Capitolo 5

Watermarking e Manipolazione dei Colori

In questo capitolo viene presentato un nuovo efficiente schema di *watermarking* su immagini digitali basato sulla manipolazione dei colori. L'idea di fondo che sta alla base di tale algoritmo risiede nella possibilità di alterare impercettibilmente i colori di un'immagine inserendo così un marchio digitale che risulta essere robusto rispetto alle più comuni operazioni di *image processing*. Tale algoritmo viene poi analizzato, sempre in merito alla robustezza dello schema, anche dal punto di vista teorico-statistico.

1. Introduzione

Lo sviluppo di Internet come mezzo di comunicazione per così dire universale ha aperto moltissime e svariate possibilità ed opportunità nell'ambito della

comunicazione e nella trasmissione dell'informazione, quest'ultima considerata nella sua accezione più generale. D'altra parte diverse sono le problematiche più specificatamente applicative che ci si trova ad affrontare e che questo scenario ha completamente rivoluzionato soprattutto per quanto riguarda la titolarità del diritto di proprietà associato ad un qualsivoglia dato digitale. Risulta chiaro infatti che, trattandosi di dati digitali, è possibile realizzare delle copie, non solo in maniera banale ma soprattutto senza alcuna percettibile degradazione. Si riesce cioè ad ottenere copie fedeli dell'originale, con tutte le problematiche legali che questo comporta, in assoluta libertà e semplicità.

In questo contesto si inserisce il *watermarking* digitale che può essere definito come il tentativo di arginare tale fenomeno inserendo appunto un "marchio" all'interno di dati digitali, siano essi immagini, suoni o video, che identifichi, in maniera univoca, il legittimo proprietario o la provenienza specifica di quei dati. Ciò rende il *watermarking* una delle tecniche più attuali nella lotta alla pirateria digitale. Tale marchio però, per assolvere il suo compito deve essere al tempo stesso impercettibile e robusto. Questi, che sembrano essere due criteri per certi versi opposti sono comunque requisiti necessari ed indispensabili al fine di garantire un effettiva utilità pratica del processo di *watermarking* digitale.

Esistono diversi criteri e parametri per classificare i vari schemi di *watermarking* oggi presenti e ciò è dovuto principalmente alla vasta sfera di applicazioni e settori specifici potenzialmente coinvolti. Per una completa e vasta panoramica sui vari aspetti del *watermarking* digitale si veda (Acken - 1998; Bell, Braudaway, Mintzer - 1998; Craver, Yeo, Yeung - 1998; Luo, Koch, Zhao - 1998; Memon, Wah, Wong - 1998; Yeung - 1998). Nel successivo paragrafo verranno illustrati alcuni concetti di

base con riferimento anche ad alcune delle tecniche presenti in letteratura rivolte al *watermarking* di immagini digitali.

L'idea di base dell'algoritmo proposto in questo capitolo è quella di inserire il cosiddetto "marchio di acqua" o *watermark* in una data immagine in maniera sì impercettibile ma allo stesso tempo in un'opportuna forma che ne garantisca la facile individuabilità e la corrispondente robustezza ad attacchi di tipo malizioso e non. In un certo senso il metodo proposto è simile alla tecnica introdotta da Cox, Kilian, Leighton e Shamoon (1997) (si veda al riguardo il successivo paragrafo). Tale tecnica prevede di inserire il marchio impercettibilmente in alcune frequenze dello spettro del segnale da marchiare. Nel nostro caso invece si opera direttamente in un opportuno spazio dei colori dell'immagine e precisamente lo spazio della *Color Opponency* (Netravali, Haskell - 1988), editando in maniera impercettibile i valori della tavolozza dei colori ed inserendo in maniera univoca e non ambigua un marchio. Lo schema proposto è robusto rispetto ad una notevole classe di operazioni tipiche nella manipolazione di immagini digitali. La robustezza dell'algoritmo è analizzata anche da un punto di vista più strettamente teorico che ne evidenzia fra l'altro la dipendenza dal numero di colori dell'immagine. Il principale vantaggio di tale approccio è innanzitutto legato all'efficienza computazionale in quanto si evita di dover calcolare la DCT o la FT del segnale di *input*.

Eventuali operazioni sulla LUT (*Look Up Table*) dell'immagine riescono a rimuovere completamente il marchio dell'immagine solo a prezzo di una notevole degradazione della qualità dell'immagine stessa. A conferma di ciò si vedano l'analisi statistica e le evidenze sperimentali descritte nei successivi paragrafi.

2. Nozioni di base

In questo paragrafo vengono brevemente presentati alcuni dei parametri con i quali classificare i vari schemi di *watermarking*. Innanzitutto un *watermark* può essere *percettibile* o *impercettibile*. Nel primo caso ovviamente, viene utilizzato per codificare informazioni che devono essere rese pubbliche all'utente finale (identificativi del proprietario, istruzioni particolari, ecc.). Il *watermark impercettibile* è invece utilizzato in quei contesti in cui il proprietario legittimo vuole garantirsi tale diritto nascondendo tale marchio nell'immagine. In pratica, in questo caso, la copia marcata dell'immagine è quasi identica all'originale. E' però sufficiente garantire la sostanziale impercettibilità del marchio ad un osservatore umano; ciò infatti da una lato preserva la qualità percettiva della specifica informazione digitale in questione (immagine, video, suono, ecc.), dall'altra consente di inserire opportunamente il marchio stesso.

Uno schema di *watermarking* può essere *fragile* o *robusto*. *Watermark* fragili possono essere facilmente attaccati, distrutti o resi irriconoscibili da quasi ogni tipo di lecita manipolazione dei dati. Essi sono utili soprattutto nei cosiddetti processi di autenticazione (dato che anche "piccoli" cambiamenti possono danneggiare il marchio). D'altra parte *watermark* robusti devono resistere alle più comuni operazioni/trasformazioni sui dati, per esempio nel caso di immagini digitali al filtraggio, alla compressione *lossy* e non, alla quantizzazione, al *cropping*, al *resizing*, allo *scaling*, ecc.. Inoltre si tende a considerare un *watermark* robusto quando è in grado di resistere anche ad attacchi intenzionali o *malicious*, volti appunto alla rimozione del *watermark*. Chiaramente i *watermark* robusti hanno senso e sono utili nei contesti in cui la proprietà deve essere provata o garantita. Negli

ultimi anni diverse tecniche e schemi sono stati proposti con alterne fortune. A tutt'oggi non si è ancora trovato un metodo per così dire universale; anche gli schemi che sembravano essere più robusti e promettenti hanno mostrato i loro limiti non appena sono stati sottoposti ad attacchi intenzionali *ad-hoc*. La stessa sorte è toccata anche ad una serie di *software* commerciali che promettevano la sostanziale inattaccabilità e robustezza del marchio digitale da loro prodotto. Si veda in proposito il sito *web* (Petitcolas - 1999) per una panoramica al riguardo.

Uno dei primi schemi è stato proposto da (Van Schyndel, Tirkel, Osborne - 1994). L'idea di base è quella di inserire il segnale di *watermark* (una sequenza di *bit* pseudo casuali) nel *bit* meno significativo dell'immagine originale, *pixel* per *pixel*. Chiaramente questo schema non è abbastanza robusto: basta agire su tale *bit* e rimuovere così facilmente il relativo marchio.

Altre note tecniche apprezzabili ma comunque non sufficientemente robuste si trovano in (Bender, Gruhl, Morimoto - 1995; Brassil, Low, Maxemchuk, O'Gorman - 1994; Caronni - 1995; Kock, Rindfrey, Zhao - 1994; Macq, Quisquater - 1995; Matsui, Tanaka - 1994; Tanaka, Nakamura, Matsui - 1990).

Il lavoro di Cox et alii.(1997) sembra essere quello che al momento offre maggiori garanzie dal punto di vista strettamente legato alla robustezza. In particolare gli autori affermano che per garantire la robustezza di un qualunque schema di *watermarking* si deve necessariamente agire sulle componenti percettivamente più significative. Tale approccio sembra contraddire la fondamentale necessità di trattare solo ed esclusivamente marchi impercettibili. In realtà si tratta solamente di trovare il giusto bilanciamento tra le due esigenze sfruttando, nel loro specifico caso, alcune proprietà dello spettro del segnale digitale in questione. Tale tecnica è robusta e

sicura rispetto alle più comuni operazioni di elaborazione dei segnali, alle distorsioni geometriche e anche rispetto ad alcuni più sofisticati attacchi intenzionali.

3. Spazi di colore

La tecnica di *watermarking* presentata in questo capitolo opera direttamente sullo spazio dei colori dell'immagine. In particolare l'algoritmo richiede che per tale spazio vengano soddisfatte le due seguenti condizioni:

i) Deve esistere un semplice e compatto modo di descrivere geometricamente in tale spazio, piccole regioni percettivamente uniformi;

ii) La trasformazione dallo spazio RGB standard a tale spazio deve essere veloce e facilmente invertibile;

L'algoritmo si occupa di perturbare in maniera impercettibile la tavolozza dei colori, seguendo alcune semplici regole geometriche descritte nei paragrafi successivi. In particolare ciascun colore è *spostato* all'interno di una sfera percettivamente uniforme. Il comune spazio RGB, sfortunatamente, non è tale da garantire la presenza di tali regioni: a piccole variazioni all'interno di tale sfera potrebbero corrispondere delle alterazioni percettive non omogenee. La necessità di avere a che fare con spazi di colore percettivamente uniformi è stato da sempre uno dei temi caldi della *Computer Graphics*, vedasi al riguardo (Foley, Van Dam, Feiner - 1990; Glassner - 1995) e diverse soluzioni sono state proposte: *L*u*v**, *L*a*b**, *YIQ*, ecc.. Le trasformazioni dallo spazio RGB a tali spazi sono generalmente non lineari, difficili da invertire, e ciò che è peggio per la nostra specifica applicazione introducono una sorta di ri-quantizzazione dei colori. In pratica violano una delle due

proprietà da noi individuate e precisamente la ii). Per tali ragioni anche se tali spazi risultano essere la migliore alternativa ideale all'RGB, si è scelto di operare nello spazio CO, *Color Opponency* (Netravali, Haskell - 1988). Questa scelta garantisce una semplice e facilmente invertibile relazione tra i due spazi coinvolti ed è descritta dalle seguenti equazioni:

RGB ® CO

$$A = R + G + B;$$

$$B/Y = 2B - R - G;$$

$$R/G = R - 2G + B. \qquad (1)$$

CO ® RGB

$$R = (A + R/G - B/Y)/3;$$

$$G = (A - R/G)/3;$$

$$B = (B/Y + A)/3. \qquad (2)$$

Esistono inoltre chiare evidenze sperimentali nel campo delle scienze cognitive che indicano tale spazio come una delle migliori scelte per l'occlusione di piccole perturbazioni di colore, almeno ad un osservatore umano. Il modello della *Color Opponency*, è fra l'altro molto vicino alla struttura dei canali cromatici del sistema visivo umano. Un'altra prova, sia pur indiretta, della effettiva utilità ed efficacia della *Color Opponency*, ci viene dal fatto che tale modello è utilizzato in molti algoritmi volti al riconoscimento di specifiche tessiture o colori, in ambiti più strettamenti legati alla percezione umana (Fleck, Forsyth, Bregler - 1996). Nei paragrafi che seguono si assume che i colori siano già rappresentati nello spazio CO. Le conversioni dallo spazio RGB vengono eseguite in *pre-processing*.

4. Gli Algoritmi

Un qualunque schema di *watermarking* è realizzato attraverso l'implementazione di due ben specifici algoritmi. Uno di inserimento del marchio, che prende in *input* l'immagine originale e ne restituisce in *output* la relativa immagine opportunamente marcata e il marchio vero e proprio. Tale marchio andrà inserito nel *database* che identifica gli acquirenti dell'immagine stessa. Deve inoltre essere presente un algoritmo di *detection*, che presa in *input* l'immagine originale, un immagine marchiata e il *database* di cui sopra, restituisca il marchio associato alla data immagine sotto osservazione. L'idea che sta alla base dello schema qui presentato è quella di associare a ciascun colore dell'immagine (un punto nello spazio CO) una sfera centrata su tale colore. I punti della sfera rappresentano colori che sono percettivamente indistinguibili rispetto al colore originale. La lunghezza del raggio di tale sfera viene individuato di volta in volta, in maniera diretta, sperimentalmente ed è fortemente legato alla particolare immagine da marchiare.

4.1 L'Algoritmo di Inserimento

L'algoritmo per l'inserimento del marchio si occupa innanzitutto di classificare/partizionare i *pixel* dell'immagine in oggetto in base ai rispettivi colori. Successivamente ciascun colore viene modificato, in maniera casuale ma senza alterarne le caratteristiche percettive. In pratica per ciascun colore, il punto colore nello spazio in oggetto ad esso associato, viene spostato lungo una certa direzione fino a raggiungere il bordo della propria sfera di colore. La direzione è scelta in maniera *random* per ciascun colore presente nell'immagine.

Mark-Insert

Input: Un'immagine ***I***, data come tre matrici nello spazio della *Color Opponency A[i , j], R/G[i , j], B/Y[i , j]* dove (*i,j*) è un singolo *pixel*.

Output:
Un'immagine marchiata ***I'***, data come *A'[i , j], R/G'[i , j], B/Y'[i , j]*.
Un *marchio* dato come un vettore di piani *p[k], k =1, …, N* dove *N* è il numero di colori dell'immagine.

1. *For each k =1, …, N*
   a) *Classifica i pixel/colore;*
      Indichiamo con COL_k l'insieme dei pixel (*i,j*) che hanno lo stesso colore *k*.
      Sia inoltre (x_k, y_k, z_k) il punto dello spazio/colore associato a *k*.
   b) *Seleziona random una direzione;*
      Scegli in maniera casuale un raggio nella sfera di colore centrata in (x_k, y_k, z_k) e sia (x'_k, y'_k, z'_k) il punto di tale raggio giacente sul bordo della sfera.
   c) *Sposta tutti i pixel in COL_k nello stesso punto nello spazio della Color Opponency;*
      For each pixel (*i,j*) $\in$ COL_k poni:
      $A'[i,j] = x'_k$
      $R/G'[i,j] = y'_k$
      $B/Y'[i,j] = z'_k$
      End For.
   d) *Salva il piano normale alla direzione dello spostamento;*
      Sia *p[k]* il piano normale al raggio scelto e passante per il suo punto medio.
   End For.
2. *Return **I'** = A'[i , j], R/G'[i , j], B/Y'[i , j]* come immagine marchiata.
   Salva *MARK =[p[1], p[2], …, p[N]]*.

Fig 1. Algoritmo di Inserimento

Più in dettaglio, ogni data immagine ***I***, viene rappresentata da tre distinte matrici nello spazio della *Color Opponency, A[i , j], R/G[i , j], B/Y[i , j]* (cioè le coordinate del colore del *pixel (i,j)*). Se l'immagine ha *N* colori, indichiamo con COL_K, con $1 \leq k \leq N$, l'insieme dei *pixel* (*i,j*) che hanno lo stesso colore e di conseguenza le stesse coordinate $A[i,j] = x_k$, $R/G[i,j] = y_k$, $B/Y[i,j] = z_k$ nello spazio CO in oggetto.

Per ciascun *k* scegliamo, in maniera del tutto casuale, un raggio della sfera di colore di centro (x_k, y_k, z_k) e consideriamo il piano *p[k]* perpendicolare al raggio e

passante per il suo punto medio. Tutti i *pixel* in COL_K vengono spostati alla fine del raggio, dal lato opposto rispetto al piano *p[k]* testé individuato.

In pratica si sono cambiate le coordinate geometriche dei colori di tutti i *pixel* $(i,j) \in COL_K$ in $A'[i , j]= x'_k$, $R/G'[i , j]= y'_k$, $B/Y'[i , j]= z'_k$, dove (x'_k, y'_k, z'_k) è ovviamente il punto sul bordo della sfera.

Il risultato di questo processo algoritmico è l'immagine marcata detta *watermarked*. Il marchio è quindi costituito da un vettore di piani *p[k]*. E' possibile ridurre lo spazio fisico occupato da ciascun marchio attraverso l'utilizzo di generatori di numeri pseudo-casuali forti (*Pseudo-random number generators* - PRNG). In particolare dato un *seed s*, si generano univocamente i coefficienti del piano da utilizzare nello spazio CO. E' quindi sufficiente memorizzare fisicamente solo la stringa *s*, passando il compito di ricostruirsi i piani *[p[1], p[2], …, p[N]]* all'algoritmo di *detection.*

L'algoritmo di inserimento viene schematizzato in Fig. 1.

4.2 L'Algoritmo di *Detection*

Come noto l'algoritmo di *detection*, riceve in *input* un'immagine con il marchio, l'immagine originale e la lista, spesso organizzata in un vero e proprio *database*, dei cosiddetti marchi leciti. Assumiamo per il momento che l'immagine da analizzare abbia lo stesso numero di colori dell'immagine originale. L'algoritmo di *detection* confronta l'immagine con ciascuno dei marchi presenti nel *database*. Dato quindi un marchio *MARK =[p[1], p[2], …, p[N]]*, l'idea di base è quella di analizzare ciascun colore COL'_k dell'immagine ricevuta e controllare se il punto colore associato ad esso di coordinate (x'_k, y'_k, z'_k), si trovi o meno sul lato opposto del piano *p[k]*

(rispetto all'originale centro della sfera per COL_k e precisamente (x_k, y_k, z_k)). In caso positivo si incrementa un opportuno contatore. L'algoritmo resituirà in *output* il marchio che avrà riportato il più alto valore. Una descrizione più formale dell'algoritmo di *detection* è presente in Fig.2. Nel successivo paragrafo presentiamo invece un'accurata analisi statistica che prova come con alta probabilità venga sempre individuato il marchio corretto. Si osservi che l'algoritmo di *detection*, potrebbe ricevere un immagine manipolata da un cosiddetto *malicious attacker*, un attaccante "malizioso" detto tecnicamente l'avversario. In questo caso non è affatto detto che il numero di colori sia comunque rimasto lo stesso. In particolare l'avversario potrebbe "spostare" ciascun *pixel*, all'interno di ciascuna classe COL_k in una differente locazione dello spazio. D'altra parte per non deteriorare troppo l'immagine l'avversario deve comunque rimanere all'interno della sfera di impercettibilità relativa. Ciò significa che è comunque sempre possibile ridursi al caso in cui l'immagine marchiata abbia lo stesso numero di colori dell'originale. Si potrebbe, per esempio, adottare la tecnica di compattare i *pixel* mossi dall'avversario nel loro baricentro. Un'altra possibilità sarebbe quella di incrementare il *counter* solo quando un dato numero di *pixel* (una volta fissato una soglia) appartenga allo stesso lato del piano. Possiamo quindi assumere senza perdere di generalità, che l'immagine ricevuta per l'estrazione ed individuazione del marchio abbia esattamente N colori e le classi/colori COL_k siano rispettivamente uguali a quelle dell'immagine originale, almeno percettivamente. L'unica differenza sarà data dalle coordinate nello spazio dei colori presenti rispettivamente nell'immagine originale e in quella marchiata.

Mark-Detect

Input:
L'immagine originale $\boldsymbol{I} = A[\,i\,,j], R/G[\,i\,,j], B/Y[\,i\,,j]$.
Un'immagine marchiata $\boldsymbol{I'} = A'[\,i\,,j], R/G'[\,i\,,j], B/Y'[\,i\,,j]$.
La lista dei marchi $MARK_n=[\,p_n\,[\,1\,], p_n\,[\,2\,], \ldots, p_n\,[\,N\,]\,]$ *per* $n = 1, 2, \ldots, M$ dove M è il numero totale di immagini originariamente marchiate.
Il numero di colori dell'immagine N.

Output:
Un marchio $MARK_{Id}$.

1. Poni $max=0$ e $Id=0$
2. *For each* $n = 1, \ldots, M$
   a) Poni il contatore $C_n=0$
   b) *For each* $k=1, \ldots, N$
      Siano (x'_k, y'_k, z'_k) le coordinate del colore relativo ai pixel dell'immagine marchiata in COL_k.
      Siano (x_k, y_k, z_k) le coordinate del colore relativo ai pixel dell'immagine originale in COL_k.
      Se (x_k, y_k, z_k) e (x'_k, y'_k, z'_k) stanno su lati opposti rispetto al piano $p_n\,[\,k\,]$ allora $C_n = C_n + 1$.
      End For.
   c) Se $C_n > max$ *allora poni* $max = C_n$ *e* $Id=n$
   End For.
3. Output $MARK_{Id}$.

Fig 2. Algoritmo di *Detection*

5. Analisi Statistica

In questo paragrafo si analizzano i dettagli degli algoritmi relativi allo schema di *watermarking* proposto. Innanzitutto si dimostrerà che un'immagine marchiata, non manipolata, sarà riconosciuta in maniera univoca con alta probabilità. In particolare ciò significa che l'agoritmo non dà adito ad errori né di tipo *false-positive* né di tipo *negative-positive*.

Successivamente verrà considerato il modello in cui un avversario tenta di cancellare il marchio. La particolare natura dell'algoritmo fa sì che quest'ipotetico avversario non abbia a disposizione abbastanza informazioni per sferrare un attacco

effettivo in grado cioè di cancellare del tutto il marchio. L'unica attacco ragionevole che quest'ultimo può mettere in atto è quello di cercare di spostare i *pixel* dell'immagine marchiata in maniera casuale all'interno della propria sfera di colore. Ma in questo caso, come vedremo, tale strategia non riesce comunque a raggiungere l'obiettivo desiderato in quanto l'algoritmo riesce lo stesso ad identificare univocamente il marchio di provenienza. L'analisi che segue è stata condotta, per semplicità in uno spazio bi-dimensionale anziché in uno spazio a tre dimensioni. Assumiamo cioè che un colore è un punto del piano che sarà opportunamente spostato sulla circonferenza centrata su di esso. I piani $p[\ k\ ]$ diventano retta ortogonali al particolare raggio scelto passanti per il rispettivo punto medio. Sebbene questa scelta porti a dei risultati comunque parziali riguardo ai *bounds* trovati, si riesce comunque a mostrare l'intuizione geometrica alla base dell'analisi teorica. Nel paragrafo 5.3. mostreremo comunque come migliorare i *bounds* nel caso tridimensionale.

5.1 Identificazione di Immagini non Manipolate

Assumiamo che l'immagine marchiata $\boldsymbol{I'}$ data in pasto all'algoritmo di *detection* sia l'esatto risultato dell'applicazione dell'algoritmo di inserimento applicato all'immagine originale $\boldsymbol{I}$. Stiamo supponendo cioè che nessun'altra manipolazione sia stata applicata all'immagine dopo l'inserimento del marchio.

Sia $MARK_i$ il marchio corretto che ha generato $\boldsymbol{I'}$ da $\boldsymbol{I}$ ed indichiamo con $MARK_j$ al variare di j uno qualunque dei restanti marchi (non corretti).

Applicando l'algoritmo di *detection* su $\boldsymbol{I'}$ il contatore C_i raggiungerà di certo il valore $C_i = N$. Di conseguenza sarà possibile avere come output $id = j$ solamente se anche $C_j = N$ per qualche j.

Confrontando $\boldsymbol{I'}$ con $MARK_j$, il contatore relativo C_j sarà incrementato, relativamente al colore k, se e solo se (x'_k, y'_k, z'_k) in $\boldsymbol{I'}$ giace sul lato opposto del piano/retta $\boldsymbol{p}_j[\, k\,]$. Questo accade con probabilità 1/3, dato che tale retta interseca la circonferenza in modo tale che solo 1/3 soddisfa tale condizione (Si veda Fig 4. caso 1). Al variare di $k = 1, 2, \ldots, N$, avendo a che fare con eventi indipendenti, possiamo concludere che $Prob[C_j = N\,] = 3^{-N}$.

L'algoritmo fallisce se esiste un indice $j \neq i$ tale che gli eventi di cui sopra si verifichino. Quindi dato che esisteranno circa M-1 marchi incorretti da esaminare, la probabilità totale di "fallire", cioè di non identificare il marchio corretto è limitata da:

$$Prob[\ \textbf{Mark-Detect}\ fails\ ] \leq (M - 1)\, 3^{-N}$$

5.2 Identificazione di Immagini Manipolate

Supponiamo adesso che un avversario abbia manipolato un'immagine marchiata nel tentativo di rimuovere il *watermark* in essa presente.

Data un'immagine marchiata l'avversario non conosce le direzioni in cui i colori sono stati mossi, in quanto questa informazione fa ovviamente parte della chiave segreta utilizzata per generare il marchio stesso. Quindi l'unica cosa che l'avversario può fare e muovere a caso i colori dell'immagine, nel tentativo di annullare od eseguire una sorta di *undo*, anche parziale, sull'immagine marchiata. Supponiamo per adesso che l'avversario muova tutti i *pixel* aventi lo stesso colore, cioè quelli in

COL_k, all'interno della propria sfera di colore (non può far altro visto che altrimenti deteriorerebbe l'immagine). Si noti inoltre che l'avversario, non può conoscere né vedere la sfera di colore originale ma quella centrata in (x'_k, y'_k, z'_k). Abbiamo già osservato come la probabilità di rimuovere il marchio sia pari a 1/3; dobbiamo inoltre assicurarci che l'avversario muovendo questi colori non causi un incremento del contatore associato ad un marchio non corretto. Come vedremo questo può verificarsi con una sufficientemente piccola probabilità per ciascun colore.

L'analisi probabilistica che segue fa uso della seguente diseguaglianza dovuta Hoefding (Goldreich - 1997).

Date N variabili casuali, indipendenti e uniformemente distribuite (*independent identically distributed* - i.i.d.) $Z_1, Z_2, \ldots, Z_N$, ciascuna a valori nell'intervallo [a , b], e detto m il loro valore atteso. Allora:

$$\Pr\left[\left|\frac{\sum_{i=1}^{N} Z_i}{N} - m\right| \geq d\right] \leq 2e^{-\frac{2d^2N}{b-a}} \qquad (3)$$

Di nuovo indichiamo con $MARK_i$ il marchio corretto e con $MARK_j$ un qualsiasi altro marchio non corretto. L'analisi statistica in questo caso considera i contatori C_i e C_j utilizzati dall'algoritmo di *detection* come delle variabili casuali. L'algoritmo fallisce quando $C_j \geq C_i$, il che è lo stesso che dire che $S = C_j - C_i \geq 0$. Dimostriamo ora che S può essere visto come la somma di N variabili casuali i.i.d. il cui valore atteso è un numero negativo m. Fatto ciò sarà possibile applicare la (3) con $d = -m$ per avere un opportuno *bound* sulla probabilità che $S \geq 0$.

Consideriamo la variabile casuale X_k definita come segue: $X_k = 1$ se analizzando il colore k nell'algoritmo di *detection* incrementiamo il contatore C_i, altrimenti $X_k = 0$. Ovviamente $C_i = \sum_k X_k$. In maniera analoga definiamo una variabile casuale Y_k come segue: $Y_k = 1$ se analizzando il colore k nell'algoritmo di *detection* incrementiamo il contatore C_j, altrimenti $Y_k = 0$. Si ha chiaramente che $C_j = \sum_k X_k$. Quindi $S = \sum_k Z_k$ dove $Z_k = Y_k - X_k$. Fatto ciò ci rimane solo da stimare le distribuzioni di X_k e di Y_k. Per quanto già detto a proposito delle immagini non manipolate si può senza dubbio affermare che Prob[$X_k = 1$] = 2/3, in quanto un movimento *random* nello spazio dei colori, rimuoverà il marchio con probabilità 1/3.

Leggermente più complicata è l'analisi per la stima della distribuzione della variabile casuale Y_k. La variabile Y_k assumerà valore 1 solamente nel caso in cui l'avversario è riuscito a spostare il colore marchiato in punto dello spazio che trovasi sul lato opposto rispetto alla retta $p_j[\,k\,]$. Riferendoci alla Fig. 4, si può suddividere l'originale circonferenza in quattro aree:

1. Assumiamo che la retta $p_j[\,k\,]$ sia ortogonale al raggio contenuto dentro l'angolo a. Ciò accade con probabilità 1/3 dato che $a = 2\pi/3$. In questo caso i 2/3 dei punti della circonferenza/colore del punto "marchiato" stanno sul lato opposto rispetto a $p_j[\,k\,]$. Quindi la probabilità che l'avversario muova il punto verso il lato opposto di $p_j[\,k\,]$ è in questo caso al più 2/9;
2. Assumiamo che la retta $p_j[\,k\,]$ sia ortogonale al raggio contenuto dentro l'angolo b. Questo accade con probabilità 1/6 dato che $b = p/3$. In questo caso al più 1/2 dei punti della circonferenza/colore del punto "marchiato" stanno sul lato opposto rispetto a $p_j[\,k\,]$. Quindi la probabilità che

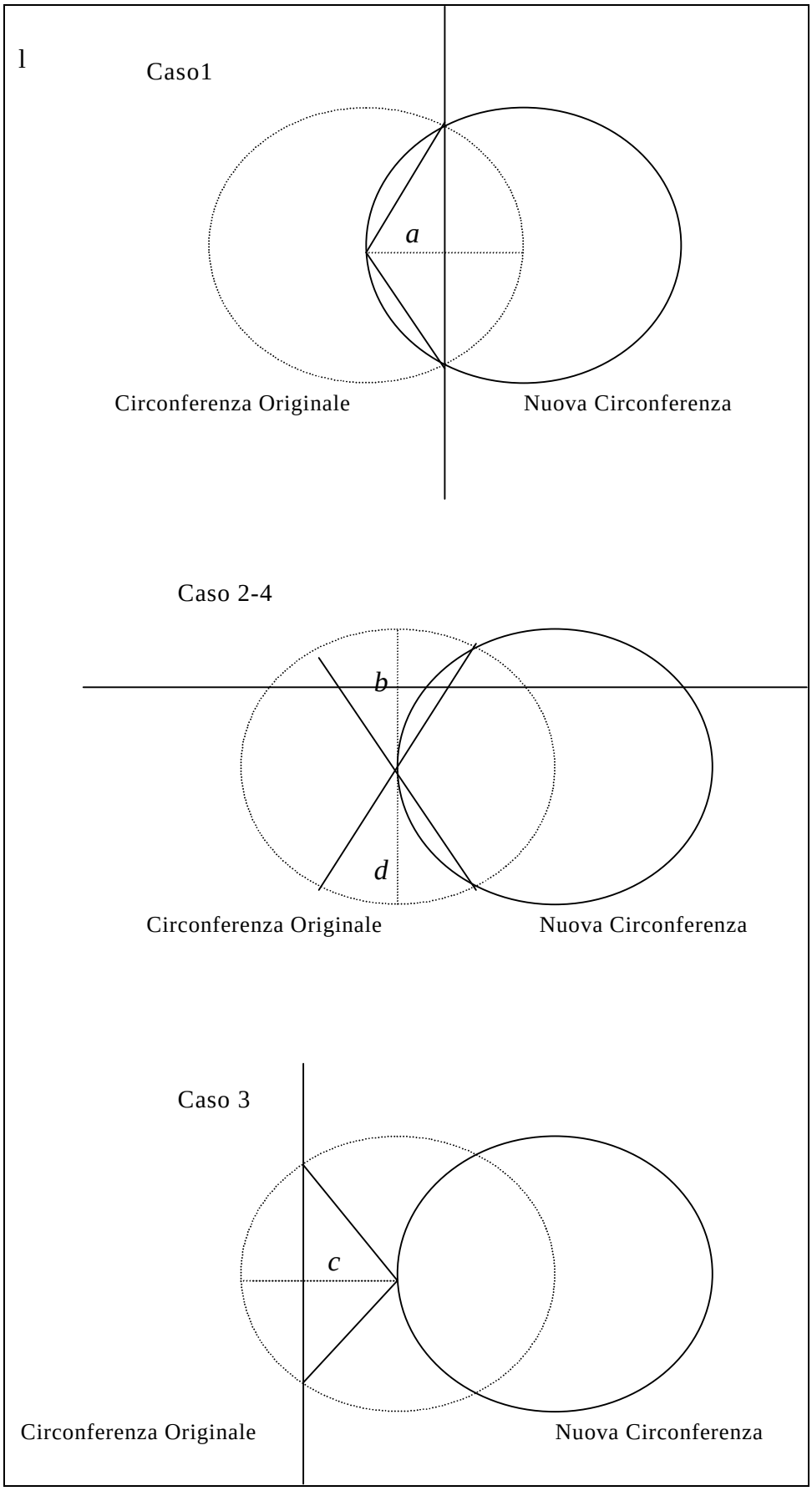
l
Caso1
a
Circonferenza Originale
Nuova Circonferenza
Caso 2-4
b
d
Circonferenza Originale
Nuova Circonferenza
Caso 3
c
Circonferenza Originale
Nuova Circonferenza

Fig.4

l'avversario muova il punto verso il lato opposto di $p_j[\,k\,]$ è in questo caso al più 1/12;

3. Simmetricamente la stessa probabilità di 1/12 è ottenuta quando la retta $p_j[k]$ è ortogonale ad un raggio contenuta nell'angolo d;
4. Assumiamo che la retta $p_j[\,k\,]$ sia ortogonale al raggio contenuto dentro l'angolo c. Questo accade con probabilità 1/3 dato che $c=2p/3$. Ma in questo caso la retta $p_j[\,k\,]$ non interseca la circonferenza cosicché l'avversario non potrà mai muovere il punto sul lato opposto. Ne segue che la probabilità in questo caso è 0.

I casi di cui sopra sono mutuamente esclusivi e coprono tutte le possibilità. Possiamo quindi concludere che Prob[$Y_k = 1$] ≤ 2/9 + 2/12 = 7/18. Nel seguito supporremo un ipotesi più forte per l'avversario assumendo che valga l'uguaglianza.

Possiamo allora stimare la distribuzione di Z_k. Dato che X_k *e* Y_k sono variabili casuali indipendenti si ha che:

$$Z_k = \begin{cases} -1 & \text{con Prob. } 11/27 \\ 0 & \text{con Prob. } 25/54 \\ 1 & \text{con Prob. } 7/54 \end{cases}$$

Ne segue che $m = -5/18$. Applicando l'Equazione (3) con b = 1, $a = -1$ e $d = -m$ si ha che:

$$\text{Prob}[S \geq 0] \leq e^{-m^2 N}$$

Come nel caso precedente la probabilità di un *failure* si ha quando esiste almeno un marchio incorretto che causa $S \geq 0$. Ne segue che:

$$Prob[\ \textbf{Mark-Detect}\ fails\ ] \leq (M - 1)\, e^{-m^2 N}$$

5.3 Analisi Statistica nello Spazio

L'analisi statistica riportata nei paragrafi precedenti ha volutamente semplificato il modello, operando in uno spazio bidimensionale. In questo modo è stato possibile illustrare l'intuizione di base che sta alla base dell'analisi stessa. Comunque rifacendoci allo spazio tridimensionale dello spazio dei colori è possibile ottenere *bounds* probabilistici migliori. In questo paragrafo ci limiteremo ad accennare come sia possibile ottenere quest'ulteriore *improvement*. Richiamiamo innanzitutto alcuni concetti di geometria di base. Una sfera di raggio R ha raggio $4pR^2$. Considerando un piano perpendicolare ad un raggio avente distanza h dal centro, l'area della calotta sferica sul lato opposto del piano rispetto al centro è $2pR(R-h)$. Dato che nel nostro caso noi scegliamo un piano passante per il punto medio del raggio, l'area della calotta sferica sul lato opposto del piano è pR^2 cioè 1/4 del totale.

Ciò immediatamente generalizza il *bound* sulla probabilità di errore nel caso di immagini non manipolate. In particolare l'*upper bound* sulla probabilità di sbagliare nel riconoscere un marchio su di un immagine non manipolata sarà $(M - 1)\, 4^{-N}$.

Nel caso di immagini manipolate da una avversario, riutilizzando la stessa notazione vista precedentemente, si ha che: $Pr[X_k = 1] = 3/4$ dato che una mossa casuale riesce a rimuovere il marchio solo spostando il punto sulla calotta sferica di cui sopra. Nel caso di marchi non corretti è possibile, generalizzando l'analisi caso per caso fatta nel mondo 2D, concludere che $Pr[Y_k = 1] < 7/16$. Ne segue che $m = E[Z_k] < -5/16$ e la probabilità di sbagliare è ancora limitata da $(M - 1)\, e^{-m^2 N}$. E' possibile osservare come la probabilità legata all'errore cresca esponenzialmente con il numero di colori dell'immagine. Per immagini non manipolate il limite teorico è già sufficientemente forte. Per immagini manipolate i *bounds* iniziano a diventare

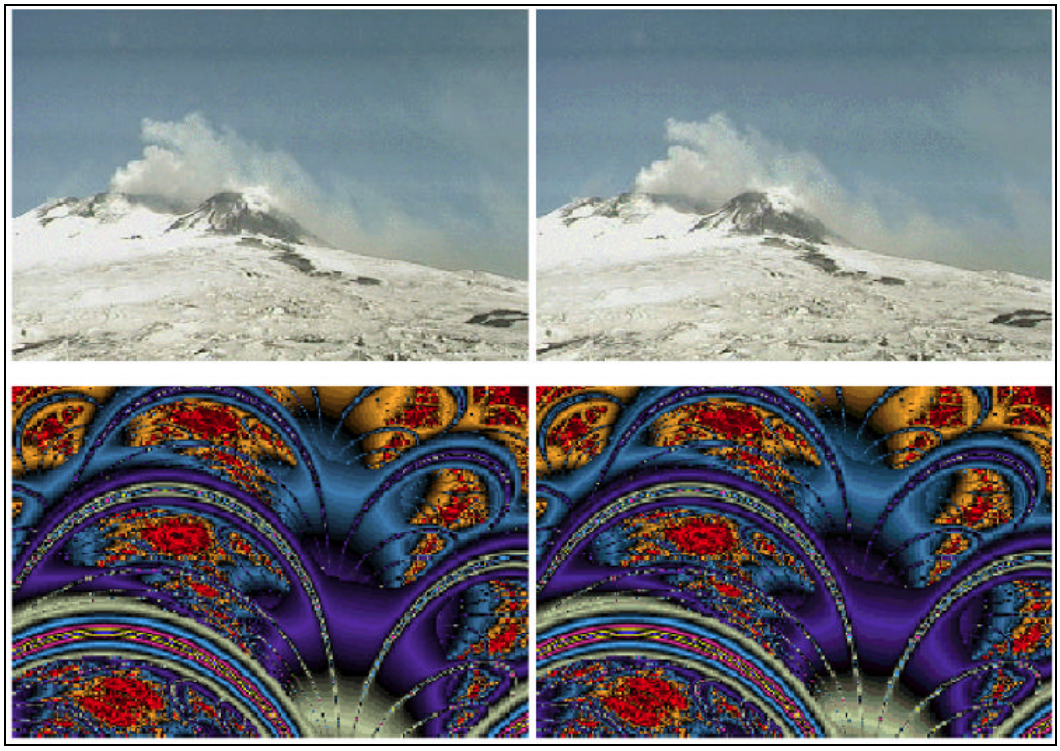

Fig. 5

significativamente robusti con immagini aventi circa un migliaio di colori. Per immagini con *N=256* colori i *bounds* teorici trovati ci danno come probabilità di errore $(M - 1)\, e^{-25}$ che non è chiaramente abbastanza. Ricordiamo comunque che l'analisi è stata generosa, nel senso che i limiti effettivi dovrebbero essere sensibilmente migliori, così come confermato dai dati sperimentali discussi nel successivo paragrafo.

Lo schema nella sua forma di base assume che l'immagine analizzata dall'algoritmo sia stata generata con uno dei marchi nella lista *MARK*. Se questo non dovesse essere il caso (per esempio l'immagine proviene da un altro fornitore) l'algoritmo comunque associa ad essa uno dei marchi della lista e precisamente quello il cui contatore relativo assume valore massimo. Per evitare ciò è sufficiente modificare l'algoritmo cosi che accetti un identificazione come corretta solo se il

Fig. 6

contatore C_{Id} del marchio selezionato sia oltre una data soglia di controllo T. Scegliendo T ≅ N/2 con un'opportuna analisi statistica, condotta sulla falsa riga di quella già vista, si ottengono dei *bounds* per la probabilità di sbagliare dello stesso ordine di grandezza dei precedenti. Tale valore di soglia è anche giustificato dai risultati sperimentali descritti nel paragrafo successivo.

6. Risultati Sperimentali

Abbiamo implementato lo schema di *watermarking* proposto e quindi sia l'algoritmo di inserimento che quello di *detection*, in MATLAB 5.0. Ciò è stato fatto al fine di ottenere una veloce prototipazione. D'altra parte le librerie di *image processing* presenti non hanno consentito di lavorare con immagini con un numero di colori superiori a 256.

Il primo tipo di esperimenti che abbiamo condotto riguarda ovviamente l'impercettibilità del marchio inserito in un'immagine digitale. La Figura 5 mostra la versione *unamarked* e *marked* rispettivamente di una fotografie e di un'immagine

sintetica. Non si notano differenze che possano essere facilmente percepite da un osservatore umano in nessuno dei casi. Ciò accade non soltanto per la naturale perdita di qualità dovuta alla riproduzione cartacea. Piccole differenze cominciano ad intravedersi solo su *monitor* di alta qualità e con degli ingrandimenti appropriati. Un'altra coppia di immagini, rispettivamente originale e marcata, è mostrata in Fig. 6. Il secondo tipo di esperimenti che abbiamo effettuato hanno avuto lo scopo di eseguire il *tuning* dell'algoritmo di *detection*. Chiamiamo *positive* un colore che è stato riconosciuto come marchiato. Per un dato marchio *M,* siamo interessati all'individuazione di un valore di soglia *T* tale che se vengono riconosciuti come *positive* un numero di colori superiore a *T* allora con alta probabilità, l'immagine proviene, magari dopo una qualche manipolazione, da un immagine contenente quel dato marchio. Per far ciò abbiamo utilizzato un insieme di 200 immagini non marcate, diversamente manipolate e un dato marchio *M*; l'algoritmo di *detection* ha dato una media di circa 38% colori *positive* con una varianza $s^2 = 7.5\%$. Abbiamo quindi deciso di utilizzare come valore di soglia un valore $T = (38+2s^2)\%$ come percentuale di colori da riconoscere dato un marchio per considerare l'immagine marchiata. Con tale valore di soglia non è stato osservato nessun falso positivo (cioè un immagine non marcata riconosciuta invece come marcata) nel nostro database di immagini. Il terzo tipo di esperimenti che abbiamo condotto ha fatto uso di una piccola libreria di 15 marchi. Abbiamo dato in pasto all'algoritmo di *detection* un insieme di 100 immagini. Di queste, il 33% erano marchiate con marchi della libreria, il 33% erano non marcate mentre il restante 33% era costituito da immagini marchiate con marchi non presenti nella data libreria. Tutte le immagini non marchiate sono state riconosciute come tali, facendo uso del valore di soglia di cui

sopra. Anche le immagini marchiate, ma con un marchio esterno alla libreria sono state correttamente escluse. Del rimanente gruppo tutte le immagini sono state riconosciute come marchiate e l'unico valore eccedente la data soglia si è registrato solo per il marchio corretto. I risultati sono riportati per completezza nella seguente tabella, dove il numero *n*, indica il numero di colori *positive* trovati analizzando sia immagini esterne alla libreria (denotate con M1) sia immagini effettivamente marchiate (denotate con M2). Le varie *entry* riportano il numero di immagini che hanno riportato, riferito ad *n*, quel dato valore:

	$0 < n \leq T/2$	$T/2 < n < T$	$n \geq T$
Unmarked	25	8	0
Marked M1	20	13	0
Marked M2	0	0	33

Abbiamo ripetuto lo stesso tipo di esperimento utilizzando lo stesso *database* di marchi, ma un differente insieme di 200 immagini, di cui questa volta 100 erano non marchiate e 100 di esse erano state marchiate con uno dei marchi della libreria e successivamente manipolate con una combinazione casuale di 4 dei seguenti operatori: *trimming*/*cropping*, distorsione geometrica, rotazione e *scaling*, equalizzazione, *strechting* del contrasto, filtro mediana e gaussiano (con un *kernel* moderato). Tutte queste operazioni sono state realizzate utilizzando il pacchetto standard *Stirmark 3.0* (Kutter, Petitcolas - 1999; Petitcolas, Anderson, Kuhn - 1998). L'algoritmo di *detection* ha correttamente classificato tutte le immagini con il relativo

marchio, sebbene il valore del *counter* associato si è avvicinato maggiormente al valore di soglia rispetto al caso precedente. Solo in un caso un immagine marchiata ha prodotto un valore minore a quello di soglia, senza essere di conseguenza correttamente identificata. I risultati di tale esperimento sono riassunti nella seguente tabella:

	$n < T$	$T < n < 3/2\ T$	$n > 3/2\ T$	*Correct Mark Identified*
Marked	8	70	22	99
Unmarked	100	0	0	--

Per finire abbiamo simulato alcuni attacchi di tipo *malicious* volti a rimuovere il marchio da un'immagine. Per ottenere ciò abbiamo operato seguendo una delle seguenti possibili alternative:

i) Variare in maniera *random* tutti i colori della *palette* dell'immagine in maniera forte, cioè con un *grosso* spostamento;

ii) Variare in maniera *random* tutti i colori della *palette* dell'immagine ma leggermente, in maniera quasi impercettibile;

iii) Variare in maniera *random* una parte dei colori della *palette* dell'immagine in maniera forte;

iv) Variare in maniera *random* una parte dei colori della *palette* dell'immagine leggermente.

Il marchio diventa difficile da trovare nel caso i), ma in questo caso anche l'immagine risulta di conseguenza visibilmente degradata: nonostante l'immagine sia adesso quasi libera dal marchio la sua utilità è quasi nulla avendo perso la fedeltà

rispetto all'originale. Si veda l'esempio di Fig. 7. Nel caso ii) tale strategia sembrerebbe produrre visivamente un effetto migliore ma in realtà il marchio viene comunque ritrovato facilmente; questo almeno è stato il riscontro che ci hanno fornito gli esperimenti.

Nel caso iii) il marchio è stato comunque riconosciuto in tutti i nostri esperimenti e comunque l'immagine si degrada in maniera evidente. Si veda per esempio la Fig. 8. Infine la strategia iv) provoca solamente un abbassamento del numero di colori *positive* riconosciuti.

Gli esperimenti sono poi proseguiti operando sulle cosiddette immagini *true color*, con 16 Milioni di colori, ed hanno mostrato la robustezza rispetto ad operazioni legate alla riduzione del numero dei colori e allo stesso *requantization attack* del pacchetto *StirMark*. In particolare si è osservato come l'algoritmo sia in grado di riconoscere, in maniera inversamente proporzionale al numero di colori effettivamente presenti nell'immagine *watermarked*, il relativo marchio. Anche passando da un'immagine *true color*, con circa 40.000 colori ad una con 256 il *counter* associato rimane comunque nettamente sopra la soglia *T*. Con le immagini *true color* è possibile ottenere significativi *improvement* legati alla robustezza statistica che come visto dipende proprio dal numero di colori dell'immagine.

Non si è ancora testato, in questa fase, lo schema in presenza di procedure di ri-compressione dell'immagine. Crediamo comunque che anche in questi casi si possano raggiungere ottimi risultati. Tale fiducia è basata sul fatto che la componente luminanza di un immagine, operando nello spazio della *Color Opponency*, analogamente ai più noti algoritmi di compressione, è uno dei fattori chiave anche nel nostro algoritmo.

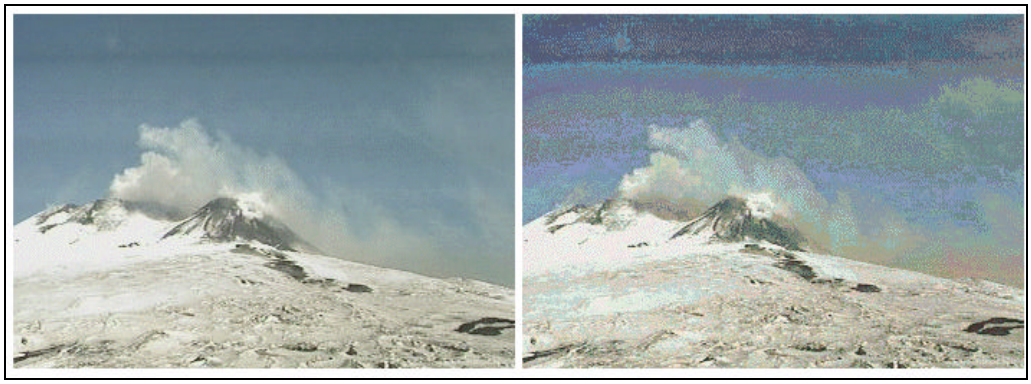

Fig. 7

Fig. 8

7. Conclusioni e Sviluppi Futuri

Si è proposto uno schema di *watermarking* che opera sui colori di un'immagine nello spazio della *Color Opponency*. In accordo con alcune semplici regole geometriche, in tale spazio, sono stati individuati ed implementati i relativi algoritmi di inserimento e di *detection* del *watermark*. Lo schema risulta essere computazionalmente efficiente. Un'analisi teorico-statistica ne ha rilevato la robustezza rispetto ai più ovvi attacchi intenzionali e non. Tale robustezza è risultata dipendere dal numero di colori dell'immagine. Diversi esperimenti preliminari ne hanno mostrato l'effettiva robustezza rispetto alle più comuni operazioni di *image processing*.

Gli sviluppi futuri di tale ricerca prevedono un ulteriore fase sperimentale volta a valutare le prestazioni su immagini con un numero di colori decisamente superiore, per esempio le cosiddette immagini *true color* a 16 milioni di colori. Si pensa altresì di confrontarne punti di forza e debolezze, operando su diversi spazi di colore anche a costo di un maggiore sforzo computazionale.

Si prevede inoltre di valutare la robustezza dello schema ad attacchi di tipo *collusion*, dove cioè si ipotizza che l'avversario abbia a disposizione diverse immagini marchiate con marchi differenti, e possa quindi, basandosi sul confronto diretto di più immagini, riuscire a ricavarne qualche informazione aggiuntiva da sfruttare poi per cancellare o quantomeno degradare significativamente il marchio. A questo proposito si osservi come almeno nella sua forma base l'algoritmo proposto, mantenendo costante la lunghezza del raggio delle sfere di colore per tutte le immagini da marcare, sono sufficienti 4 immagini marchiate per riuscire ad eliminare completamente il marchio (4 punti nello spazio individuano univocamente la rispettiva sfera di appartenenza e di conseguenza il suo centro) ed ottenere una versione *unmarked* dell'immagine stessa. Ne segue che al fine di riuscire ad ottenere un significativo livello di robustezza rispetto a questo tipo di attacco bisogna randomizzare opportunamente la lunghezza del raggio, all'interno di un opportuno *range* di valori.