UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Tesi di laurea in Ingegneria Informatica

Analisi e sviluppo di metodi per l'individuazione di manipolazioni copy-move in applicazioni di image forensics

Candidato
Luca Del Tongo

Relatori
Prof. Alberto Del Bimbo

Prof. Alessandro Piva

Correlatori
Dr. Irene Amerini

Dr. Lamberto Ballan

Dr. Roberto Caldelli

Dr. Giuseppe Serra

Anno Accademico 2011-2012

Vivi come se dovessi morire domani. Impara come se dovessi vivere per sempre.

Mahatma Gandhi

Alla mia famiglia e a Serena

Indice

Elenco delle figure

Elenco delle tabelle

Sommario

Il continuo incremento della diffusione dei sistemi informatici nel corso del ventesimo secolo ha portato inevitabilmente all'incremento dei crimini legati all'utilizzo di strumenti tecnologici; in risposta a questo fenomeno le metodologie investigative si sono avvalse di strumenti informatici legati all'uso del calcolatore. Si è cosi verificato un cambiamento nella modalità di rilevazione, gestione, raccolta delle informazioni arrivando alla definizione di metodologie investigative legate all'uso del computer all'interno di un settore specifico di indagine, il "Digital Forensics". L'obiettivo di questa nuova scienza è quello di analizzare la veridicità di documenti digitali, affinché il loro contenuto possa essere utilizzato come prova giuridica all'interno delle attività svolte durante un'indagine forense. Sebbene le prove digitali su cui indagare possano essere di varia natura, indubbiamente le immagini rivestono un ruolo fondamentale non soltanto in ambito forense ma anche nelle discipline scientifiche, nell'editoria, nella medicina; per poter usufruire appieno di informazioni digitali è quindi necessario verificarne l'autenticità.

Lo scopo di questo lavoro di tesi è quello di proporre una nuova tecnica che sia in grado di stabilire se un'immagine sia autentica o meno a fronte di una eventuale manipolazione di tipo Copy-Move, confrontando la soluzione proposta con lo stato dell'arte presente in letteratura. Nel **primo capitolo** verrà descritto il contesto generale in cui si inserisce questo lavoro di tesi, illustrando con maggior ettaglio sia il percorso che ha portato alla nascita del Multimedia Forensic che gli scenari applicativi ad essa collegati. Nel **secondo capitolo** saranno presentate le più importanti tecniche di individuazione di manipolazioni copy-move (*copy-move detection*) illustrando gli scenari applicativi in cui possono essere utilizzate. Nel **terzo capitolo** verrà illustrato

l'approccio di identificazione di manipolazioni di tipo copy-move progettato e sviluppato all'interno di questo lavoro di tesi. **Nel quarto capitolo** vengono mostrati gli esperimenti con cui si è validato l'approccio proposto in riferimento all'insieme di immagini utilizzate. Nel **capitolo conclusivo** vengono analizzati possibili sviluppi futuri del metodo proposto in questo lavoro di tesi.

Firenze, 20 Giugno 2011

Capitolo 1

Introduzione

Il capitolo descrive il contesto generale in cui si inserisce questo lavoro di tesi, illustrando sia il percorso che ha portato alla nascita del Multimedia Forensic che gli scenari applicativi ad essa collegati.

1.1 Motivazioni

A seguito della "rivoluzione digitale", fenomeno che negli ultimi venti anni ha radicalmente modificato lo stile di vita di ogni singolo individuo, ci troviamo in un mondo ricco di contenuti multimediali, in cui mass-media, scienze e vita quotidiana fanno uso di una comunicazione non verbale da cui è ormai impossibile prescindere. Questo cambiamento ancora in divenire ha creato un rapporto diverso tra contenuto ed utente a livello sia della fruizione che della produzione: coloro che un tempo erano i destinatari, oggi sono diventati gli artefici di un certo tipo di comunicazione visiva. Ad esempio, mai prima d'ora sono state quotidianamente prodotte e messe in circolazione così grandi quantità di immagini fotografiche: accessori quali una webcam, una fotocamera di un dispositivo mobile o una macchina fotografica digitale sono diventati onnipresenti nella nostra vita. Tali dispositivi richiedono competenze tecniche alla portata di tutti dunque una vasta fascia di utenti è in grado di utilizzarli per documentare momenti di vita privata o pubblica. Elemento chiave di diffusione è Internet, dove fotografie e video vengono

caricati su piattaforme digitali come blog, social network e siti personali che ormai hanno raggiunto pari rilevanza dei media e dei canali di informazione tradizionali. Parimenti al proliferare di dati digitali abbiamo assistito ad un rapido sviluppo tecnologico che ha permesso con facilità all'utente di modificare ed alterare contenuti senza lasciar evidenti tracce di manipolazione. Per poter usufruire di informazioni digitali in ambito medico, forense, giornalistico ed in generale nella nostra realtà quotidiana è quindi necessario verificarne l'autenticità.

Molteplici sono i casi di manomissioni di immagini saliti alla ribalta delle cronache, in particolar modo negli ultimi anni. Storicamente i primi esempi di contraffazione risalgono alla seconda metà dell'800: in Figura 1.1 viene mostrato il ritratto risalente al 1860 del presidente americano Abraham Lincoln dato da una composizione della testa del presidente e del corpo del politico sudista John Calhoun. Occorre considerare come questa immagine venga universalmente riconosciuta come il primo esempio di contraffazione fotografica.

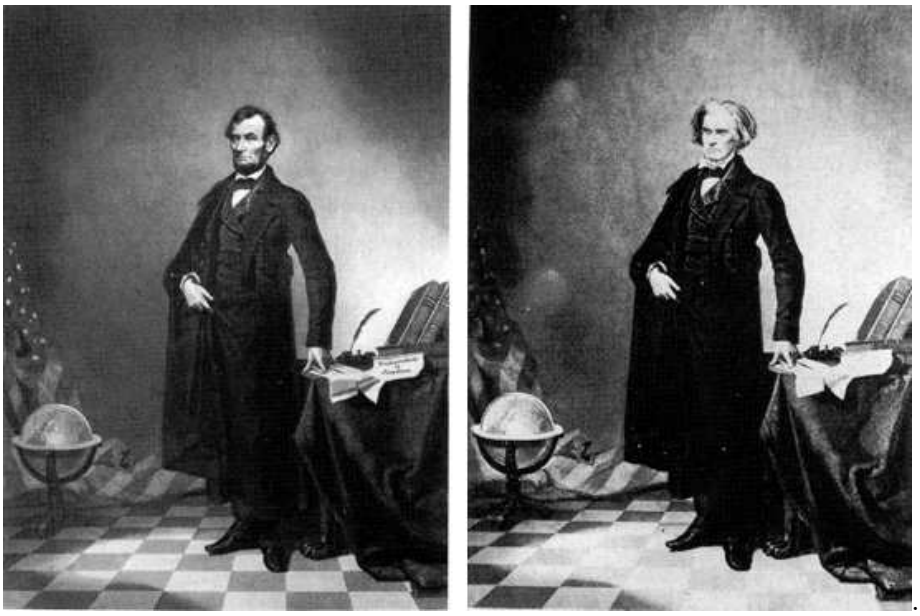


Figura 1.1: A sinistra il ritratto di Lincoln del 1860, a destra il politico Calhoun.

In tempi più recenti, in particolare nel giugno del 1994 venne pubblicata una fotografia (Figura 1.2) manipolata di OJ Simpson sulla copertina della rivista Time Magazine, subito dopo il suo arresto per omicidio. L'immagine originale comparve pochi giorni dopo sulla copertina della rivista Newsweek. La rivista Time venne accusata di aver manipolato la fotografia al fine di rendere la figura di Simpson più scura e minacciosa.

Figura 1.2: A sinistra il ritratto di OJ Simpson pubblicato sul Time, a destra quello pubblicato su Newsweek.

Venendo ai giorni nostri, la televisione pakistana Express News ha pubblicato recentemente una foto del presunto cadavere di Osama Bin Laden (Figura 1.3), ottenuta come composizione di due fotografie: una di Bin Laden in vita, l'altra di una persona deceduta.

Figura 1.3: Composizione della foto del cadavedere di Bin Laden.

A fronte di un incremento costante di manipolazioni delle informazioni multimediali la comunità scientifica ha sviluppato un insieme di tecniche atte

a verificare l'integrità e la fonte delle informazioni digitali dando vita ad una nuova scienza: il Digital Forensics.

L'obiettivo di questa nuova scienza è quello di analizzare la veridicità di documenti digitali, affinché il loro contenuto possa essere utilizzato come prova giuridica all'interno delle attività svolte durante un'indagine forense.

Le maggiori difficoltà che si trovano nell'analisi forense di contenuti multimediali sono legate a:

- rappresentazione astratta delle informazioni.
- eterogeneità dei dispositivi utilizzati per creare e memorizzare informazioni digitali.
- elevata quantità di dati.
- disponibilità di strumenti in grado di mascherare o cancellare le prove digitali.

Esistono diverse specializzazioni legate al Digital Forensics che si differenziano per il contesto operativo in cui operano, in particolare: *Computer Forensics*, *Network Forensics*, *Mobile Forensics* e ***Multimedia Forensics***(vedi Figura 1.4). Nel prossimo paragrafo verranno introdotte le principali tecniche per quanto rigarda l'Image Forensics.

1.2 Principali Tecniche di Image Forensics

Una delle prime tecniche proposte in ambito di Multimedia Forensics, risale al 1993 anno in cui viene proposta l'idea di *fotocamera affidabile* [11] come strumento comprovante l'autenticità di una immagine digitale. Questa tecnica adottava un tipo di approccio "attivo": una fotocamera affidabile, incorporando all'atto della creazione del dato un'informazione detta "watermark", avrebbe reso possibile l'identificazione di eventuali manomissioni dell'immagine analizzando l'integrità del watermark inserito. La realizzazione pratica di quest'idea avrebbe richiesto, da una parte l'adesione dei produttori di macchine fotografiche ad un protocollo standard, dall'altra la riduzione

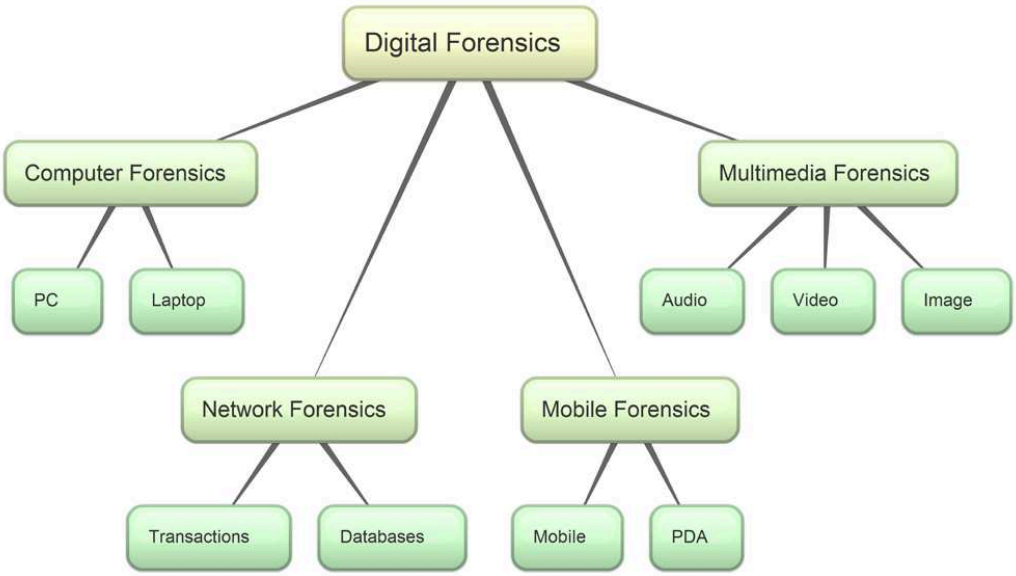


Figura 1.4: Struttura delle specializzazioni che compongono la Digital Forensics.

della qualità delle immagini dovuta all'inserimento di un dato aggiuntivo. Le tecniche attuali di Multimedia Forensics hanno adottato un approccio detto "passivo" in cui nessun tipo di informazione aggiuntiva viene inserita all'interno del dato digitale.

Consideriamo adesso i principali scenari applicativi in cui agiscono le tecniche di Multimedia Forensics, riferendoci in particolare all'Image Forensic, oggetto di studio di questo lavoro di tesi. Si identificano due scenari particolarmente rilevanti, sia dal punto di vista scientifico che forense:

1. identificare la sorgente di acquisizione di contenuti multimediali (Source Identification).

2. rilevare la presenza di manipolazioni di contenuti multimediali (Tampering Detection).

Source Identification

Nel ricostruire la storia di un'immagine, notevole importanza riveste l'identificazione del dispositivo di acquisizione: in una corte di giustizia, l'origine di una particolare immagine può rappresentare una prova fondamentale dell'intero processo; per essere ritenuta attendibile, non devono esistere dubbi sul fatto che l'immagine sia stata catturata o meno da un certo tipo di dispositivo.

Indizi utili sul dispositivo utilizzato possono essere semplicemente ricavati dai metadati (EXIF) presenti nell'intestazione dei files creati dalle fotocamere digitali: è possibile infatti risalire a dettagli relativi alle tabelle di quantizzazione, al tipo di fotocamera, alla risoluzione, ai settaggi di messa a fuoco ed altre caratteristiche [7]. Sebbene i metadati forniscano informazioni significative, questi stessi possono essere facilmente modificati o rimossi rendendoli di fatto inutilizzabili in ambito forense. Le tecniche di Source Identification di tipo passivo utilizzano le informazioni che i principali elementi del processo di acquisizione memorizzano all'interno di un'immagine; in particolare gli elementi più caratteristici sono:

- il Color Filter Array (CFA),
- le imperfezioni nel sensore (PRNU),
- l'aberrazione delle lenti.

Color filter array

I sensori utilizzati all'interno delle macchine fotografiche digitali, siano essi di tipo CCD che CMOS, sono di per sé monocromatici, riescono cioè a rendere il bianco, il nero e la scala di grigi. Sulla superficie del sensore viene quindi collocato un filtro a mosaico detto Color Filter Array (CFA) con il compito di separare e distribuire le tre componenti cromatiche sui diversi pixel del sensore, facendo registrare ad ognuno il segnale relativo ad una sola componente RGB presente nell'immagine; i valori delle due componenenti RGB restanti vengono stimati per ogni pixel, attraverso un processo di interpolazione conosciuto come "demosaicing". La non universalità del processo

di interpolazione viene così sfruttata per l'identificazione del dispositivo di acquisizione: vengono ricercati il pattern CFA e l'algoritmo di interpolazione dei colori del dispositivo digitale che ha acquisito l'immagine. Queste osservazioni sono state sfruttate nel lavoro proposto in [2], in cui, a partire dalla constatazione che la maggior parte delle fotocamere commerciali adotta un CFA di tipo RGB con periodicità 2×2 viene effettuata una stima lineare dei coefficienti di interpolazione dei colori utilizzando regioni dell'immagine a diversa tessitura (liscia, gradiente orizzontale e gradiente verticale). Una delle principali limitazioni di questa tecnica consiste nel fatto che solitamente un produttore utilizza lo stesso tipo di CFA sui vari modelli di macchine fotografiche prodotte, rendendo i diversi modelli dello stesso produttore praticamente indistinguibili. Un'altra limitazione da considerare è che questa tecnica presuppone che il CFA pattern sia di tipo RGB e quindi questo non permette di riconoscere ad esempio le fotocamere che adottano i SuperCCD o che addirittura non hanno il CFA come ad esempio le fotocamere che adottano sensori Foveon X3, che permettono di acquisire direttamente tutte e tre i colori.

Imperfezioni del sensore

Quando un sensore acquisisce una scena, anche nelle migliori condizioni di illuminazione, l'immagine digitale mostrerà comunque piccole variazioni di intensità tra i singoli pixel, a causa delle numerose fonti di rumore che intervengono nel processo di formazione dell'immagine. Analizziamo brevemente le due componenti principali del rumore di acquisizione.

- *fixed pattern noise* (FPN): è un rumore additivo causato da correnti ("*dark current*") che si spostano dal substrato del sensore verso i singoli pixel. E' il responsabile della differenza pixel-to-pixel quando il sensore non è esposto alla luce. La dark current è funzione della dimensione del pixel, della densità di drogaggio del silicio e dei materiali estranei intrappolati nel sensore durante la fabbricazione. Si tratta di un rumore additivo e molte macchine digitali lo sopprimono automaticamente sottraendo un fotogramma nero da ogni immagine scattata.

- *photo-response non-uniformity* (PRNU): è determinato principalmente dalla differenza di sensibilità alla luce dei pixel dello stesso sensore (*pixel non-uniformity*, PNU), la quale è dovuta alla disomogeneità dei wafer di silicio con i quali sono realizzati i sensori. In secondo luogo è influenzato dalle differenze di dimensione, risposta spettrale, spessore del rivestimento e altre imperfezioni che hanno origine durante il processo di fabbricazione. Insieme al FPN è la componente principale di rumore e non risulta affetto da temperatura ed umidità dell'ambiente esterno.

Il rumore generato da ogni sensore è discriminativo nell'identificazione di una fotocamera. In particolare, ogni singola fotocamera digitale possiede idealmente un sensore identico a tutte quelle dello stesso modello, in realtà il disturbo generato nelle immagini è legato sia al sensore, che alle minuzie nella costruzione e nell'assemblaggio di ogni dispositivo. Questo assicura una differenza tra singoli dispositivi sufficiente a rendere improbabile la presenza di due camere che generino il medesimo disturbo. Nel lavoro di Fridrich [6] è proposto l'utilizzo del PRNU come *fingerprint* univoco di una fotocamera stimando il PRNU da un'immagine e calcolando successivamente la correlazione con un set di PRNU di riferimento.

Aberrazione delle lenti

La lente ha lo scopo di presentare al sensore ottico un'immagine quanto più fedele possibile alla scena che si vuole ritrarre. Tuttavia la formazione delle immagini da parte delle lenti non è perfetta, ma si verificano delle distorsioni dette "aberrazioni ottiche", che sono di vario tipo. Le due distorsioni principalmente analizzate, anche nell'identificazione della sorgente di acquisizione dell'immagine, sono: la distorsione radiale della lente descritta in [16] e l'aberrazione cromatica descritta in [19]. Ogni produttore di lenti adotta una propria tecnica di compensazione della distorsione radiale, applicandola solitamente all'intera gamma di lenti prodotte; conseguentemente, lenti di macchine fotografiche di produttori diversi lasciano una traccia unica sulla foto catturata.

Il secondo tipo di aberrazione esaminato per risolvere il problema dell'identificazione della sorgente di acquisizione è l'aberrazione cromatica, ovvero il fenomeno dove le differenti lunghezze d'onda della luce non convergono nella stessa posizione sul piano focale. Esistono due tipi di aberrazioni cromatiche: quelle longitudinali e quelle laterali. In entrambi i casi le aberrazioni cromatiche provocano varie imperfezioni sotto forma di colori nell'immagine. Tipicamente, l'aberrazione cromatica si manifesta come un alone attorno all'oggetto osservato, rosso da una parte e blu dall'altra. Questo perché rosso e blu sono ai due estremi dello spettro della luce visibile, e sono quindi i colori per i quali la differenza di rifrazione è maggiore. Solo le aberrazioni cromatiche laterali sono prese in considerazione nel metodo descritto in [19] per l'identificazione della sorgente di acquisizione. Questo disallineamento delle componenti RGB è stimato attraverso dei parametri di distorsione massimizzando le mutue informazione tra i vari canali. Infine questi parametri sono usati in [19] per identificare la sorgente dei cellulari attraverso l'uso di un classificatore SVM.

Tampering detection

Esistono al giorno d'oggi diversi strumenti di image editing utilizzabili in maniera semplice e intuitiva (es. Adobe Photoshop e Gimp). Con tali programmi, è consentita una grande quantità di operazioni: oggetti possono essere cancellati dalla scena, particolari possono essere clonati nella foto, oggetti di computer grafica possono essere aggiunti a una scena reale e così via. Queste tecniche stanno diventando sempre più sofisticate tanto che le alterazioni sono diventate quasi impercettibili. Quindi stabilire l'autenticità di una foto è un punto chiave per permettere di usare immagini digitali come prove cruciali. Le varie tecniche di manipolazioni di immagine possono essere classificate in tre macro-categorie: fotoritocco, composizione (*Image Splicing)* e duplicazioni di Regioni *(Copy-Move)*.

Fotoritocco

Il fotoritocco viene considerato la tipologia meno dannosa di manipolazione dell'immagine digitale in quanto non ha come obiettivo quello di modifica-

re il contenuto semantico di un'immagine. Questa tecnica mette in risalto talune peculiarità andando ad esempio a modificare l'istogramma dei colori come nell'immagine di OJ Simpson in Figura1.2), oppure applicando effetti di sfocatura; viene principalmente utilizzata all'interno dell'editoria per "migliorare" determinate caratteristiche di un'immagine in modo tale da renderla più attraente nei confronti del lettore. L'identificazione di questa tipologia di manipolazione è stata affrontata più volte in letteratura, nel lavoro di Stamm e Ray Liu [31] ad esempio viene illustrato un metodo automatico di rilevazione di aumento di contrasto mentre in [17] viene mostrato un approccio in grado di rilevare manipolazioni di tipo motion blur.

Image Splicing

Questa tipologia di manipolazione consiste nel combinare due o più immagini per creare un'immagine falsa, come quella del cadavere di Bin Laden in Figura 1.3. E' una tecnica maggiormente aggressiva rispetto al fotoritocco che mira a modificare la percezione che l'utente ha dell'immagine originale. I programmi di image editing moderni rendono agevole combinare più immagini (ad esempio attraverso l'utilizzo dei livelli), ottenendo risultati che difficilmente sono rilevabili dall'occhio umano. Dopo aver combinato le immagini è inoltre possibile utilizzare avanzate tecniche di sfumatura per mascherare i confini delle regioni interessate e per dare all'immagine un aspetto più uniforme. E' opportuno sottolineare come le immagini utilizzate nella composizione finale non debbano essere necessariamente provenienti da immagini naturali: è possibile infatti utilizzare immagini create attraverso algoritmi di Computer Grafica. L'identificazione di questa tipologia di manipolazione è stata affrontata più volte in letteratura ad esempio in [32] molte caratteristiche dell'immagine particolarmente sensibili a manipolazioni di tipo splicing, vengono estratte ed utilizzate per addestrare un classificatore.

Copy-Move

Modificare un'immagine attraverso una manipolazione di tipo Copy-Move consiste nell'aggiungere o rimuovere del contenuto da un'immagine digitale, sfruttando del contenuto presente nell'immagine stessa; in altre parole una parte dell'immagine viene "copiata ed incollata" su sé stessa.

Semanticamente le manipolazioni di tipo Image Splicing e Copy-Move so-

no equivalenti in quanto hanno entrambe come obiettivo quello di aggiungere o rimuovere del contenuto informativo, differiscono operativamente poiché la prima opera per composizione di molteplici immagini mentre la seconda opera per duplicazione di contenuto. Quando viene eseguita una manipolazione di tipo Copy-Move, le porzioni di immagine duplicate condividono ancora la maggior parte delle proprietà intrinseche (es. rumore o tavolozza dei colori) con il resto dell'immagine, rendendo particolarmente difficile individuare e localizzare le eventuali manipolazioni. Un esempio di manipolazione di immagine di tipo Copy-Move salito alla ribalta delle recenti cronache vede coinvolti i media iraniani che nel 2008 hanno manipolato le immagini di un test missilistico non completamente riuscito, al fine di mostrare il corretto lancio di missili di corto raggio(vedi Figura 1.5). Le principali tecniche di identificazione di manipolazioni di tipo copy-move, verranno illustrato nel prossimo capitolo.

Figura 1.5: Esempio di manipolazione di tipo Copy-Move.

1.3 Obiettivi

Lo scopo di questo lavoro di tesi è quello di proporre una nuova tecnica che sia in grado di stabilire se un'immagine sia autentica o meno a fronte di una eventuale manipolazione di tipo Copy-Move, confrontando la soluzione proposta con lo stato dell'arte presente in letteratura.

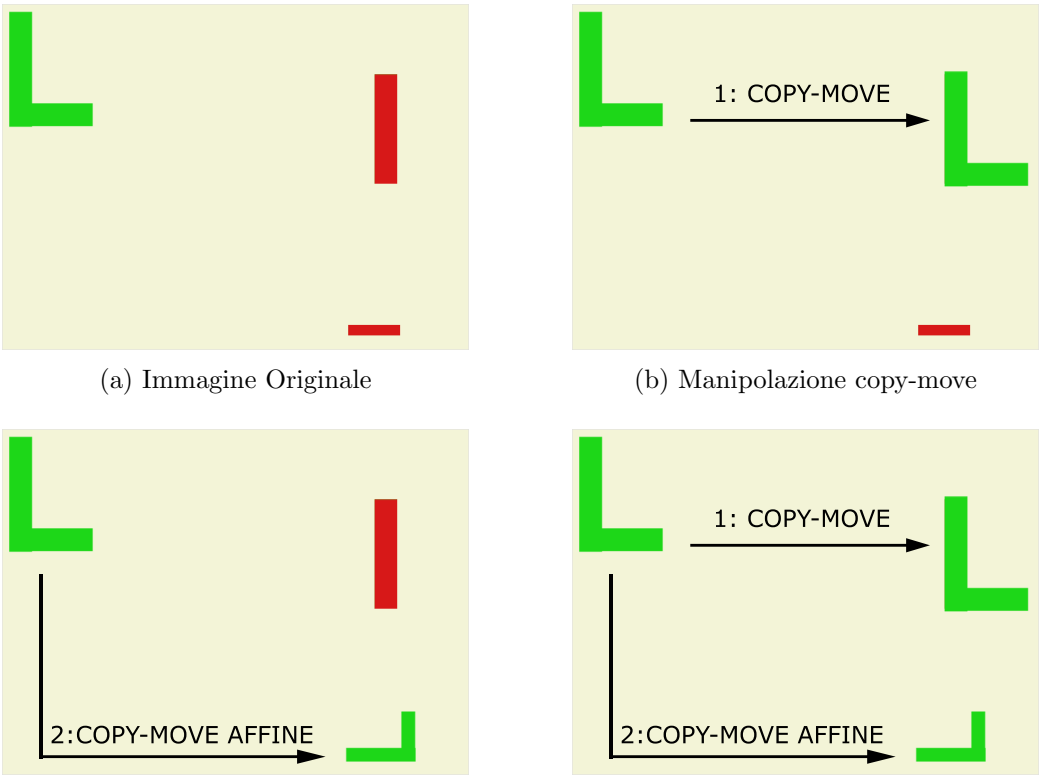
Capitolo 2

Tecniche di Individuazione di manipolazioni Copy-Move

In questo capitolo saranno presentate le più importanti tecniche di individuazione di manipolazioni copy-move (copy-move detection) presenti in letteratura analizzandone robustezza e relativa complessità computazionale;

La pratica maggiormente utilizzata nel manipolare un'immagine, al fine di aggiungervi od eliminarne del contenuto, consiste nel "copiare ed incollare" intere regioni appartenenti ad essa. Quando questa operazione viene effettuata con meticolosità, può essere veramente arduo identificare visualmente la zona alterata (vedi Figura 1.5 Capitolo 1). Definire uno strumento di identificazione di manipolazioni Copy-Move (*Copy-Move Detection*) risulta parimenti difficile per le scarse tracce lasciate durante la manipolazione; la duplicazione del contenuto infatti viene fatta sfruttando regioni provenienti dalla medesima immagine, rendendo inutilizzabili approcci basati sulle caratteristiche statistiche quale rumore [22] o compressione [21]. E' importante evidenziare come in scenari reali, un utente malintenzionato che voglia manipolare un'immagine non si limiti semplicemente ad una mera duplicazione di contenuto, frequentemente per rendere la manipolazione più realistica applica operazioni di scalatura e rotazione alla regione duplicata per poi "incollarla" nella destinazione scelta (vedi Figura 2.1). Dopo aver "incollato" la

regione duplicata, l'utente può ulteriormente perfezionare la manipolazione applicando sfocatura, rumore e comprimendo l'immagine.



(a) Immagine Originale

(b) Manipolazione copy-move

(c) Manipolazione copy-move affine

(d) Manipolazione copy-move multipla

Figura 2.1: esempi di manipolazione di tipo Copy-Move di un immagine dimostrativa: la regione di colore verde viene duplicata al fine di nascondere quelle di colore rosso.

Le manipolazioni di tipo copy-move che sfruttano traslazioni, scale e rotazioni possono essere modellate mediante trasformazioni affini. Una trasformazione affine di una regione viene espressa come composizione di una trasformazione lineare avente matrice associata $\mathbf{A}_{2\times2}$ e di una traslazione, esprimibile attraverso il vettore $\tilde{\mathbf{t}}_{2\times1}$. Indicando con $\mathbf{x}_i = (x, y)^\intercal$ i punti di una regione R_1 dell'immagine e con $\mathbf{x}'_i = (x', y')^\intercal$ i punti appartenenti ad una

regione R_2, copia di R_1 sottoposta a trasformazione affine, possiamo scrivere:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} + \tilde{\mathbf{t}}. \tag{2.1}$$

Data un'immagine digitale, una prima strategia di identificazione di manipolazioni copy-move, consiste nell'esaminare in modo esaustivo ogni possibile coppia di regioni; supponendo di lavorare con una immagine di dimensioni M×N un approccio a forza bruta presenta una complessità pari a $O(MN)^2$ risultando di fatto inapplicabile su immagini reali. Ad esclusione dell'approccio operante per forza bruta, i numerosi metodi di copy-move detection proposti in letteratura condividono una serie di passi base, illustrati in Figura 2.2, differenziandosi poi per l'implementazione data.

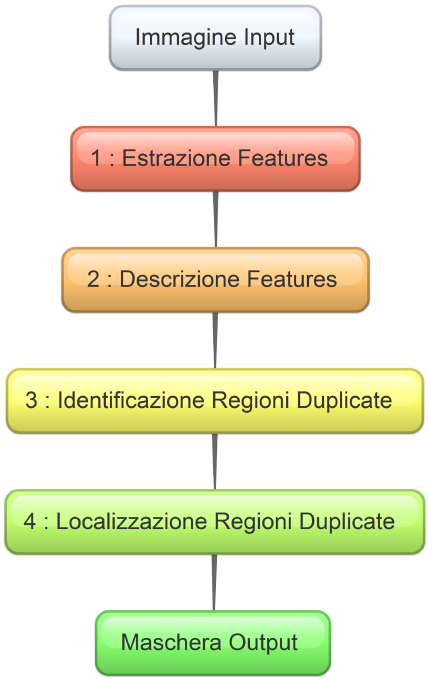


Figura 2.2: Pipeline di un generico algoritmo di copy-move detection.

Nelle prossime sezioni vengono illustrati alcuni tra i più significativi lavori presenti in letteratura. In particolare nella sezione 2.1 vengono mostrare le tecniche invarianti a manipolazioni Copy-Move in cui la regione duplicata viene semplicemente traslata, mentre nella successiva sezione 2.2 verranno mo-

strati i lavori più recenti in grado di identificare regioni duplicate sottoposte a trasformazioni affini.

2.1 Tecniche di Identificazione Copy-Move base

2.1.1 Metodo basato sulla trasformata discreta del coseno

J. Fridrich, David Soukal e Jan Lukas [14] hanno proposto una delle prime tecniche di individuazione di manipolazioni Copy-Move sfruttando le **caratteristiche della trasformata discreta** (DCT).

Analizziamo in dettaglio le principali fasi del metodo in riferimento alla Figura 2.2 indicando con **I** l'immagine in input contenente $N \times N$ pixel.

Durante la fase di estrazione delle features, una finestra rettangolare di dimensioni $b \times b$, viene fatta scorrere di una posizione lungo l'immagine a partire dall'angolo superiore destro sino a raggiungere quello inferiore sinistro. L'immagine **I** viene così suddivisa in un insieme di blocchi rettangolari sovrapposti di cardinalità pari a:

$$N_b = (N - b + 1)^2. \tag{2.2}$$

La scelta della dimensione di un blocco determina implicitamente un limite inferiore pari a b^2 sulla minima dimensione rilevabile di una regione duplicata.

Nella fase di descrizione delle features viene creata una matrice S di dimensioni pari a $N_b \times b^2$. Per ogni blocco $i \in N_b$, vengono poi calcolati i coefficienti della **trasformata discreta del coseno** (DCT); i coefficiente calcolati, vengono quantizzati attraverso una tabella di quantizzazione **Q** ed infine memorizzati all'interno della i-esima riga della matrice S.

La fase di identificazione delle regioni duplicate, consiste nell'ordinare in modo lessicografico le righe della matrice S; questa operazione permuta le righe in modo tale che blocchi simili, corrispondenti a regioni duplicate, si trovino in posizioni contigue.

Al fine di localizzare le regioni clonate, gli autori suggeriscono di considerare una regione come duplicata qualora essa presenti un certo numero T di blocchi identici con stesso vettore di spostamento (*vettore di shift*). Indicando con $s1 = (i1, j1)$ ed $s2 = (i2, j2)$ le coordinate spaziali di due blocchi $S1$ ed $S2$, con stessi coefficienti DCT quantizzati, il vettore di shift s risultante sarà pari a:

$$s = (s1, s2) = (i1 - i2, j1 - j2). \tag{2.3}$$

Al termine dell'esecuzione l'algoritmo indicherà come regioni duplicate solo quelle che presentano un vettore di shift con un contatore maggiore di T.

La robustezza complessiva dell'intero metodo dipende dai coefficienti di quantizzazione Q e dalla soglia T legata al vettore di shift. La complessità del lavoro analizzato, determinata principalmente dalla fase di ordinamento della matrice S risulta pari a $O(N^2 \log_2 N^2)$.

2.1.2 Metodo basato sull'analisi delle componenti principali

Il lavoro di Farid e Popescu [29] si basa sull'utilizzo di una tecnica di riduzione dimensionale conosciuta in letteratura come **analisi delle componenti principali** (PCA [9]). L'approccio proposto utilizza la stessa strategia di suddivisione dell'immagine in blocchi sovrapposti illustrata nella sezione 2.1.1, differenziandosi dal metodo di Fridrich [14] solamente nella fase di descrizione delle features: in riferimento alla Figura 2.2, l'analisi del metodo proposto si concentrerà principalmente su questa fase (blocco numero 3).

Indicando con $\vec{x_i} = 1 \dots N_b$ l'insieme degli N_b blocchi estratti dall'immagine (equazione 2.2), la fase di descrizione delle features prevede come primo passo il calcolo della matrice di covarianza C:

$$C = \sum_{i=1}^{N_b} \vec{x_i} \vec{x_i}^{\mathrm{T}} \tag{2.4}$$

Gli autovettori $\vec{e_j}$ della matrice C, insieme ai corrispondenti autovalori $\vec{\lambda_j}$ che soddisfano la seguente equazione:

$$C\vec{e_j} = \vec{\lambda_j}\vec{e_j} \tag{2.5}$$

definiscono le *componenti principali*. Gli autovettori $\vec{e_j}$ ottenuti formano una nuova base lineare con cui rappresentare ciascun blocco $\vec{x_i}$:

$$\vec{x_i} = \sum_{j=1}^{b} a_j \vec{e_j} \tag{2.6}$$

La dimensionalità di questa rappresentazione viene semplicemente ridotta andando a troncare la serie dell'equazione 2.6 ai primi N_t termini, con $t < b$. Dopo aver memorizzato la rappresentazione PCA di ogni blocco all'interno di una matrice S, quest'ultima viene ordinata in modo lessicografico. La fase di localizzazione delle regioni duplicate consiste nell'analisi della frequenza dei vettori di spostamento: solo le regioni i cui vettori di shift superano una certa soglia T, in analogia al lavoro di Fridrich illustrato nella sezione 2.1.1, vengono considerate duplicate.

La complessità dell'algoritmo, dominata dalla fase di ordinamento risulta pari a $O(N_t N \log N)$. L'utilizzo di una rappresentazione di tipo PCA conferisce al metodo proposto maggiore robustezza rispetto ai metodi basati sulla DCT in presenza di rumore di tipo gaussiano e di variazione dei fattori di compressione.

2.1.3 Metodo basato sui momenti invarianti a sfocatura

Mahdian e Saic recentemente hanno proposto un approccio [25] basato sull'utilizzo di momenti invarianti ad operazioni di sfocatura (*blur moment invariants* [13]): gli autori evidenziano come una delle tecniche maggiormente utilizzate per eseguire manipolazioni di immagini realistiche, consista nel creare una transizione graduale tra le regioni duplicate e quelle originali; questa operazione solitamente viene eseguita dall'utente malintenzionato con l'ausilio dei molteplici strumenti di sfocatura (blur tools) presenti nella maggior parte dei moderni programmi di elaborazione delle immagini. Un momento invariante a sfocatura rappresenta un funzionale B che soddisfa la condizione $B(f) = B(D(f))$, dove con D è stato indicato l'operatore di sfocatura.

In riferimento alla Figura 2.2, il primo passo consiste nel suddividere l'immagine originale in blocchi sovrapposti così come descritto nella sezione

2.1.1. Per comprendere la successiva fase di descrizione delle features verrà brevemente richiamata la teoria dei momenti centrali utilizzata nel calcolo dei momenti invarianti.

Sia $f(x, y)$ la rappresentazione in termini di intensità di una immagine bidimensionale **I**. Il momento m_{pq} di ordine $(p + q)$ è definito come:

$$m_{pq} = \iint x^p y^q f(x, y)\, dx\, dy \tag{2.7}$$

Sfruttando il teorema di unicità di Papoulis [1], valido nel caso in cui $f(x, y)$ sia continua a tratti e presenti valori non nulli solo in una porzione finita del piano, si dimostra come la sequenza dei momenti m_{pq} di qualsiasi ordine sia unicamente determinata da $f(x, y)$. Questo risultato permette di descrivere una immagine in funzione dei suoi momenti di ordine più basso. Per rendere i momenti invarianti a traslazione, utilizzabili cioè in tecniche di copy-move detection, è necessario calcolarli in funzione del valor medio (momenti centrali). Il momento centrale μ_{pq} di ordine $(p + q)$ viene definito come:

$$\mu_{pq} = \iint (x - \overline{x})^p (y - \overline{y})^q f(x, y)\, dx\, dy \tag{2.8}$$

in cui $\overline{x}$ e $\overline{y}$ rappresentano il valor medio.

A partire dai momenti centrali descritti nell'equazione 2.8 gli autori mostrano come sia possibile derivare una serie di momenti invarianti a sfocatura; l'algoritmo proposto dagli autori ne utilizza 24, rappresentando ogni blocco attraverso un vettore delle features $B_i = \{B1, B2, \ldots, B24\}$. Successivamente viene applicata la tecnica PCA ad ogni vettore delle features B_i ottenendo un nuovo vettore B'_i di dimensione pari a 3.

La successiva fase di identificazione delle regioni duplicate, memorizza ogni features B'_i all'interno di una struttura dati di tipo kd-tree al fine di velocizzare la ricerca delle regioni/blocchi simili. Il grado di similarità calcolato tra due vettori delle features B'_i e B'_j viene espressa attraverso un indice s utilizzando la seguente formula:

$$s(B'_i, B'_j) = \frac{1}{1 + \rho(B'_i, B'_j)}, \tag{2.9}$$

dove ρ rappresenta la distanza euclidea tra le componenti dei vettori:

$$\rho(B'_i, B'_j) = \left(\sum_{k=1}^{dim} \left(B'_i[k] - B'_j[k] \right)^2 \right)^{\frac{1}{2}}. \tag{2.10}$$

Per ciascun blocco, rappresentato dal vettore delle features B'_i, vengono ricercati i blocchi con grado di similarità maggiore di una soglia T. Nella fase finale di questo approccio vengono indicate come duplicate solamente quelle regioni che siano simili tra di loro ed abbiano un intorno composto da regioni simili a loro volta.

2.2 Tecniche di Identificazione Copy-Move affine

2.2.1 Metodo basato su blocchi circolari

Molti dei metodi presenti in letteratura, tra cui [14], [29] e [25] esposti nella Sezione 2.1 di questo capitolo, utilizzano blocchi rettangolari sovrapposti per rappresentare le informazioni contenute all'interno di un'immagine. L'utilizzo di blocchi rettangolari impedisce a questi metodi di rilevare manipolazioni di tipo Copy-Move nel caso in cui la regione copiata ed incollata venga inoltre ruotata. A partire da queste osservazioni J. Wang, G. Liu, H. Li, e Z. Wang hanno proposto un approccio [15] che supera le suddette limitazioni utilizzando blocchi circolari (Figura 2.3 lato sinistro).

In riferimento alla Figura 2.2, la fase di estrazione delle features comincia con il calcolo della piramide gaussiana dell'immagine: indicando con G l'immagine di partenza, ovvero il livello zero della piramide gaussiana, il generico livello l viene ottenuto da un'operazione di convoluzione del livello $l-1$ con un filtro passa basso gaussiano $w(m,n)$ seguita da un sottocampionamento:

$$G_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) G_{l-1}(2i+m, 2j+n) \tag{2.11}$$

Gli autori nel lavoro proposto utilizzano il primo livello l_1 della piramide gaussiana; questo permette di ottenere sia una riduzione delle dimensioni

dell'immagine pari ad $\frac{1}{4}$ che una rappresentazione delle informazioni in bassa frequenza. Dopo aver calcolato il livello l_1 della piramide gaussiana, questo viene suddiviso in $N_{bc} = \frac{1}{4}(N - 2r + 1)^2$ blocchi circolari di raggio r (Figura 2.3 lato sinistro).

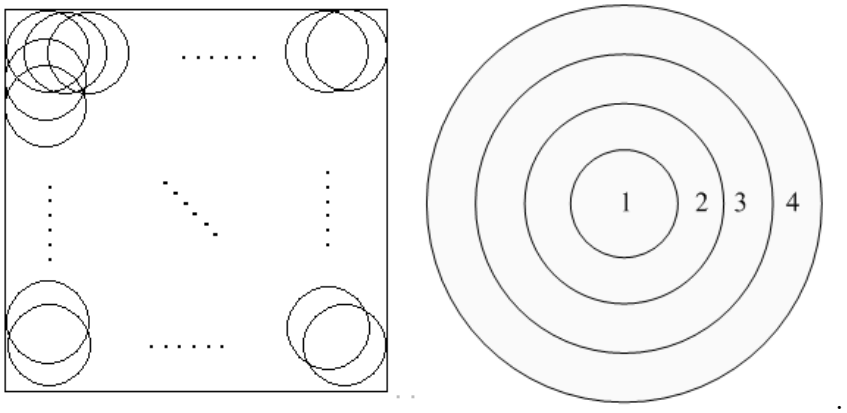


Figura 2.3: Sul lato sinistro vengono mostrati i blocchi circolari sovrapposti, sulla parte destra i cerchi concentrici.

Nella fase di descrizione delle features ogni blocco circolare B_i viene suddiviso in quattro cerchi concentrici Ω_1, Ω_2, Ω_3 ed Ω_4 (Figura 2.3 lato destro). Successivamente viene creata una matrice S di dimensioni pari a $N_{bc} \times 4$ in cui per ogni blocco B_i viene memorizzato l'insieme dei quattro valori medi $\phi_k = \frac{\sum x_{i,j}}{|\Omega_i|}, x_{i,j} \in \Omega_k$ dei cerchi concentrici. Dopo essere stata ordinata in modo lessicografico la matrice S viene scandita durante la fase finale di localizzazione in cui vengono ricercate le regioni simili calcolando la distanza euclidea tra elementi di S successivi.

Per minimizzare il numero di falsi positivi durante la procedura di localizzazione vengono utilizzate tre soglie:

Ts : soglia di similarità tra coppie di regioni,

Td : soglia sulla distanza minima presente tra coppie di regioni,

Ta : soglia sulla dimensione minima T_a della regione duplicata.

L'utilizzo di uno schema di estrazione dell'informazione di tipo circolare unitamente all'utilizzo di features basate sul valor medio, rende questo meto-

do invariante a manipolazioni copy-move in cui le regioni duplicate abbiano subito rotazioni.

2.2.2 Metodo basato sui momenti di Zernike

Seung-Jin Ryu, Min-Jeong Lee, e Heung-Kyu Lee hanno recentemente proposto un approccio [30] basato sui momenti di Zernike. I momenti di Zernike rappresentano una proiezione dell'immagine su un insieme di basi complesse ortogonali; oltre ad essere in grado di codificare il contenuto di un'immagine, gli autori dimostrano algebricamente come il modulo di tali momenti sia invariante rispetto a rotazione; il metodo proposto sfruttando tali caratteristiche, è in grado di rilevare manipolazioni copy-move anche in presenza di rotazioni.

In sintesi in accordo con i passi illustrati in Figura 2.2, il lavoro degli autori prevede che:

- l'immagine viene suddivisa in blocchi rettangolari sovrapposti,
- vengono calcolati i momenti di Zernike di ciascun blocco,
- la matrice contenente tali momenti viene ordinata,
- le regioni duplicate vengono localizzate scorrendo la matrice e cercando momenti di Zernike simili.

2.2.3 Metodo basato sulle coordinate logaritmiche polari

Solorio e Nandi [4] hanno proposto un approccio che rappresenta il contenuto di un'immagine sfruttando il sistema di coordinate logaritmiche polari (*log-polar*).

In riferimento alla Figura 2.2, il primo passo è comune al lavoro di Fridrich descritto nella sezione 2.1.1 e consiste nella suddivisione dell'immagine in blocchi rettangolari.

Nella successiva fase di descrizione delle features, vengono sommate lungo l'asse angolare le coordinate polari logaritmiche di ciascun pixel appartenente

ad un blocco, creando così un vettore di features monodimensionale invariante a riflessione, scala e rotazione. Illustriamo adesso con maggiore dettaglio questa fase per mostrare le caratteristiche di invarianza sopra descritte.

Sia $(x, y) \in \mathbb{R}^2$ un punto dell'immagine espresso mediante le coordinate logaritmiche polari secondo le seguenti equazioni:

$$\begin{aligned} x &= e^{\rho} \cos\theta \\ y &= e^{\rho} \sin\theta \end{aligned} \tag{2.12}$$

in cui $\rho \in \mathbb{R}$ e $0 \leq \sigma < 2\pi$. Applicando adesso una trasformazione di similarità al punto (x, y) otteniamo un nuovo punto (x', y'):

$$\begin{aligned} x' &= \mu(x \cos\varphi + y \sin\varphi) \\ y' &= \mu(x \sin\varphi - y \cos\varphi) \end{aligned} \tag{2.13}$$

in cui μ e φ rappresentano la scala e l'angolo di rotazione della trasformazione di similarità. Riscrivendo l'equazione 2.13 in coordinate polari logaritmiche si ottiene:

$$\begin{aligned} x' &= e^{(\rho + \log\mu)} \cos(\varphi - \theta) \\ y' &= e^{(\rho + \log\mu)} \sin(\varphi - \theta) \end{aligned} \tag{2.14}$$

Osservando l'equazione 2.14 emerge che, utilizzando una rappresentazione logaritmica polare, una trasformazione geometrica di scala o di rotazione consista semplicemente in una traslazione del sistema di riferimento (vedi Figura 2.4). Per passare alla rappresentazione monodimensionale di ciascun blocco $B_i(\rho, \theta)$ si esegue una somma su θ delle coordinate logaritmiche polari di ciascun pixel di B_i, ottenendo così il descrittore monodimensionale: $\vec{v_i}$:

$$\vec{v_i}(\rho) = \sum_{\theta} B_i(\rho, \theta) \tag{2.15}$$

Per ottenere una feature, maggiormente robusta a fenomeni di compressione dell'immagine o di introduzione di rumore, gli autori suggeriscono di calcolare la trasformata di Fourier (FFT) del descrittore monodimensionale illustrato in equazione 2.15. La fase finale di localizzazione delle regioni duplicate utilizza un coefficiente di correlazione calcolato sulle ampiezze delle trasformate di Fourier dei vettori delle features precedentemente ordinati.

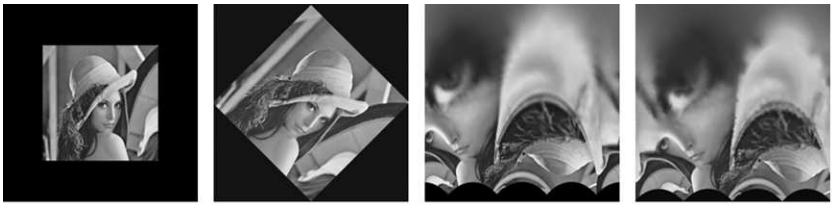


Figura 2.4: (a) Immagine originale *Lena*; (b) immagine ruotata e scalata di (a); (c) conversione log-polar dell'immagine (a); (d) conversione log-polar dell'immagine (b).

2.2.4 Metodo basato sui SIFT

A partire dal lavoro di Fridrich, illustrato nella sezione 2.1 di questo capitolo, tutti gli approcci proposti hanno sempre utilizzato una suddivisione dell'immagine in blocchi sovrapposti come primo passo di rappresentazione dell'informazione: in sostanza un'immagine, in accordo alla teoria atomistica, viene vista rappresentabile dalla somma delle parti che la compongono. Recentemente sono stati proposti metodi di copy-move detection [12], [28], [3], che sfruttano un approccio diverso: nel rappresentare un oggetto od una regione dell'immagine non vengono utilizzati blocchi sovrapposti, bensì un sottoinsieme di punti salienti (*keypoint*) rappresentati da un opportuno descrittore [27]. Nel lavoro di Amerini et al. [3] viene suggerito l'utilizzo del descrittore SIFT [20] per l'invarianza di quest'ultimo a manipolazioni di tipo copy-move affine; in riferimento alla Figura 2.2, nell'approccio proposto la fase di estrazione delle features, comunemente all'algoritmo SIFT, consiste nel:

- individuare gli estremi locali nello scale–space filtrando ripetutamente l'immagine in input con kernel gaussiani di diversa varianza ottenendo keypoints invarianti a scala.

- eliminare i punti con basso contrasto (sensibili al rumore) ed eseguire un'interpolazione di una funzione quadratica per aumentare l'accuratezza di localizzazione dei keypoint.

Dopo aver estratto i keypoints, la fase di descrizione delle features prevede:

- l'assegnazione di una o più orientazioni canoniche ad ogni keypoint; questo passo fornisce invarianza a rotazioni.
- la generazione dei descrittori locali; ogni keypoint viene rappresentato attraverso un vettore a 128 dimensioni basato sulle informazioni locali del gradiente di intensità.

Al termine di queste prime due fasi un'immagine in input I, di dimensione $N \cdot N$, viene rappresentata attraverso un insieme di M features, con $M \ll N$. E' importante sottolineare come il numero di features estratte M sia funzione delle componenti di intensità e tessitura di un'immagine: due differenti immagini delle stesse dimensioni contengono infatti un numero di keypoints solitamente diverso.

Durante la fase di identificazione delle regioni duplicate vengono cercati keypoints con descrittori simili; il criterio di similarità suggerito dagli autori è basato sul calcolo della distanza euclidea tra i descrittori. In dettaglio due vettori delle features vengono considerati duplicati qualora, il rapporto dr tra la distanza d_1 del descrittore più prossimo e la distanza d_2 del secondo più prossimo, risulti inferiore ad una soglia T prefissata:

$$dr = \frac{d_1}{d_2} < T \tag{2.16}$$

Il valore della soglia T utilizzata all'interno dell'approccio viene posto pari a 0.6 in accordo con i risultati sperimentali proposti da Lowe [20]. Per identificare le possibili regioni duplicate, gli autori propongono l'utilizzo di una strategia dl clustering spaziale sull'insieme dei keypoints con descrittori simili; in particolare viene utilizzato un algoritmo di clustering gerarchico di tipo agglomerativo *(hierarchical agglomerative clustering)*. Terminata la fase di clustering l'immagine in input verrà considerata manipolata attraverso operazioni di copy-move affini, qualora siano presenti due o più clusters contenenti ciascuno almeno tre keypoints.

Capitolo 3

Approccio Proposto

In questo capitolo viene presentata la strategia di identificazione di manipolazioni copy-move sviluppata in questo lavoro di tesi.

La progettazione del metodo di identificazione di manipolazioni copy-move, proposto in questo lavoro di tesi, ha incluso una prima fase di raccolta ed analisi dei requisiti. Sono state analizzate le varie modalità con le quali un utente manipola un'immagine mediante un attacco di tipo copy-move al fine di nascondere o duplicare oggetti presenti nell'immagine; l'analisi svolta ha messo in luce le caratteristiche che un qualsiasi algoritmo di copy-move detection "ideale" debba possedere:

1. essere in grado di operare su qualsiasi tipo di immagine di qualsiasi dimensione.

2. rilevare manipolazioni a prescindere dalle dimensioni delle regioni duplicate, dal loro spostamento e dalla loro cardinalità.

3. essere invariante rispetto a:

   (a) trasformazioni di tipo geometrico,

   (b) variazioni di colore e luminosità,

   (c) operazioni di sfocatura,

(d) introduzione di rumore e deterioramento della qualità dell'immagine.

La fase di progettazione e successiva implementazione dell'approccio proposto è stata quindi guidata dalle specifiche sopra elencate opportunamente rilassate per rendere la soluzione realizzabile. Il sistema proposto, illustrato in Figura 3.1, è in grado di stabilire se un'immagine abbia subito manipolazioni di tipo copy-move fornendo in uscita una maschera di localizzazione che evidenzia le eventuali regioni duplicate. Le scelte progettuali e la descrizione delle varie fasi del sistema, verranno illustrate approfonditamente nelle sezioni successive di questo capitolo.

3.1 Estrazione e descrizione delle features

La prima decisione da effettuare, in fase di progettazione di un sistema di copy-move detection, consiste nella scelta della strategia di estrazione e descrizione delle features da utilizzare; in riferimento alle tecniche di identificazione di manipolazioni copy-move presenti in letteratura, alcune delle quali illustrate nel secondo capitolo, possiamo individuare due modalità di rappresentazione di un'immagine:

1. suddividendola in blocchi sovrapposti
2. utilizzando un insieme di punti salienti (*keypoints*)

Nell'approccio proposto si è deciso di rappresentare un'immagine attraverso un insieme di keypoints, estratti e descritti utilizzando l'algoritmo SIFT [20]; la scelta effettuata è basata sull'analisi dei fattori relativi alla complessità computazionale, all'invarianza ed al grado di copertura delle regioni dell'immagine. Analizziamo ad esempio la complessità computazionale tipica di un approccio a blocchi per un'immagine di dimensioni pari a 3648×2736: in accordo all'equazione 2.2, questa viene suddivisa in un numero di blocchi sovrapposti che si aggira sui 10 milioni; ipotizzando che ogni blocco venga descritto attraverso un vettore delle features contenente 72 valori double, la

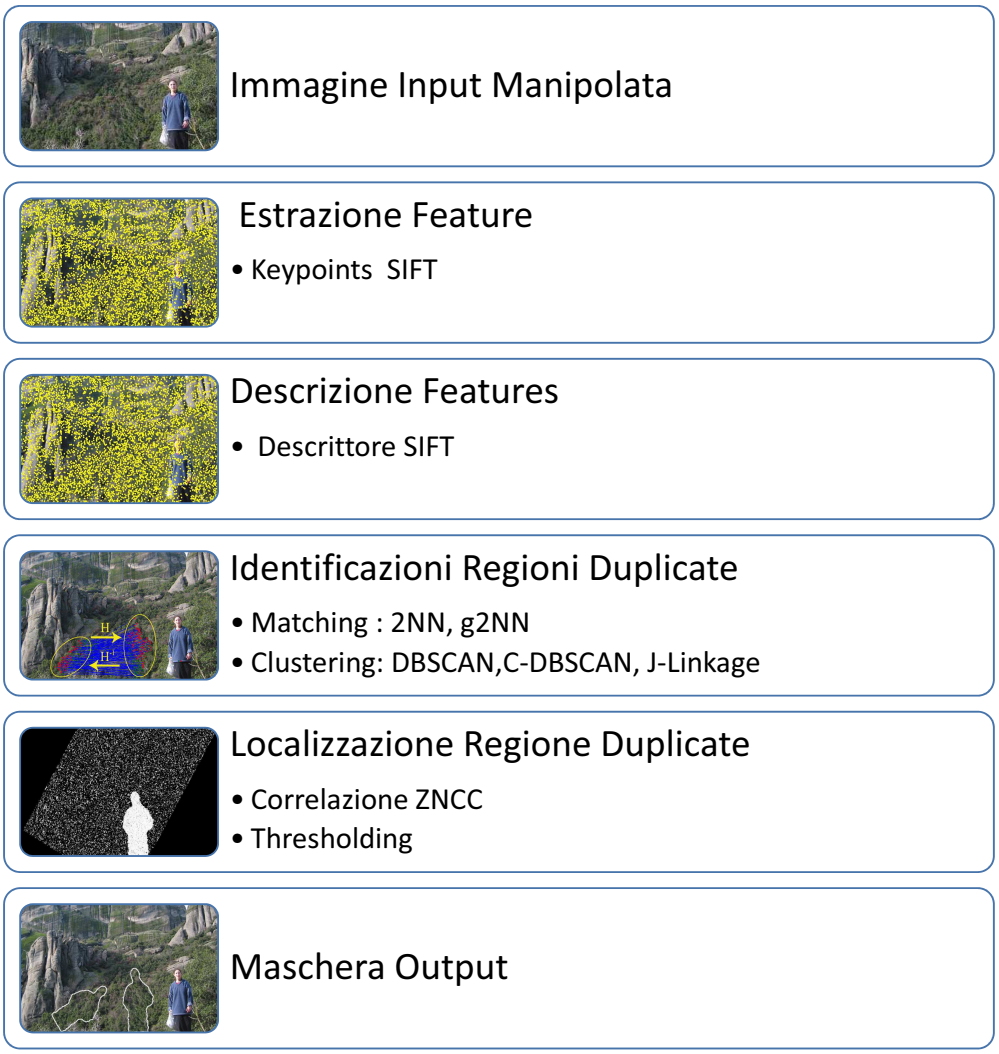


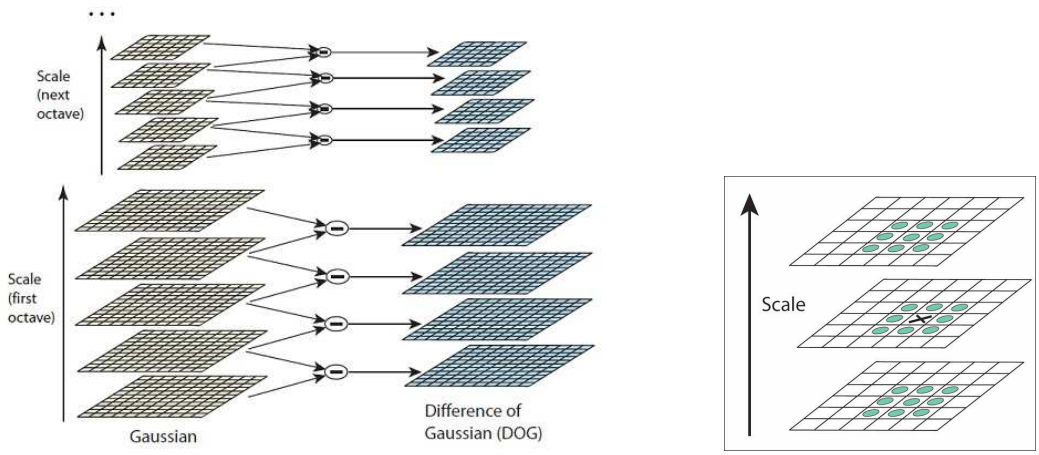
Figura 3.1: Pipeline sviluppata durante il lavoro di tesi.

complessità spaziale di una ipotetica matrice S contenente l'insieme dei vettori delle features è quantificabile in circa 3GB di memoria. In termini di complessità computazionale e di invarianza il lavoro di Mikolajczyk e Schmid [27] mostra come l'utilizzo di SIFT sia indicato in scenari di matching da differenti viste di uno stesso oggetto, simile in alcnu aspetti allo scenario applicativo di questo lavoro di tesi in cui vengono analizzate immagini potenzialmente contenenti regioni duplicate.

In sintesi il processo di individuazione dei punti salienti e dei relativi descrittori può essere riassunto nella seguente successione di passi:

1. **individuazione degli estremi locali nello scale-space**: si effettua filtrando ripetutamente l'immagine originale con kernel gaussiani di diversa varianza, ottenendo due piramidi di immagini (Figura 3.2 (a)): la prima costituita dalle immagini ripetutamente convolute con filtri Gaussiani $G(x; y; \sigma)$, la seconda dalle diverse DoG, $D(x; y; \sigma)$; gli estremi locali saranno quindi ricercati ad ogni livello della piramide di DoG, ottenendo keypoints invarianti a scala.

2. **localizzazione dei keypoints**, cioè dei massimi e dei minimi locali della $D(x; y; \sigma)$: viene effettuata comparando ciascun campione con gli otto adiacenti del livello corrente e con i nove delle due scale immediatamente superiore ed inferiore (Figura 3.2 (b)). Vengono eliminati i punti con basso contrasto (sensibili al rumore) ed eseguita un'interpolazione di una funzione quadratica per aumentare l'accuratezza di localizzazione del keypoint.



(a) Creazione delle piramidi di immagini

(b) Estremi locali DoG

Figura 3.2: (a) Processo di creazione della piramide di immagini convolute con filtri Gaussiane e della piramide di DoG. (b) Gli estremi locali della DoG sono individuati confrontando il pixel (contrassegnato dalla X) con i 26 adiacenti in una regione 3x3 alla scala corrente, ed alle due adiacenti.

3. **assegnazione di una (o più) orientazioni canoniche**: una volta individuate le coordinate e la scala del keypoint, si assegna una orientazione che garantisce invarianza a rotazioni. La scala del keypoint, viene utilizzata per selezionare il corretto livello della piramide di immagini; a questo punto si crea un istogramma delle direzioni del gradiente locale dei campioni contenuti in un intorno del keypoint e per ogni picco dell'istogramma verrà creato un keypoint con tale orientazione.

4. **generazione dei descrittori locali**: per ogni keypoint viene creato un corrispondente descrittore SIFT. Sinteticamente, le orientazioni di un intorno 16×16 del keypoint, vengono accumulate in istogrammi che racchiudono le informazioni contenute in sotto–regioni 4×4 (Figura 3.3). Ognuno di questi 16 istogrammi è formato da 8 bin, corrispondenti ad 8 diverse direzioni, il descrittore SIFT sarà quindi costituito da un array di $16 \times 8 = 128$ elementi.

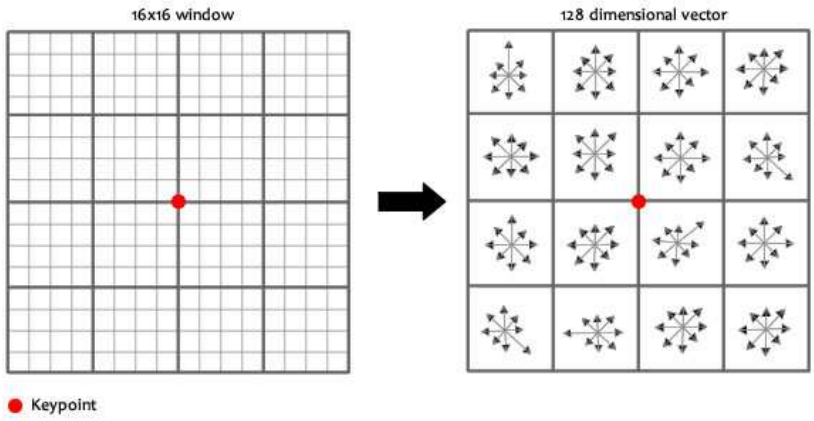


Figura 3.3: Generazione dei descrittori locali. La figura mostra la creazione di un descrittore di dimensioni 128, costruito a partire da un set di 16x16 campioni centrati sul keypoint indicato in rosso.

3.2 Identificazione regioni duplicate

Per identificare eventuali regioni duplicate, all'interno di questo lavoro di tesi, si è ritenuto necessario l'utilizzo di una duplice fase di identificazione:

- in un primo passo vengono ricercati keypoints con descrittori SIFT simili,
- successivamente si esegue un raggruppamento di keypoints "corrispondenti" attraverso una procedura di clustering.

Nella sezione 3.2.1 verrà illustrato in dettaglio la fase di matching dei descrittori SIFT estratti dall'immagine, mentre nella successiva sezione 3.2.2 verranno mostrate le varie strategie di clustering implementate in questo lavoro di tesi.

3.2.1 Ricerca features duplicate

Al termine delle fasi di estrazione e descrizione delle features un'immagine in input I viene rappresentata attraverso un insieme sparso di features $\mathbf{X} = \{\mathbf{x_i}\}_{i=1...n}$ in cui:

$$\mathbf{x}_i = \{x, y, \sigma, o, \mathbf{f}\}. \tag{3.1}$$

Ogni features è composta quindi dalle informazioni relative alle coordinate in pixel (x, y) del punto (**keypoint**), alla scala σ, all'orientazione canonica o ed infine al descrittore locale $\mathbf{f}$. Nel caso in cui un'immagine I venga sottoposta a manipolazioni di tipo copy-move all'interno dell'insieme delle features $\mathbf{X}$ estratto, saranno presenti un numero di descrittori identici provenienti da regioni duplicate durante il processo di manipolazione; possiamo in definitiva sfruttare l'eventuale presenza di descrittori simili come "spia" di una eventuale manipolazione copy-move dell'immagine in input. L'approccio più semplice nel ricercare coppie di descrittori simili, consiste nel fissare una soglia T_0 e selezionare le coppie di descrittori che presentino una distanza, ad esempio quella euclidea, inferiore a T_0; l'alta dimensionalità dello spazio delle features associato ai vari descrittori (128-dimensionali) penalizza l'utilizzo di metodi che fissano una T_0 introducendo corrispondenze poco accurate

(*outliers*); nel lavoro di Lowe [20], viene suggerito l'utilizzo di un criterio di similarità (*2NN Ratio*), basato sul rapporto tra la distanza d_1 del descrittore più prossimo (*1NN*) a quello in esame e la distanza d_2 del secondo più prossimo (*2NN*). Analizziamo in dettaglio il funzionamento del criterio *2NN Ratio* sopra descritto: siano $\mathbf{F} = \{f_1, f_2, \ldots, f_n\}$ l'insieme dei descrittori estratti da un'immagine, vogliamo stabilire se il descrittore $\mathbf{f_k}$ abbia una corrispondenza. Indicando con $\mathbf{D} = \{d_1, d_2, \ldots, d_{n-1}\}$ il vettore ordinato delle distanze euclidee tra f_k e gli altri $n-1$ descrittori ($d_1 \leq d_2 \leq d_3 \leq \cdots \leq d_{n-1}$), il descrittore $\mathbf{f_k}$ viene considerato simile al descrittore a lui più prossimo ($1NN$), qualora il rapporto dr tra d_1 e d_2, risulti inferiore ad una soglia T prefissata:

$$dr = \frac{d_1}{d_2} < T \qquad T \cong 0.6 \tag{3.2}$$

Gli approcci di copy-move detection basati sui SIFT [12], [28], [3] utilizzano il *2NN Distance Ratio* suggerito da Lowe come strategia di identificazione di descrittori simili: due descrittori vengono considerati l'uno la copia dell'altro qualora essi differiscano in modo marcato dal resto dei descrittori estratti.

Nel presente lavoro di tesi si è deciso di definire un nuovo criterio di similarità (**g2NN**) estendendo il *2NN Ratio*; questa decisione è emersa dall'analisi effettuata sul comportamento del criterio di matching *2NN Ratio*, in uno scenario di copy-move "multiplo"; supponiamo che l'utente malintenzionato nel manipolare un'immagine, abbia selezionato una regione R_1 della stessa e questa sia stata duplicata in due regioni distinte R_2 ed R_3. Utilizzando la notazione introdotta precedentemente, possiamo osservare come il vettore D conterrà nelle prime due posizioni, due distanze d_1 e d_2 praticamente identiche ($d_1 \approx d_2$): queste infatti rappresentano la distanza tra il descrittore f_k e le rispettivi copie f_{1nn} ed f_{2nn} appartenenti alle regioni duplicate R_2 ed R_3 ($f_k \approx f_{1nn} \approx f_{2nn}$). Applicando quindi l'equazione 3.2 la coppia di descrittori formato da f_k ed il descrittore a lui più prossimo f_{1nn} non verranno rilevati come duplicati dal criterio di similarità *2NN Ratio* poiché il rapporto $dr = d_1/d_2$ sarà prossimo ad uno.

Il criterio ***g2NN*** proposto in questa lavoro di tesi (procedura 3.4), al fine di valutare la similarità tra descrittori adotta la seguente logica: ricevuto in input il vettore ordinato delle distanze relative all'i-esimo punto, il calcolo

del rapporto tra le distanze d_1/d_2 viene ripetuto finché non supera la soglia T (equazione 3.2); se la procedura termina dopo k iterazioni ogni keypoint corrispondente ad una delle prime d_k distanze viene considerato come duplicato dell'i-esimo punto; prove sperimentali dimostrano empiricamente come questa generalizzazione aumenti l'accuratezza della fase di identificazione dei descrittori duplicati ed al tempo stesso non introduca outliers rispetto al metodo originale *2NN Ratio* (Figura 3.5).

g2NN($Features\ F; T \in N$)
1. $pairwiseDistances = EvaluateDistances(F)$
2. $Sort(pairwiseDistances)$
3. $i = i + 1$
4. $neighbours_i = \emptyset$
5. **while** $i \leq |F|$
   5.1. $j = 2$
   5.2. **while** $pairwiseDistances[i][j]/pairwiseDistances[i][j+1] \leq T$
      5.2.3 $neighbours_i = neighbours_i + j$
      5.2.4 $j = j + 1$
   5.3. $i = i + 1$

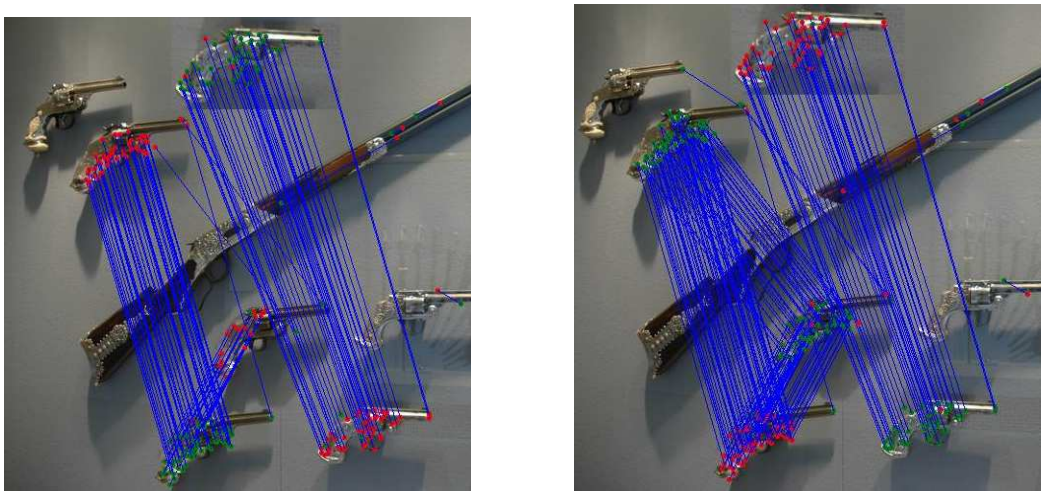
Figura 3.4: Procedura g2NN

3.2.2 Clustering

La fase di ricerca di features duplicate all'interno di un'immagine, illustrata nella sezione precedente di questo capitolo, produce un insieme di features $\mathbf{X'} = \{\mathbf{x_i}\}_{i=1...k}$, con $\mathbf{x}_i = \{x, y, \sigma, o, \mathbf{f}\}$, che godono della seguente proprietà:

$$\forall x'_i \in \mathbf{X'}, \exists j : f_i \approx f_j \tag{3.3}$$

L'insieme delle features $\mathbf{X'}$ è composto da una famiglia di sotto-insiemi mutuamente esclusivi, in cui ciascun sotto-insieme sarà composto da un numero di descrittori simili tra di loro pari al numero di volte che un utente abbia clonato una regione. A causa del rumore introdotto dal criterio di similarità g2NN alcuni sotto-insiemi possono contenere descrittori appartenenti a



(a) Strategia di Identificazione 2NN

(b) Strategia di Identificazione g2NN

Figura 3.5: (a) Strategia di identificazione SIFT duplicati 2NN. (b) Strategia di identificazione SIFT duplicati g2NN. Rispetto ad (a) il numero di corrispondenze individuato risulta incrementato.

regioni che non state realmente duplicate; per eliminare eventuali sottoinsiemi non corretti ed identificare in modo robusto possibili regioni duplicate, si è deciso di utilizzare, come suggerito all'interno del lavoro di Amerini et al. [3], una procedura di clustering in grado di raggruppare keypoints appartenenti alla medesima regione duplicata.

Nelle successive sotto-sezioni vengono illustrati i principali algoritmi di clustering utilizzati in questo lavoro di tesi.

DBSCAN

Tra i vari approcci conosciuti al problema del clustering si è scelto di agire mediante un algoritmo di tipo density-based (basato sulla densità dei punti), in particolare si è utilizzato l'algoritmo DBSCAN (acronimo per Density-Based Spatial Clustering of Applications with Noise) [26]. Rispetto ad altre tipologie di clustering, DBSCAN (pseudocodice 3.6) presenta alcune caratteristiche che ben si adattano allo scenario applicativo dell'approccio proposto, in particolare:

- non necessità di conoscere a priori il numero di cluster da formare,
- esegue una clusterizzazione parziale sui dati robusta alla presenza di elementi outliers,
- è in grado di scoprire cluster di qualsiasi forma spaziale

L'idea di base dietro il concetto di clustering basato sulla densità è quella di costituire un insieme di oggetti appartenenti allo stesso cluster in maniera iterativa, in modo che la distanza minima tra due oggetti appartenenti allo stesso cluster, non sia mai maggiore di un soglia ϵ scelta a priori, e che tale cluster presenti una cardinalità maggiore di un certo numero minimo di punti, indicato con *MinPts*.

DBSCAN($DataPoints\ D; MinPts \in N; \epsilon \in N$)
1. si scelgono a priori valori adeguati per i parametri ϵ e *MinPts*
2. finché esistono punti non etichettati
   3. si sceglie un punto p non etichettato e lo si etichetta come visitato
   4. si determinano i punti p_n contenuti in un intorno centrato in p di raggio pari a ϵ
   5. si inseriscono i punti p_n nell'insieme N dei vicini di p
   6. if $p_n < minPts$
      7. il punto p viene etichettato come rumore
   8. else
      9. crea un nuovo cluster C
      10. aggiungi il punto p a C
      11. per ogni punto p' contenuto in p_n
          12. if p' è un punto non visitato
              13. etichetta p' come visitato
              14. calcola p'_n in un intorno centrato in p' di raggio pari a ϵ
              15. si inseriscono i punti p'_n nell'insieme N' dei vicini di p'
              16. if $p'_n >= minPts$
                  17. $N = N \cup N'$
          18. if p' non appartiene a nessun cluster
              19. aggiungi p' al cluster C

Figura 3.6: Procedura di Clustering DBSCAN

L'efficacia dell'algoritmo DBSCAN è fortemente influenzata dalla coppia di soglie $(\epsilon, MinPts)$ che ne regolano il funzionamento: gli esperimenti degli

autori [26] dimostrano come una buona scelta del valore MinPts, lavorando con punti bidimensionali, sia pari a 4; non forniscono però alcun suggerimento su come selezionare in modo automatico il parametro ϵ che regola il raggio dell'intorno di un punto. Si è deciso quindi di utilizzare un metodo di stima analitico per il parametro ϵ proposto da M. Daszykowski et al [24]; in questo lavoro gli autori evidenziano come la densità dei punti da clusterizzare può essere confrontata con la densità dello stesso numero di punti distribuiti in modo uniforme nel volume determinato dai punti da clusterizzare; analiticamente questo concetto viene espresso mediante la seguente formula:

$$\epsilon = \sqrt{\frac{\mathbf{V} \cdot \mathbf{k} \cdot \mathbf{\Gamma}(\frac{n}{2} + 1)}{\mathbf{m} \cdot \sqrt{\pi^{\mathbf{n}}}}} \tag{3.4}$$

in cui **k** rappresenta il parametro MinPts, **m** ed **n** indicano rispettivamente il numero di punti da clusterizzare e relativa dimensionalità, Γ è la funzione Gamma di Eulero mentre V rappresenta il volume occupato dall'insieme dei punti nello spazio n:

$$V = \prod_{i=1}^{n} \left\{ (max(x_i) - minx_i) \right\} \tag{3.5}$$

L'utilizzo congiunto del valore minPts (4) suggerito dagli autori di DBSCAN in aggiunta alla stima della densità ϵ eseguita sui dati ha permesso di definire una procedura automatica di clustering adattiva priva di parametri; esistono tuttavia una serie di configurazioni spaziali critiche (Figura 3.7), in cui DBSCAN produce risultati non corretti, in particolare tra le cause di malfunzionamento possiamo annoverare:

1. presenza di cluster sovrapposti (Figura 3.7 (a)),
2. presenza di clusters a differente densità di keypoints (Figura 3.7 (b)),
3. presenza di regioni con punti "bridge" che collegano clusters vicini (Figura 3.7 (c)).

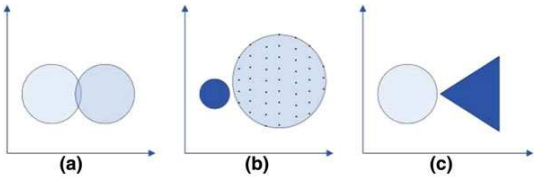
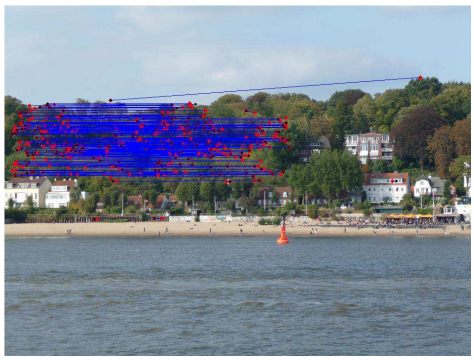


Figura 3.7: Configurazioni spaziali critiche per DBSCAN: (a) clusters sovrapposti, (b) clusters multi-densità, (c) configurazione con punti bridge.

(a) Immagine originale

(b) Immagine manipolata



(c) Descrittori SIFT Simili

(d) DBSCAN clusters

Figura 3.8: L'immagine originale è stata manipolata nascondendo una delle villette presenti con degli alberi limitrofi. In (d) si osserva come DBSCAN non riesca ad individuare i due cluster corrispondenti alle regioni duplicate.

In uno scenario di manipolazioni copy-move di immagini digitali, le configurazioni spaziali critiche per DBSCAN sono piuttosto frequenti: supponiamo che un utente, come mostrato in figura 3.8, duplichi una parte dell'immagine per poi "incollarla" in una regione adiacente; in un simile scenario DBSCAN non sarà in grado di accorgersi della presenza di due regioni duplicate considerando l'intera nuvola di keypoints come un unico cluster.

C-DBSCAN

L'algoritmo di clustering DBSCAN, presentato nella sezione 3.2.2, fa parte di una ampia famiglia di metodi conosciuti in letteratura come non supervisionati: questi metodi vengono utilizzati ogni qualvolta si è di fronte a dati dei quali non è conosciuta la classe di appartenenza; in riferimento allo scenario affrontato in questo lavoro di tesi ricordiamo come l'obiettivo della fase clustering sia di raggruppare a livello spaziale keypoints appartenenti alla medesima regione duplicata.

Recentemente sono state proposte in letteratura, una serie di tecniche definite come semi-supervisionate in cui il processo di formazione dei clusters si avvale di informazioni, espresse sotto forma di vincoli, derivanti dalla conoscenza del dominio applicativo; poter imporre dei vincoli ad un algoritmo di clustering implica la definizione di un modello a priori che guidi il processo di aggregazione dei punti, limitando l'insieme delle soluzioni valide ad un sottoinsieme di quelle possibili. Si individuano due tipologie di vincoli esprimibili su una coppia (p_i, p_j) distinta di punti (vedi Figura 3.9):

1. **vincoli must-link**: esprime il vincolo che la coppia di punti (p_i, p_j) debba appartenere ad uno stesso cluster;

2. **vincoli cannot-link**: esprime il vincolo che la coppia di punti (p_i, p_j) non possa appartenere allo stesso cluster;

A partire da queste osservazioni, nel corso del lavoro di tesi si è notato come la fase di ricerca delle features duplicate definisse una serie di vincoli sui keypoints da clusterizzare: per ogni coppia di keypoints (k_i, k_j) aventi descrittori (f_i, f_j) simili, è possbile definire un vincolo implica un vincolo *cannot-link* in

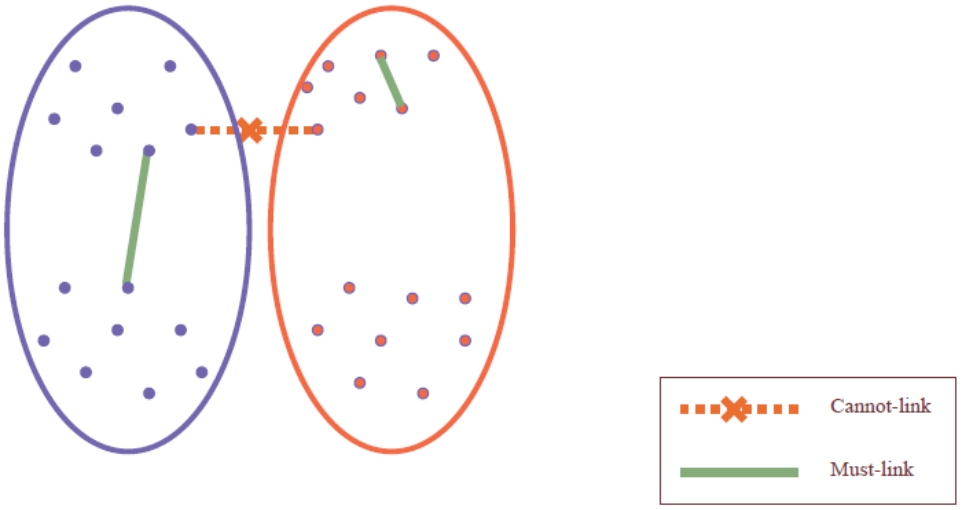


Figura 3.9: Esempio di vincoli di tipo must-link e cannot-link

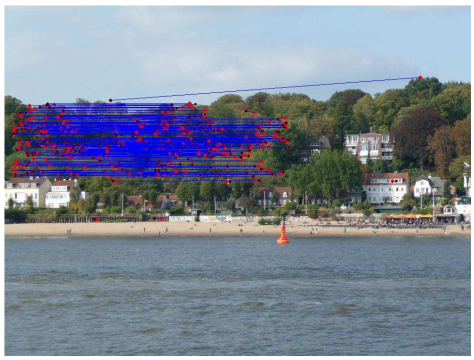
modo tale che k_i e k_j debbano appartenere a cluster diversi. L'algoritmo di clustering semi-supervisionato implementato, conosciuto in letteratura come C-DBSCAN [5] (*Constrained-DBSCAN*) estende la versione adattiva dell'algoritmo DBSCAN illustrato nella sezione precedente, ricevendo in input sia l'insieme dei keypoints che quello dei vincoli *cannot-link* corrispondenti. L'utilizzo di una serie di vincoli di tipo *cannot-link* solleva dei dubbi sulla soddisfacibilità di un algoritmo di clustering semi-supervisionato. Davidson e Ravi [8], hanno infatti dimostrato che il problema della soddisfacibilità di un insieme di vincoli *must-link* e *cannot-link*, imposti ad un algoritmo di clustering partizionale quale ad esempio *K-Means*, risulti NP-Completo. Gli algoritmi semi-supervisionati basati sulla densità, superano i problemi legati alla soddisfacibilità dei vincoli ed in generale quelli legati alla convergenza, lavorando per costruzione su ottimizzazioni locali. In figura 3.10 viene mostrato il risultato dell'esecuzione di C-DBSCAN mostrato relativo all'esempio precedentemente illustrato in Figura 3.8.

J-Linkage

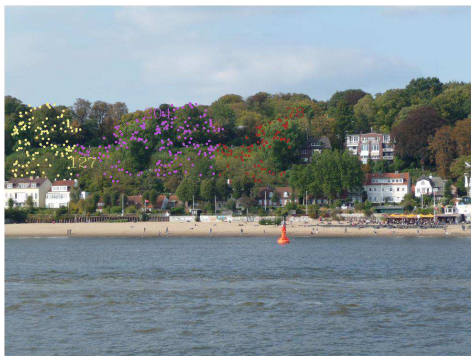
Gli algoritmi di clustering, esposti nelle sezioni precedenti, operano raggruppando punti vicini nello spazio bidimensionale dell'immagine. Nel corso del

(a) Immagine originale

(b) Immagine manipolata



(c) Descrittori SIFT Simili



(d) C-DBSCAN clusters

Figura 3.10: L'immagine originale è stata manipolata nascondendo una delle villette presenti con degli alberi limitrofi. In (d) si osserva come C-DBSCAN a causa dei molteplici vincoli di tipo must-not-link individui un numero di clusters superiore a quello delle reali regioni duplicate.

lavoro di tesi svolto ci siamo chiesti se davvero lo spazio $\Re^2$ dell'immagine fosse quello maggiormente discriminativo al fine di raggruppare keypoints. Il criterio di similarità **g2NN** illustrato nella sezione 3.2.1 permette di ottenere corrispondenze a coppie valutando la similarità dei descrittori SIFT 128-dimensionali; per estendere il concetto di similarità a livello di insieme di keypoint, si è deciso di definire un nuovo criterio di raggruppamento basato non più sulla prossimità spaziale dei punti nell'immagine bensì sulla condivisione di uno o più modelli nello spazio delle trasformazioni affini. L'idea proposta consiste nel riuscire a stimare, sfruttando le informazioni derivanti da keypoints in corrispondenza, i modelli di trasformazione geometrica con i quali l'utente ha eseguito una manipolazione copy-move.

Tra i molteplici approcci di stima robusta presenti in letteratura, abbiamo scelto di utilizzare *J-Linkage* [33]; attraverso un meccanismo di clustering gerarchico J-Linkage è in grado di stimare i modelli parametrici alla base di un processo di manipolazione di tipo copy-move in cui una o più regioni siano state duplicate; vediamo in dettaglio il funzionamento di tale metodo.

Ricevuti in input N punti l'algoritmo J-Linkage esegue M selezioni di punti casuali in cui ogni selezione contiene il numero minimo di punti sufficiente a calcolare i parametri del modello cercato; per ognuno degli M modelli ottenuti viene calcolato il relativo insieme di consenso (CS) [10]; quest'ultimo è formato dall'insieme dei punti d_i che presentano residui ($R(m, d)$), rispetto al modello m, minori di una certa soglia ε. Gli autori dualmente al concetto di insieme di consenso di un modello definiscono l'insieme di preferenza (PS) di un punto d:

$$PS(d, M, \varepsilon) = \{m \in M : R(m, d) < \varepsilon\} \tag{3.6}$$

Gli autori estendono l'equazione 3.6, definendola anche per un insieme D di punti:

$$PS(D, M, \varepsilon) = \bigcap_{d \in D} PS(d, M, \varepsilon) \tag{3.7}$$

La funzione caratteristica dell'insieme di preferenza di un punto d viene utilizzata come rappresentazione di quest'ultimo nel nuovo spazio binario M-dimensionale $\{0, 1\}^m$; punti che provengono da regioni duplicate condividendo gli stessi modelli parametrici, si troveranno molto vicini in questo

spazio facilitando la procedura di clustering gerarchico con la quale J-Linkage identifica i modelli parametrici (procedura 3.11). Il criterio di similarità che J-Linkage adotta per raggruppare in ordine decrescente insiemi di preferenza, è basato sulla distanza di Jaccard; dati due insiemi A e B la distanza di Jaccard viene espressa dalla seguente formula:

$$d_j(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{3.8}$$

In particolare la distanza di Jaccard vale zero per insiemi identici ($A \cup B = A \cap B = A = B$) e vale uno per insiemi disgiunti ($A \cap B = 0$). Il valore di taglio utilizzato come criterio di arresto viene posto pari ad uno, ovvero J-Linkage raggrupperà punti che presentano insiemi di preferenza sovrapposti. Questa valore di taglio permette la formazione di cluster che godono delle seguenti proprietà:

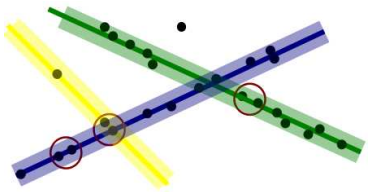
1. ogni cluster contiene al proprio interno punti che condividono almeno un modello,

2. uno stesso modello non può trovarsi all'interno degli insiemi di preferenza dei punti di due cluster distinti.

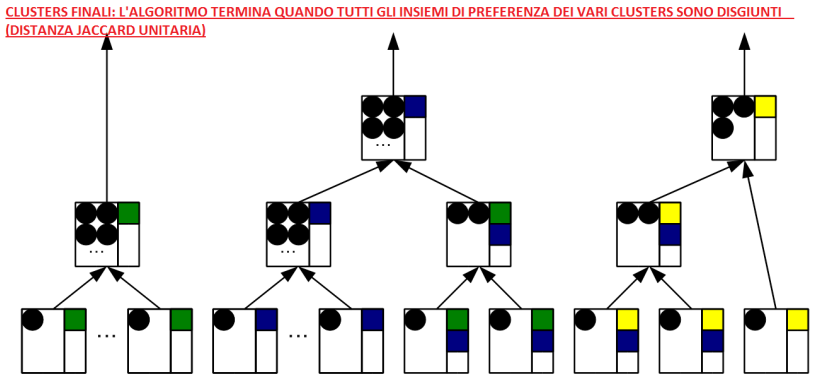
J-Linkage($DataPoints\ D; M \in N; \varepsilon \in R^+; T \in N$)
1. $Clusters = \{\{d\} : d \in DataPoints\}$
2. Calcola M modelli ($models$) a partire da insiemi di set minimali
3. **while** True
   3.1. $A, B = argmin D_j(PS(A, models, \varepsilon), PS(B, models, \varepsilon)$
   3.2. $minDist = D_j(PS(A, Models, \varepsilon), PS(B, Models, \varepsilon))$
   3.3 **if** $minDist == 1$
      3.3.1 **break**
   3.4. $Clusters = (Clusters - \{A, B\}) \cup \{A \cup B\}$
4. **return** $\{c \in Clusters : |c| \geq T$

Figura 3.11: Procedura J-Linkage

In figura 3.12 (a) viene mostrato un esempio di dataset composto da ventisei punti in cui il modello parametrico da stimare corrisponde a quello di una retta; vengono evidenziati in rosso tre tra gli M insiemi minimali selezionati; ciascun insieme minimale è composto da due punti poichè questo è il numero di punti minimo che serve per determinare i parametri di una retta. In figura 3.12 (b) viene illustrata la fase di clustering gerarchico eseguita da J-Linkage: a partire dagli insiemi di preferenza di ogni punto (rettangoli in basso) questi ultimi vengono raggruppati in ordine decrescente rispettivamente alla reciproca distanza di Jaccard (equazione).



(a) Dataset esempio J-Linkage



(b) Esempio esecuzione J-Linkage stima linee

Figura 3.12: Esempio in cui viene mostrata un'esemplificazione dell'esecuzione di J-Linkage nello stimare molteplici modelli di rette.

Nello scenario analizzato in questo lavoro di tesi i modelli parametrici da stimare sono quelli relativi alle trasformazioni indotte dalle manipolazioni copy-move: traslazione, scala e rotazione sono modellabili mediante trasformazioni affini. Una trasformazione affine mette in corrispondenza i punti $\mathbf{x}_i = (x, y)^\intercal$ di una regione R_1 dell'immagine con i punti $\mathbf{x}'_i = (x', y')^\intercal$ di una

regione R_2:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} + \tilde{\mathbf{t}}. \tag{3.9}$$

Una trasformazione affine è caratterizzata completamente da sei gradi di libertà: 2 relativi alla traslazione, 2 relativi alla scala, uno relativo alla rotazione ed uno alla deformazione. Per riuscire a determinare una trasformazione affine sono necessarie almeno tre corrispondenze ($|MSS| = 3$);

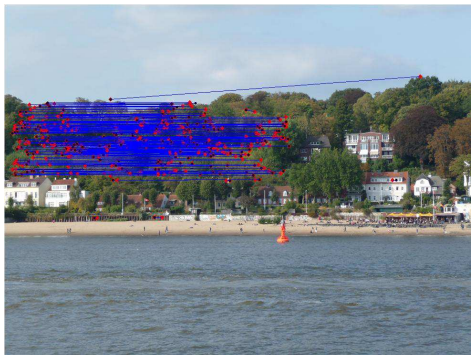
Particolarmente importante risulta la fase di selezione degli M modelli iniziali: la rappresentazione dei keypoints in funzione di questo spazio vettoriale binario ad M dimensioni, dipende da una buona stima dei modelli iniziali; gli autori suggeriscono di utilizzare il metodo proposto da Kanazawa [18] che consiste nel selezionare il primo punto del set minimale in modo random mentre la selezione degli altri punti è soggetta ad una probabilità inversamente proporzionale alla loro distanza; in particolare indicando con x_i un punto già selezionato, la probabilità di selezionare successivamente un punto x_j è pari a:

$$P(x_j|x_i) = \begin{cases} \frac{1}{Z} \exp^{-\frac{\|x_j - x_i\|^2}{\sigma^2}} & \text{se } x_j \neq x_i \\ 0 & \text{se } x_j = x_i \end{cases} \tag{3.10}$$

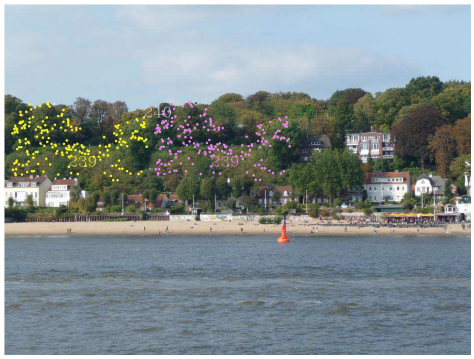
in cui Z e σ vengono stabiliti come fattore di normalizzazione sperimentali. Nel corso degli esperimenti effettuati abbiamo modificato la strategia descritta nell'equazione in quanto la selezione di insiemi minimali effettuata con punti spazialmente troppo vicini introduce rumore nella stima dei parametri di una trasformazione affine con conseguente creazione di micro-clusters in uscita. Si è deciso conseguentemente di utilizzare una soglia sulla distanza minima con la quale i punti debbano essere campionati; questa scelta è corroborata dalla bassa percentuale di outliers presente nell'insieme dei keypoints. In figura 3.13 viene illustrato il risultato dell'esecuzione di J-Linkage sull'immagine di esempio utilizzata in questa sottosezione;

(a) Immagine originale

(b) Immagine manipolata



(c) Descrittori SIFT Simili



(d) Clusters creati da JLinkage

Figura 3.13: L'immagine originale è stata manipolata nascondendo una delle villette presenti con degli alberi limitrofi. In (d) si osserva come JLinkage crei clusters corrispondenti alle aree delle due regioni duplicate.

3.3 Localizzazione regioni duplicate

Le informazioni riportate al termine delle procedura di identificazione di regioni duplicate, sono sufficienti a stabilire se un'immagine sia stata manipolata o meno attraverso un attacco di tipo copy-move; queste informazioni possono non essere sufficienti in scenari di image forensics: all'interno di una corte giudiziaria, l'attività di controllo sulla veridicità di un'immagine digitale prevederà sicuramente un ulteriore attività di ispezione visiva. Creare una maschera di localizzazione permette di determinare in modo accurato le regioni sottoposte a duplicazione. Nel progettare quest'ultima fase, all'interno di questo lavoro di tesi si è previsto l'utilizzo principalmente di due passi:

1. il calcolo delle trasformazioni geometriche T_i tra clusters,
2. il calcolo delle maschere di correlazione ottenute applicando T_i sull'intera immagine.

Le procedure di clustering proposte all'interno del lavoro svolto, forniscono in output un insieme di cluster in cui ogni elemento si trova in corrispondenza con uno o più elementi appartenenti ad altri cluster. Per il calcolo delle trasformazioni affini si è deciso di rappresentare in modo compatto le informazioni di clustering unitamente a quelle di matching tra i descrittori, sfruttando la teoria dei grafi. Si è passati alla definizione di un grafo $G = (V, E)$ pesato (Figura 3.14 (d)) in cui, il set dei nodi V rappresenta l'insieme dei clusters corrispondenti alle regioni duplicate, il set di archi E rappresenta il collegamento tra coppie di regioni duplicate ed infine il peso associato ad ogni arco rappresenta il numero di corrispondenze tra coppie di clusters. Questa nuova forma di rappresentazione viene utilizzata all'interno di un algoritmo di copertura di peso massimo al fine di stabilire il numero di trasformazioni minime per individuare ogni regione duplicata, attraverso coppie di clusters aventi maggior numero di corrispondenze; per il calcolo effettivo delle trasformazioni affini si è utilizzato il metodo di stima robusta RANSAC [10].

L'osservazione che permette di identificare con esattezza l'estensione delle regioni duplicate si basa sulla proprietà che la trasformazione stimata da un

set di corrispondenze discreto, può essere estesa alle regioni dense sulle quali la trasformazione ha realmente agito. Tutti i punti facenti parte di una regione duplicata R_t sono legati ai punti delle regione originale R_o attraverso una medesima trasformazione T_i:

$$R_t = T_i[R_o] \tag{3.11}$$

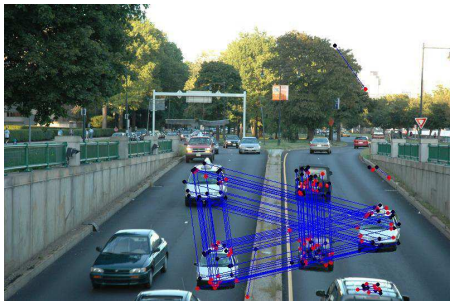
$$R_o = T_i^{-1}[R_t] \tag{3.12}$$

Applicando la trasformazione T_i stimata all'intera immagine, la regione R_o andrà a sovrapporsi alla regione R_t; equivalentemente applicando T_i^{-1} alla regione R_t questa andrà a sovrapporsi alla regione R_o. Per individuare le regioni duplicate, si confrontano i valori dei pixel dell'immagine in input I con quelli dell'immagine J distorta secondo T_i, calcolando la correlazione tra pixels mediante la seguente formula:

$$\mathrm{ZNCC}(I,J) = \frac{\sum_{s\in\Omega_1, t\in\Omega_2} \left(I(s)-\bar{I}\right)\cdot\left(J(t)-\bar{J}\right)}{\sqrt{\sum_{s\in\Omega_1, t\in\Omega_2} \left(I(s)-\bar{I}\right)^2\cdot\left(J(t)-\bar{J}\right)^2}} \tag{3.13}$$

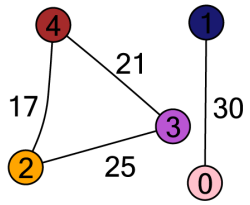
Per ottenere le regioni duplicate si esegue una binarizzazione della maschera di correlazione utilizzando un valore di soglia T_h. La fase di localizzazione esposta in questa sezione viene illustrato all'interno dell'esempio di figura 3.14.

(a) Immagine manipolata

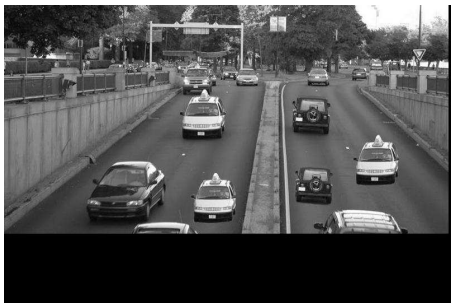


(b) Descrittori SIFT Simili

(c) Clusters creati da JLinkage

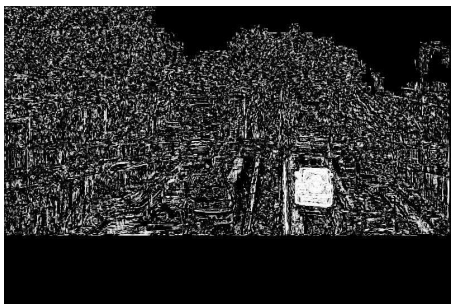


(d) Grafo pesato dei clusters

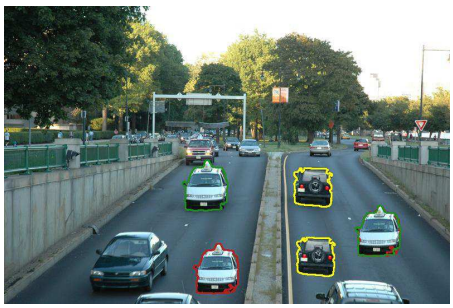


(e) Immagine distorta T

(f) Immagine distorta T^{-1}



(g) Maschera Correlazione



(h) Localizzazione Regioni

Figura 3.14: L'immagine originale è stata manipolata duplicando più volte le vetture presenti nella scena. La trasformazione affine stimata T corrisponde alla duplicazione eseguita sulla vettura di color nero.

Capitolo 4

Risultati Sperimentali

In questo capitolo vengono presentati i dataset utilizzati, la descrizione dei parametri sperimentali ed i risultati ottenuti con le tecniche illustrate nell'approccio proposto; vengono inoltre comparati i risultati ottenuti con quelli di recente pubblicazione.

Nel valutare le prestazioni dell'approccio proposto, si è ritenuto necessario valutare il sistema attraverso una serie di esperimenti in grado di emulare uno scenario reale di image forensics; in particolare, data l'assenza di dataset che rappresentino uno "standard de facto" in queso ambito unita alla volontà degli autori di validare il sistema, sia in termini di identificazione che in termini di localizzazione, si è deciso di utilizzare tre dataset:

1. **DB2000**: dataset utilizzato nel lavoro proposto da Amerini et al. [3], comprende immagini manipolate ed immagini originali; non include maschere di localizzazione delle regioni duplicate.

2. **SATS-130**: dataset utilizzato nel lavoro proposto da Riess et al. [34], comprende esclusivamente un insieme di immagini manipolate; include maschere di localizzazione delle regioni duplicate.

3. **DB-1982**: dataset creato durante il lavoro di tesi, comprende immagini manipolate derivanti dal dataset SATS-130 ed immagini originali derivanti dal dataset DB2000; include maschere di localizzazione delle regioni duplicate.

Analizzando brevemente le caratteristiche di ogni dataset si è ritenuto utile far utilizzo del dataset DB2000 poichè contiene una elevata quantità di immagini, sia originali che manipolate attraverso trasformazioni geometriche/affini; essendo privo di maschere di localizzazione questo dataset è stato utilizzato per valutare la fase di identificazione dell'approccio proposto, confrontandolo con il lavoro di Amerini [3]. Il dataset SATS-130 è composto da 130 immagini contraffatte di cui viene fornita l'esatta maschera di localizzazione; le immagini presenti all'interno del dataset presentano notevole variabilità sia in termini di dimensioni delle immagini che delle rispettive regioni duplicate; gli autori del dataset, all'interno del lavoro proposto [34] comparano le principali tecniche di copy-move detection permettendo un confronto diretto con la soluzione proposta in questo lavoro di tesi. Infine il dataset DB-1982 è stato creato per unire in un'unica collezione di 600 immagini le caratteristiche salienti dei due dataset sopra esposti: in uno scenario di image forensics reale saranno infatti presenti immagini di varia natura e dimensione, alcune delle quali saranno contraffate.

4.1 Metriche di valutazione

Per valutare l'accuratezza con la quale il sistema proposto sia in grado di classificare un'immagine di input come originale o contraffatta, si è scelto di utilizzare due metriche ampiamente utilizzate in problemi di classificazione binaria: **TPR**, **FPR**; prima di illustrare i due indici introduciamo una simbologia comune:

- TP: true positive, si verifica quando un caso positivo viene identificato come tale;
- TN: true negative, si verifica quando un caso negativo è identificato come tale;
- FP: false positive, si verifica quando un caso negativo è identificato come positivo;

- FN: false negative, si verifica quando un caso positivo è identificato come negativo.

TPR rappresenta la frazione di immagini manipolate classificate come tali, mentre FPR rappresenta la frazione di immagini originali non correttamente classificate come tali:

$$\mathbf{TPR} \doteq \frac{TP}{TP + FN} \tag{4.1}$$

$$\mathbf{FPR} \doteq \frac{FP}{FP + TN} \tag{4.2}$$

L'accuratezza di identificazione di un algoritmo di copy-move detection sarà tanto più elevata quanto maggiore sarà l'indice TPR e tanto minore l'indice FPR.

Al fine di valutare l'accuratezza di localizzazione di regioni duplicate, si è deciso di utilizzare due indici ([23], [4]) diffusi nell'ambito dell'image forensics: l'indice F_P rappresenta la percentuale di regioni identificate erroneamente come manipolate, mentre l'indice F_N indica la percentuale di regioni manipolate non rilevate come tali. In particolare indicando con R_1 la regione selezionata per la copia, con $R_i, i > 1$ la i-esima regione duplicata ed "incollata" ed infine con B l'insieme di regioni di background possiamo scrivere:

$$\mathbf{F_P} = \frac{|matches\ in\ B|}{|B|} \tag{4.3}$$

$$\mathbf{F_N} = \frac{|missed\ matches\ in\ (\cup_i R_i)|}{|\cup_i R_i|} \tag{4.4}$$

L'accuratezza di localizzazione di regioni duplicate di un algoritmo di copy-move detection sarà tanto più elevata quanto minori saranno gli indici FP ed FN.

4.1.1 DB2000

Il dataset DB2000 come anticipato in precedenza è composto da 2000 immagini di cui 1300 sono originali mentre le restanti 700 hanno subito contraffazioni

Attack	θ	s_x	s_y
a	0	1	1
b	0	0.5	0.5
c	0	0.7	0.7
d	0	1.2	1.2
e	0	1.6	1.6
f	0	2	2
g	0	1.6	1.2

Attack	θ	s_x	s_y
h	0	1.2	1.6
i	5	1	1
j	30	1	1
l	70	1	1
m	90	1	1
n	40	1.1	1.6
o	30	0.7	0.9

Tabella 4.1: Le 14 differenti tipologie di trasformazioni geometriche applicate a regioni appartenenti ad immagini del dataset DB2000.

di tipo copy-move. Le 700 immagini contraffatte sono state ottenute applicando 14 tipologie di manipolazioni, a 50 diverse immagini originali (vedi Tabella 4.1).

Si è deciso di eseguire esperimenti utilizzando il dataset DB2000 sia per avere un confronto con una tecnica di copy-move detection [3] recente sia per avere una maggiore comprensione degli algoritmi di copy-move detection implementati durante il lavoro di tesi; la logica applicativa con la quale sono stati definiti i molteplici esperimenti si basa sul testare ogni blocco funzionale del sistema, in particolare durante la serie di esperimenti abbiamo testato il comportamento del criterio di matching g2NN proposto nei confronti del criterio 2NN, analizzandone gli effetti sui tre algorimti di clustering implementati: DBSCAN, C-DBSCAN ed infine J-Linkage. Per le sei configurazione del sistema proposto abbiamo fatto variare il numero minimo di corrispondenze (Pts), utilizzato come criterio decisionale sulla presenza o meno di una manipolazione all'interno dell'immagine; in tabella 4.2 sono riportati i risultati dell'esperimento sopra descritto in termini quantitativi.

Dall'analisi di figura 4.1 emerge in modo evidente, come il criterio di matching g2NN introdotto non influisca sulle performance di detection dei vari algoritmi; si ricorda al lettore che l'introduzione del criterio di similarità g2NN è stata introdotta in contesti di copy-move caratterizzati da più copie della stessa regione; per quanto concerne gli algoritmi di clustering,

	DBSCAN 2NN		*C-DBSCAN 2NN*		*JLinkage 2NN*	
Pts	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)
4	**0,881**	**0,034**	0,938	0,352	0,931	0,169
5	0,865	0,022	0,917	0,311	0,902	0,117
6	0,828	0,013	0,888	0,284	0,871	0,094
7	0,798	0,009	0,852	0,254	0,852	0,066
8	0,776	0,008	0,838	0,234	0,825	0,057
	DBSCAN g2NN		*C-DBSCAN g2NN*		*JLinkage g2NN*	
Pts	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)
4	0,878	0,033	0,939	0,356	0,934	0,175
5	0,864	0,021	0,92	0,317	0,904	0,120
6	0,83	0,013	0,891	0,289	0,874	0,092
7	0,798	0,010	0,856	0,256	0,852	0,071
8	0,778	0,008	0,842	0,235	0,829	0,055

Tabella 4.2: Valori di TPR ed FPR ottenuti in funzione del numero di punti Pts utilizzato per identificare regioni duplicate.

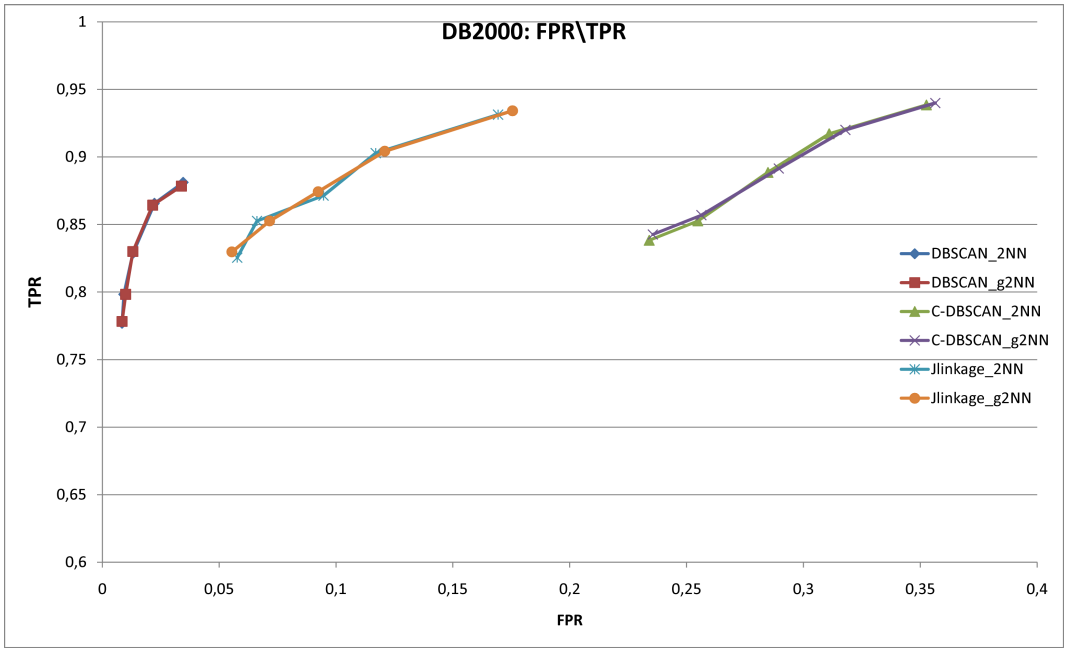


Figura 4.1: Visualizzazione qualitativà delle differenze di performance degli algoritmi implementati sul dataset DB2000.

sul dataset in esame DBSCAN presenta il tasso più basso di falsi positivi mantenendo elevata la capacità di rilevare region duplicate. Osservando la tabella 4.2, il risultato ottenuto dall'algoritmo dbscan con quattro punti ($TPR = 88\%$,$FPR = 3\%$) è in linea con quanto pubblicato da Amerini [3] ($TPR = 93\%$,$FPR = 11\%$) in quanto ad una minor tasso di detection, quantificabile in cinque punti percentuali corrisponde un minor tasso di falsi positivi quantificabile in otto punti percentuali; a differenza del setup sperimentale da loro utilizzato, gli algoritmi implementati in questo lavoro di tesi non necessitano di fasi di addestramento e validazione, per la proprietà adattiva di cui godono DBSCAN e derivati (vedi equazione 3.4);

4.1.2 SATS-130

Si è deciso di utilizzare il dataset SATS-130 [34], composto da un totale di 130 immagini accuratamente manipolate, per confrontare il sistema proposto con le tecniche stato dell'arte di copy-move detection. Nelle creazione del

dataset gli autori hanno selezionato 10 immagini, eterogenee tra di loro nelle dimensioni, nella gamma dei colori e nella tessitura (Figura **??**) creando un insieme finale composto da 130 immagini; il processo di manipolazione delle singole immagini avviene duplicando e ruotando ognuna di esse per tredici volte: a partire da una rotazione nulla, ogni regione duplicata viene ruotata di cinque gradi.

(a) Immagine manipolata bue

(b) Maschera di localizzazione delle regioni duplicate

(c) Immagine manipolata bue

(d) Maschera di localizzazione delle regioni duplicate

Figura 4.2: Esempio di immagine manipolata e relativa maschera di localizzazione appartenente al dataset SATS-130.

Uno degli aspetti interessanti di questo dataset sono le accurate maschere di localizzazione relative alle singole immagini manipolate (Figura 4.2); attraverso gli indici F_p ed F_n è cosi possibile verificare l'accuratezza di localizzazione della regione a livello del singolo pixel, valutando le prestazioni di un sistema di identificazione e localizzazione di regioni duplicate in modo

rigoroso. Abbiamo cosi deciso di replicare l'esperimento originale definito dagli autori di SATS-130 al fine di poter comparare i risultati della nostra tecnica con quelli pubblicati dagli autori di SATS. Illustriamo prima di tutto

	Metodi Originali		Metodi SATS	
Feat	Fp	Fn	Fp	Fn
INT2	4 ± 3	96 ± 9	0 ± 0.4	22 ± 2
INT4	24 ± 19	66 ± 30	0 ± 0	41 ± 32
MOM3	0.4 ± 1	88 ± 24	0 ± 0.0	23 ± 1
DBSCAN-2NN	0.002	0.23		
DBSCAN-G2NN	0.002	0.23		
C-DBSCAN-2NN	0.002	0.23		
C-DBSCAN-G2NN	0.003	0.20		
J-LINKAGE-2NN	**0.008**	**0.14**		
JLINKAGE-G2NN	**0.008**	**0.14**		

Tabella 4.3: Misura delle performance di localizzazione degli algoritmi descritti in [34] e di quelli proposti nel lavoro di tesi.

i metodi utilizzati dagli autori per validare il proprio approccio di stima di trasformazione affine a partire da blocchi per poi analizzare i risultati complessivi. In tabella 4.3 sono riportati i valori dei tre approcci ritenuti ottimali dagli autori dell'articolo. Tutti e tre i metodi considerati lo stato dell'arte nell'articolo ($INT2$, $INT4$ e $MOM3$) sono stati illustrati all'interno del capitolo 2. In tabella 4.3 abbiamo riportato nella colonna di sinistra i risultati ottenuti su SATS-130 dai metodi suggeriti dagli autori insieme ai i risultati relativi alle varie componenti del nostro sistema; nella parte superiore della colonna di destra troviamo i risultati ottenuti dai metodi della colonna di sinistra dopo essere stati raffinati attraverso il metodo SATS proposto dagli autori.

Analizzando i risultati ottenuti dai vari metodi utilizzati nell'articolo, risulta che il metodo descritto dagli autori, SATS, riesca ad apportare un notevole incremento delle performance degli algoritmi basati sui blocchi; l'approccio proposto nel presente lavoro di tesi non necessita di tale fase di stima

e raffinamento in quanto la stima delle trasformazioni affini avviene durante la normale fase di esecuzione del metodo.

4.1.3 DB-1982

Mentre gli esperimenti eseguiti sul dataset DB2000 hanno dato indicazioni utili sull'accuratezza di detection del sistema proposto, quelli eseguiti sul dataset SATS-130 composto da sole immagini manipolate ha fornito indicazione sull'accuratezza di localizzazione del sistema. Si è sentita la necessità di creare un dataset che contenesse al proprio interno sia immagini manipolate che immagini originali ed inoltre fornisse per ogni immagine manipolata un'opportuna maschera di localizzazione; a partire da queste considerazioni si è deciso di creare un dataset di 600 immagini cosi composto:

- 440 immagini originali provenienti dal dataset DB2000,
- 160 immagini manipolate, a partire da 40 immagini originali appartenenti al dataset SATS-130, cosi definite:
  - 40 immagini in cui una regione sia stata duplicata più volte.
  - 40 immagini in cui la regione sia stata copiata e successivamente solo traslata (copy-move base),
  - 40 immagini in cui una regione sia stata ruotata di 30 gradi,
  - 40 immagini in cui una regione sia stata ruotata di 30 gradi e scalata del 120%;

La composizione sopra illustrata è sufficientemente generale da permettere di testare il metodo proposto in condizioni equivalenti a quelle reali. Il sistema proposto viene analizzato in modo esaustivo a partire dalla fase di estrazione dei dati sino alla fase di localizzazione delle regioni duplicate. Coerentemente con i passi illustrati precedentemente analizziamo dapprima i risultati relativi all'identificazione e successivamente quelli legati alla localizzazione.

Dall'analisi di figura 4.3 emerge come il criterio di matching g2NN introdotto non influisca sulle performance di detection dei vari algoritmi; l'analisi

	DBSCAN 2NN		*C-DBSCAN 2NN*		*JLinkage 2NN*	
Pts	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)
4	0,717	0,056	0,901	0,420	0,894	0,375
5	0,675	0,036	0,887	0,382	0,860	0,264
6	0,664	0,025	0,888	0,309	0,855	0,202
7	0,609	0,025	0,827	0,252	0,821	0,113
8	0,62	0,020	0,826	0,218	**0,82**	**0,09**
	DBSCAN g2NN		*C-DBSCAN g2NN*		*JLinkage g2NN*	
Pts	TPR(%)	FPR(%)	TPR(%)	FPR(%)	TPR(%)	FPR(%)
4	0,710	0,052	0,901	0,425	0,894	0,379
5	0,66	0,034	0,887	0,384	0,860	0,271
6	0,657	0,025	0,881	0,313	0,855	0,215
7	0,602	0,025	0,827	0,270	0,821	0,125
8	0,606	0,022	0,826	0,225	0,82	0,097

Tabella 4.4: Valori di TPR ed FPR ottenuti in funzione del numero di punti Pts utilizzato per identificare regioni duplicate.

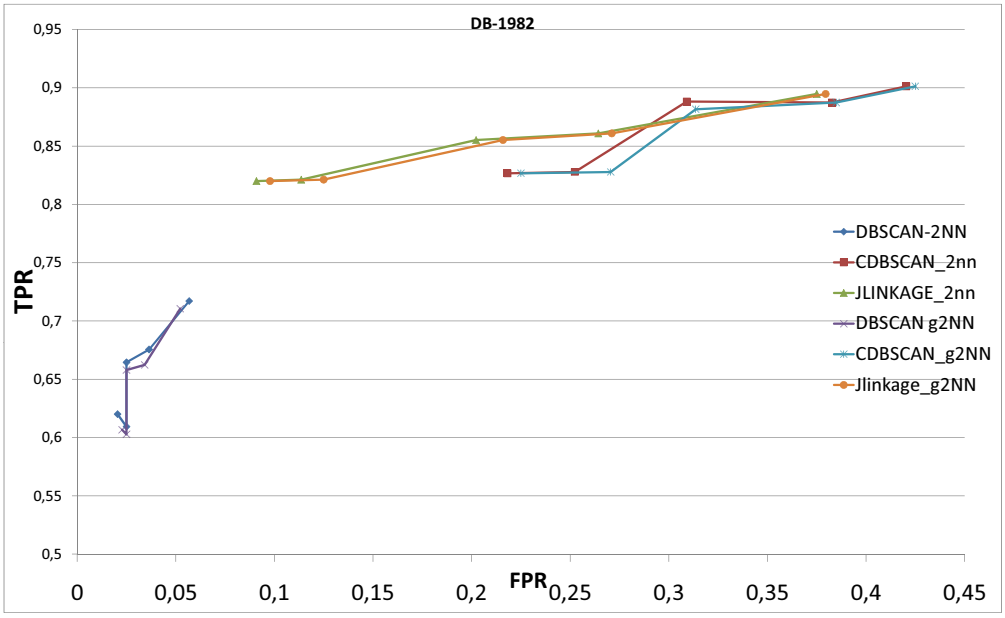


Figura 4.3: Visualizzazione qualitativà delle differenze di performance degli algoritmi implementati sul dataset DB-1982.

degli algoritmi di clustering rivela come J-Linkage in un dataset più eterogeneo del DB2000 non risenta di una diminuzione di accuratezza; la presenza di un tasso di falsi positivi maggiore rispetto ad esempio a DBSCAN, è dovuta principalmente alla presenza in natura di oggetti reali che rappresentano pattern naturali che un algoritmo basato sulla stima di trasformazioni quale J-Linkage interpreterà come regioni duplicate; rispetto alla figura 4.1 la posizione relativa in termini di fpr, tra la curva di DBSCAN, quella di C-DBSCAN ed infine quella di J-Linkage rimane invariata, il cambiamento che si ha in valore assoluto è dovuto alla perdita di detection da parte di DBSCAN; questa può essere semplicemente dovuta al fatto che le regioni duplicate siano considerate limitrofe e quindi non vengono rilevate.

	DBSCAN 2NN		*C-DBSCAN 2NN*		*JLinkage 2NN*	
Pts	$F_P(\%)$	$F_N(\%)$	$F_P(\%)$	$F_N(\%)$	$F_P(\%)$	$F_N(\%)$
4	0,0018	0,1281	0,002	0,083	0,003	0,058
5	0,0016	0,1380	0,002	0,085	0,003	0,058
6	0,0018	0,1300	0,002	0,083	0,003	0,060
7	0,0018	0,1312	0,002	0,086	0,003	0,065
8	0,0017	0,130	0,002	0,090	0,003	0,065
	DBSCAN g2NN		*C-DBSCAN g2NN*		*JLinkage g2NN*	
Pts	$F_P(\%)$	$F_N(\%)$	$F_P(\%)$	$F_N(\%)$	$F_P(\%)$	$F_N(\%)$
4	0,002	0,127	0,003	0,079	0,003	0,055
5	0,001	0,135	0,003	0,080	0,003	0,056
6	0,002	0,130	0,003	0,082	0,003	0,058
7	0,002	0,136	0,003	0,085	0,003	0,063
8	0,001	0,131	0,002	0,085	0,003	0,063

Tabella 4.5: Valori di F_P ed F_N ottenuti in funzione del numero di punti Pts utilizzato per localizzare le regioni duplicate.

L'acccuratezza di localizzazione calcolata attraverso i due indici F_p ed F_n viene mostrata in figura 4.4, rappresentazione qualitativa della tabella 4.5.

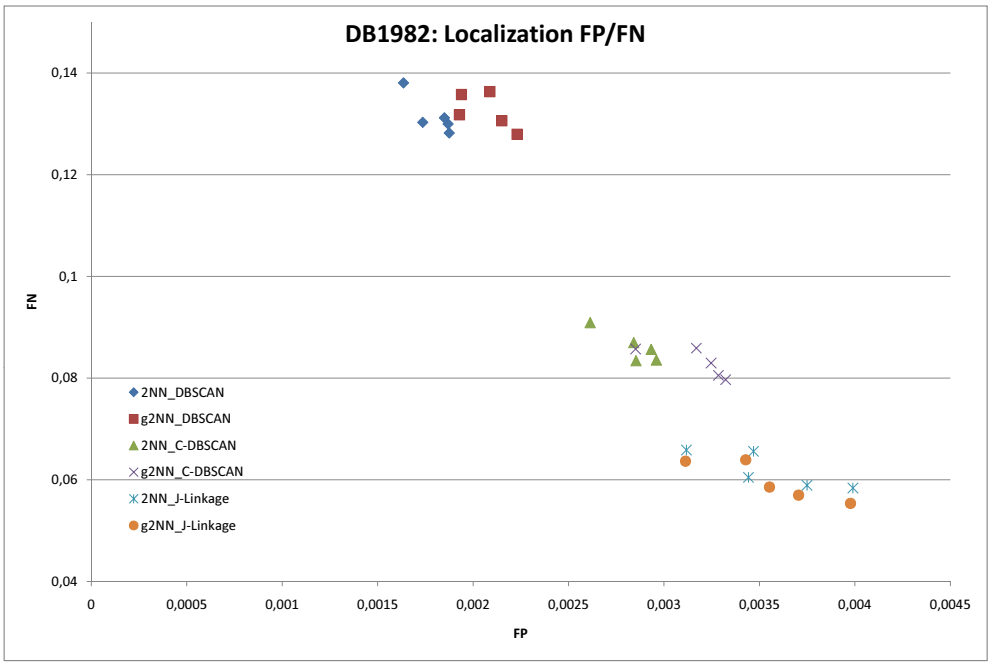


Figura 4.4: Visualizzazione qualitativa delle differenze di performance degli algoritmi implementati sul dataset DB-1982.

4.2 Conclusioni e sviluppi futuri

In questo lavoro di tesi è stata sviluppata ed implementata una soluzione al problema dell'individuazione e localizzazione di manipolazioni di tipo copy-move in applicazioni di image forensics. La soluzione proposta, consiste nel rappresentare un'immagine attraverso un insieme di punti caratteristici rappresentati da descrittori di tipo SIFT, eseguire una fase di ricerca volta a trovare punti corrispondenti che permettano di stimare l'eventuale trasformazione geometrica compiuta dall'utente durante la manipolazione dell'immagine. I risultati sperimentali hanno mostrato come l'approccio proposto garantisca un'elevata accuratezza sia nella fase di identificazione che nella seguente di localizzazione nei confronti di una regione duplicata all'interno di un'immagine. Data la natura discreta con cui viene rappresentata un'immagine, un naturale sviluppo di tesi consiste nell'individuare una strategia di rappresentazione dell'immagine che garantisca copertura su ogni zona dell'immagine in input.

Bibliografia

[1] *Characteristic Functions in Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[2] M.Wu A. Swaminathan and K. J. Ray Liu. Non-intrusive forensics analysis of visual sensors using output images. In *Proc. of IEEE ICIP*, 2006.

[3] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. Geometric tampering estimation by means of a sift-based forensic analysis. In *Proc. of IEEE ICASSP*, Dallas, USA, 2010.

[4] Sergio Bravo-Solorio and Asoke K. Nandi. Passive method for detecting duplicated regions affected by reflection, rotation and scaling. In *European Signal Processing Conference*, 2009.

[5] E. M. Ruiz C. Ruiz, M. Spiliopoulou. C-dbscan: Density-based clustering with constraints. *RSFDGrC, volume 4482, pages 216–223*, 2007.

[6] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Trans. on Information Forensics and Security*, 3(1):74–90, 2008.

[7] Kevin Cohen. Digital still camera forensics. *SMALL SCALE DIGITAL DEVICE FORENSICS JOURNAL*, '1 no. 1.

[8] Ravi SS Davidson I. Clustering with constraints: feasibility issues and the k-means algorithm. In *SIAM'05 society for industrial and applied mathematics international conference on data mining, 2005.*

[9] R. Duda and P. Hart. Pattern classification and scene analysis. *John Wiley and Sons*, 1973.

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM 24, June 1981.*

[11] G. Friedman. The trustworthy digital camera: Restoring credibility to the photographic image. In *IEEE Transactions on Consumer Electronics*, 1993.

[12] H. Huang, W. Guo, and Y. Zhang. Detection of copy-move forgery in digital images using sift algorithm. In *Proc. of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008.

[13] S. Saic J. Flusser, T. Suk. Image features invariant with respect to blur. *Pattern Recogn. 28*, 1995.

[14] D. Soukal J. Fridrich and J. Lukas. Detection of copy–move forgery in digital images. In *IEEE Computer Society, Cleveland, OH, USA (August 2003).*

[15] Hongyuan Li Yuewei Dai Zhiquan Wang Junwen Wang, Guangjie Liu. Detection of image region duplication forgery using model with circle block. 2009.

[16] E. Y. Lam K. S. Choi and K. K. Y. Wong. Source camera identification using footprints from lens aberration. In *Proc. of SPIE*, 2006.

[17] Ser W Kakar P, Sudha N. Exposing digital image forgeries by detecting discrepancies in motion blur. *Information Forensics and Security, IEEE Transactions on*, 2011.

[18] Kawakami H. Kanazawa, Y. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *British Machine Vision Conference, 2004*.

[19] Mohan S Kankanhalli Lanh Tran Van, Sabu Emmanuel. Identifying source cell phone using chromatic aberration. In *IEEE International Conference on Multimedia and Expo*, 2007.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[21] J. Lukas and J. Fridrich. Estimation of primary quantization matrix in double compressed jpeg images. In *Proc. of DFRWS*, 2003.

[22] J. Lukáš, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. In *In Proceedings of the SPIE*, page 2006. West, 2006.

[23] W. Luo, J. Huang, and G. Qiu. Robust detection of region-duplication forgery in digital image. In *Proc. of ICPR*, Washington, D.C., USA, 2006.

[24] D.L. Massart M. Daszykowski, B. Walczak. Looking for natural patterns in analytical data: tracing local density with optics. *Journal of Chemical Information and Computer Sciences*, 2002.

[25] B. Mahdian and S. Saic. Detection of copy-move forgery using a method based on blur moment invariants. *Forensic Science International*, 171(2-3):180–189, 2007.

[26] Jorg Sander e Xiaowei Xu Martin Ester, Hans-Peter Kriegel. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining, pages 226–231,1966*.

[27] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[28] X. Pan and S. Lyu. Detecting image region duplication using SIFT features. In *Proc. of IEEE ICASSP*, Dallas, USA, 2010.

[29] A.C. Popescu and H. Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Dartmouth College, Computer Science, 2004.

[30] Seung-Jin Ryu, Min-Jeong Lee, and Heung-Kyu Lee. Detection of copy-rotate-move forgery using zernike moments. In *International Workshop on Information Hiding*, 2010.

[31] Matthew Stamm and K. J. Ray Liu. Blind forensics of contrast enhancement in digital images. In *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP 2008).*

[32] S. F. Chang T. Ng and Q. Sun. Blind detection of photomontage using higher order statistics. In *Proc. of ISCAS*, 2004.

[33] R. Toldo and A. Fusiello. Robust multiple structures estimation with jlinkage. In *In Proc. European Conf. Computer Vision, pages 537,547, 2008.*

[34] Elli Angelopoulou Vincent Christlein, Christian Riess. On rotation invariance in copy-move forgery detection. In *IEEE Workshop on Information Forensics and Security*, 2010.

Ringraziamenti

Desidero innanzitutto ringraziare il Prof. Alberto Del Bimbo per avermi dato unitamente al Prof. Alessandro Piva l'opportunitá di svolgere questo lavoro di tesi. Un ringraziamento particolare va inoltre al neonato gruppo del "forensic" composto da Irene, Lamberto, Roberto e Giuseppe con cui ho condiviso l'intero percorso di tesi (purtroppo per loro!!).

Un doveroso ringraziamento va inoltre ai miei genitori ed alla mia ragazza, senza di cui non avrei potuto raggiungere questo traguardo.

Un grandissimo grazie va a tutti i miei amici che mi hanno accompagnato nell'avventura di questi anni. In particolare a: Alessio, Alessandro, Luca e Gianluca.