nel seguente frammento di codice vengono eseguiti i passi 1 e 2 dell'algoritmo. Matlab fornisce l'implementazione della DWT 2D nel toolbox Wavelet [2] e il calcolo di SVD[3].

```
% DWT alla cover image
[LL,HL,LH,HH] = dwt2(I,'haar');
[LL2,HL2,LH2,HH2] = dwt2(LL, 'haar');

% SVD per ogni sottobanda
```

[2] https://it.mathworks.com/help/wavelet/ref/dwt2.html

[3] https://it.mathworks.com/help/matlab/ref/double.svd.html

```
[Ui_LL, Si_LL, Vi_LL]   = svd(LL2);
[Ui_HL, Si_HL, Vi_HL]   = svd(HL2);
[Ui_LH, Si_LH, Vi_LH]   = svd(LH2);
[Ui_HH, Si_HH, Vi_HH]   = svd(HH2);
```

Adesso si passa al preprocessing del watermarking che dovrà avere la stessa dimensione dell'immagine approssimata dopo l'applicazione della trasformata e successivamente viene eseguito il passo 3 dell'algoritmo.

```
% Preprocessing watermark
I_w = imresize(I_w, size(Ui_LL));

% SVD watermark
[Uw, Sw, Vw] = svd(double(I_w));
```

Per il passo 4 bisogna scegliere un valore per α, essi sono scelti basandosi sul fatto che i coefficienti più grandi si trovano nella sottobanda LL e i più bassi nella sottobanda HH. Come spiegato nell'articolo, vengono usati due valori per α, uno per la sottobanda LL e un valore piccolo per le altre sottobande, in questa implementazione sono stati scelti rispettivamente 0.005 per la sottobanda LL e 0.008 per le altre [4].

```
% Embedding
SLL_emb = Si_LL + 0.05*Sw;
SHL_emb = Si_HL + 0.008*Sw;
SLH_emb = Si_LH + 0.008*Sw;
SHH_emb = Si_HH + 0.008*Sw;
```

[4] Valori più grandi danno migliori risultati nella qualità del watermarking che verrà estratto ma degraderanno di più l'immagine marchiata

Infine vengono eseguiti i passi 5 e 6 dell'algoritmo che genereranno l'immagine marchiata finale.

```
% Calcolo della matrice A
LL_new = Ui_LL*SLL_emb*Vi_LL';
HL_new = Ui_HL*SHL_emb*Vi_HL';
LH_new = Ui_LH*SLH_emb*Vi_LH';
HH_new = Ui_HH*SHH_emb*Vi_HH';

% IDWT sulla nuova A
F_pass = idwt2(LL_new,HL_new,LH_new,HH_new,'haar');
I_out = idwt2(F_pass,HL,LH,HH,'haar');
I_out = uint8(I_out);
% Ricostruzione immagine
original_I(1:n,1:n,1) = I_out(:,:,1);
imwrite(original_I,'watermarked.png');
```

Adesso verrà descritto lo script *extract.m* che implementa l'algoritmo di estrazione dei 4 watermark. Come prima cosa vengono caricati l'immagine marchiata, l'immagine originale ed il watermark originale (metodo non-blind).

```
% Caricamento immagine marchiata
I = imread('watermarked.png');
I = I(:,:,1);
% Caricamento watermark originale
I_w = imread('watermark.png');
I_w = I_w(:,:,1);
% Caricamento immagine originale
cover_I = imread('image.png');
```

Per gli stessi motivi spiegati per lo script *embed.h*, viene effettuato il preprocessing dell'immagine originale e di quella marchiata.

```
% Preprocessing
original_I = cover_I(:,:,1);
cover_I = cover_I(:,:,1);
size_I = size(cover_I);
disp(size_I);
if (size_I(1) == size_I)
    n = size_I(1);
elseif (size_I(2) > size_I(1))
    n = size_I(1);
else
    n = size_I(2);
end
% Ricerca finestra divisibile per 4
while (mod(n,4)~=0)
    n = n - 1;
end
cover_I = cover_I(1:n,1:n,1);
I = I(1:n,1:n,1);
```

Infine si esegue l'intero algoritmo di estrazione ottenendo come output 4 watermark: **W1, W2, W3, W4**.

```
% DWT
[LL1_wmk,HL1_wmk,LH1_wmk,HH1_wmk] = dwt2(I, 'haar');
[LL2_wmk,HL2_wmk,LH2_wmk,HH2_wmk] = dwt2(LL1_wmk, 'haar');
```

```
% SVD per i 4 watermark
[U1_wmk,S1_wmk,V1_wmk] = svd(LL2_wmk);
[U2_wmk,S2_wmk,V2_wmk] = svd(HL2_wmk);
[U3_wmk,S3_wmk,V3_wmk] = svd(LH2_wmk);
[U4_wmk,S4_wmk,V4_wmk] = svd(HH2_wmk);

% DWT e SVD dell'immagine originale
[LL,HL,LH,HH] = dwt2(cover_I,'haar');
[LL2,HL2,LH2,HH2] = dwt2(LL, 'haar');
[Ui_LL,Si_LL,Vi_LL]  = svd(LL2);
[Ui_HL,Si_HL,Vi_HL]  = svd(HL2);
[Ui_LH,Si_LH,Vi_LH]  = svd(LH2);
[Ui_HH,Si_HH,Vi_HH]  = svd(HH2);

% Extract
S1 = (S1_wmk - Si_LL)/0.05;
S2 = (S2_wmk - Si_HL)/0.008;
S3 = (S3_wmk - Si_LH)/0.008;
S4 = (S4_wmk - Si_HH)/0.008;

% Calcolo delle matrici A
I_w = imresize(I_w, size(Ui_LL));
[Uw, Sw, Vw] = svd(double(I_w));
W1 = Uw*S1*Vw';
W2 = Uw*S2*Vw';
W3 = Uw*S3*Vw';
W4 = Uw*S4*Vw';
```

5 Esperimenti

6 Conclusioni

Riferimenti bibliografici

[1] N. Komatsu and H. Tominaga, "Authentication system using concealed image in telematics," *Memoirs of the School of Science and Engineering*, 1988.

[2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, "Digital watermarking and steganography," *Morgan Kaufmann*, 2008.

[3] H. V. Desai, "Steganography, cryptography, watermarking: A comparative study," *Journal of Global Research in Computer Science*, vol. 3, December 2012.

[4] A. Zigomitros1, A. Papageorgiou, and C. Patsakis, "Social network content management through watermarking," *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012.

[5] N. Rani, "Digital watermarking," *Global Journal of Computer Science and Technology Graphics & Vision*, vol. 12, 2012.

[6] S. Mohanty, "Digital watermarking: A tutorial review," 05 2003.

[7] S. Mohanty, K. Ramakrishnan, and M. Kankanhalli, "A dual watermarking technique for images," *ACM Multimedia*, vol. 2, 10 2002.

[8] E. Ganic and A. M. Eskicioglu, "Robust dwt-svd domain image watermarking: embedding data in all frequencies.," pp. 166–174, 01 2004.

[9] M. S. Raval and P. P. Rege, "Discrete wavelet transform based multiple watermarking scheme," vol. 3, pp. 935–938 Vol.3, Oct 2003.

[10] M. Gupta, "Numerical methods and software (david kahaner, cleve moler, and stephen nash)," *Siam Review - SIAM REV*, vol. 33, 03 1991.

[11] B. Zhou and J. Chen, "A geometric distortion resilient image watermarking algorithm based on svd," *Chinese Journal of Image and Graphics*, vol. 9, 01 2004.