

UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

FACOLTÀ DI INGEGNERIA

CORSO DI STUDI IN INGEGNERIA INFORMATICA



TESI DI LAUREA SPECIALISTICA

**Sviluppo di una tecnica per l'identificazione di
immagini contraffatte basata sul rumore
introdotta dalle fotocamere digitali**

ANNO ACCADEMICO 2009/2010

RELATORI

Ch.mo prof. **Luisa Verdoliva**

Ch.mo prof. **Carlo Sansone**

CANDIDATO

Giovanni Chierchia

matr. **885/201**

Alla mia famiglia.

Ai miei amici più cari.

*A tutti quelli che, nel bene o nel male,
hanno fatto di me ciò che sono.*



SOMMARIO

UN APPROCCIO MOLTO POTENTE per il riconoscimento delle immagini contraffatte si basa su un disturbo introdotto involontariamente dalle fotocamere digitali, il *Photo Response Non-Uniformity*. Il PRNU nasce dalle imperfezioni del wafer di silicio usato per fabbricare il sensore d'immagini. Esso è una sorta di *impronta digitale* che può essere utilizzata, tra le altre cose, per individuare le contraffazioni. L'operazione chiave per stimare il PRNU di un'immagine è il filtro di denoising. L'obiettivo primario di questo lavoro è stabilire l'importanza del denoising nel processo globale di riconoscimento delle contraffazione. A questo proposito, una parte importante della tesi è stata dedicata allo studio e l'implementazione delle più recenti tecniche di denoising. Un ulteriore contributo è stato quello di proporre un approccio alternativo per il riconoscimento delle contraffazione, basato sulle reti neurali. Il lavoro di tesi è organizzato come segue:

Il primo capitolo descrive la tecnica di riconoscimento delle contraffazioni proposta nell'articolo [3] da *Fridrich et al, 2008*.

Il secondo capitolo fornisce una panoramica delle tecniche di denoising, partendo dai metodi più elementari fino ad arrivare agli algoritmi che sono lo stato dell'arte in questo settore.

Il terzo capitolo è dedicato ai risultati sperimentali. Nella prima parte, si forniscono alcuni dettagli implementativi e, nella seconda parte, si mostra l'esito degli esperimenti con le soluzioni proposte nella tesi.



INDICE

SOMMARIO	ii
INTRODUZIONE	v
CAPITOLO 1. RILEVARE LE CONTRAFFAZIONI MEDIANTE IL PRNU	1
1.1 Processo di acquisizione digitale	1
1.1.1 Cattura di un'immagine	2
1.1.2 Modello di rumore	4
1.2 Impronta digitale della fotocamera	9
1.2.1 Stima del PRNU	11
1.2.2 Pre-elaborazione del PRNU	13
1.3 Verifica di integrità	14
1.3.1 Test d'ipotesi	15
1.3.2 Riduzione dei falsi negativi	19
1.3.3 Predittore	22
CAPITOLO 2. TECNICHE DI DENOISING	24
2.1 Restauro di immagini	24
2.2 Tecniche nel dominio spaziale	25
2.2.1 Filtro a media pesata	27
2.2.2 Filtro di Wiener	28
2.3 Tecniche nel dominio wavelet	30

2.3.1	Shrinkage marginale	31
2.3.2	Shrinkage contestuale	32
2.3.3	Shrinkage in due passi	33
2.4	Tecniche non locali	34
2.4.1	Algoritmo NL-means	35
2.4.2	Algoritmo BM3D	36
2.5	Confronto delle prestazioni	38
CAPITOLO 3. RISULTATI SPERIMENTALI		43
3.1	Introduzione	43
3.2	Preparazione della fotocamera	44
3.2.1	Stima del PRNU	46
3.2.2	Stima delle ipotesi	48
3.2.3	Addestramento del predittore	50
3.3	Verifica d'integrità	53
3.3.1	Esperimenti sul denoising	56
3.3.2	Esperimenti sulla compressione JPEG	60
3.3.3	Esperimenti con le reti neurali	61
BIBLIOGRAFIA		67
INDICE DELLE FIGURE		68
INDICE ANALITICO		70





INTRODUZIONE

LA FOTOGRAFIA È UNO STRUMENTO MOLTO POTENTE che ci accompagna, ormai da quasi due secoli, in moltissimi aspetti della nostra vita quotidiana. Il primo approccio alla fotografia risale ai primi anni del 1800, ma solo intorno alla metà del secolo si raggiunse un livello di maturità del processo fotografico tale da poter scattare un foto senza che il soggetto dovesse rimanere in esposizione per molti minuti. Con il passare del tempo, la fotografia cominciò ad essere usata non solo per ritrarre paesaggi o persone, ma anche in ambito commerciale, giornalistico e politico. Oggigiorno, le foto sono usate anche in campo giuridico, non di rado infatti sono impiegate come testimoni silenziosi di fatti avvenuti.

Il continuo progresso tecnologico ha portato a tantissime innovazioni, fino ad arrivare ai nostri giorni, dove il processo fotografico ha subito una rapida modifica con l'avvento delle macchine digitali. Oggigiorno siamo letteralmente invasi da foto. Esse irrompono nella nostra vita e colpiscono la nostra attenzione tanto da rimanere indelebili nella nostra mente anche col passare degli anni. Tuttavia, non possiamo *sempre* credere alle foto che vediamo. Infatti, nel 1860 le fotografie venivano già modificate, appena pochi decenni dopo che Niepce creasse la prima fotografia, nel 1814.

Una delle prime immagini manipolate è quella che ritrae il presidente Abramo Lincoln, risalente al 1860 circa (vedi Figura 1). La modifica che fu realizzata consisté nell'inserire la testa del presidente sul corpo di un altro uomo, il politico John Calhoun. Questo fu fatto perché non c'erano immagini di Lincoln che lo ritraevano in uno stile *eroico*.

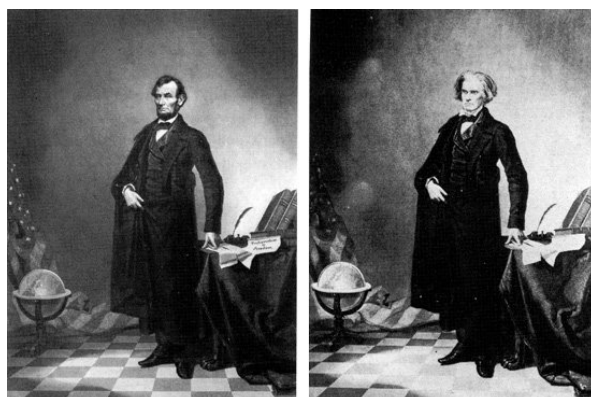


Figura 1 Falso di Abramo Lincoln (a sinistra) e la foto originale, 1860.

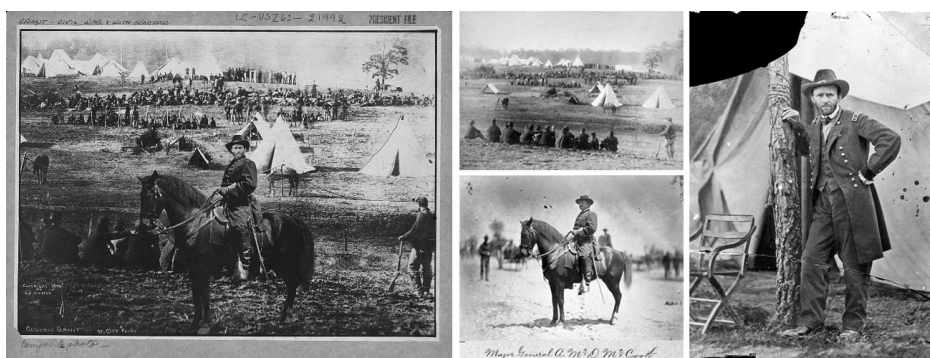


Figura 2 Falso del generale Grant (a sinistra) e le tre immagini originali, 1864.

Un altro falso famoso, risalente al 1864, è quello che ritrae il generale Ulysses S. Grant di fronte le sue truppe sul campo di battaglia a City Point, Virginia, durante la guerra civile americana (vedi Figura 2). Un attento lavoro di ricerca alla Libreria del Congresso ha rivelato che quest'immagine è in realtà l'ottima composizione di tre distinte immagini: (1) la testa in questa foto è presa da un ritratto di Grant, (2) il cavallo e il corpo sono presi dal Generale maggiore Alexander M. McCook, e (3) lo sfondo raffigura dei prigionieri catturati alla battaglia di Fisher's Hill.

Nel 1865 la foto del famoso fotografo Mathew Brady che ritraeva il generale Sherman con i suoi generali fu modificata al fine di aggiungere il generale Francis P. Blair (vedi Figura 3).

Nel 1942, per dare l'idea di un stile prode ed eroico, fu modificata la foto di Benito Mussolini eliminando la persona che manteneva fermo il cavallo.



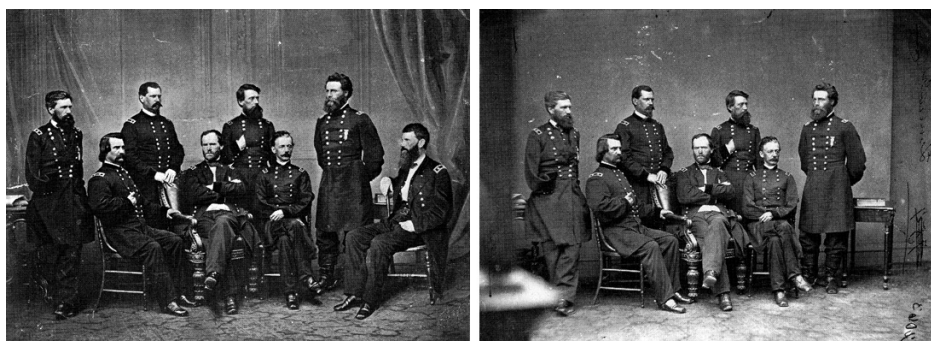


Figura 3 Falso del generale Blair (a sinistra) e la foto originale, 1865.



Figura 4 Falso di Mussolini (a sinistra) e la foto originale, 1942.



(a) Millard Tydings.



(b) Earl Browder.



(c) Fotomontaggio.

Figura 5 Falso di Browder e Tydings, 1950.





Figura 6 Falso di Oprah Winfrey (a sinistra) e il corpo usato per crearlo, 1989.



Figura 7 Falso di Martha Stewart (a sinistra) e il suo vero corpo, 2005.

Nel 1950 una foto modificata contribuì sensibilmente alla sconfitta elettorale del senatore Millard Tydings (vedi Figura 5). Tale foto lo ritraeva mentre conversava con Earl Browder, un leader del partito comunista americano. Questo promosse l'idea che Tydings fosse un simpatizzante comunista, quando il *maccartismo* era agli apici della diffusione.

Con l'avvento di computer sempre più potenti e software più sofisticati, la creazione di fotomontaggi è diventata sempre più semplice. Tuttavia il tipo di modifica apportata è rimasto sostanzialmente invariato nel corso degli anni. A titolo di esempio, mettere la testa di una persona sul corpo di un'altra è ancora una pratica molto diffusa.

Nel 1989 la copertina di *Tv Guide* mostrava la famosa conduttrice Oprah Winfrey, ma in realtà si trattava di un falso: la testa di Oprah sul corpo di un'attrice (vedi Figura 6). Il falso fu creato senza l'autorizzazione dell'attrice, che scoprì l'inganno riconoscendo il vestito nella foto.





(a) Fotomontaggio, 2004.



(b) Immagini originali, 1972.

Figura 8 Falso di John Kerry e Henry Fonda, 2004.

Nel 2005 la copertina di Newsweek presentava una foto di Martha Stewart con una didascalia che alludeva al suo dimagrimento dovuto alla prigionia (vedi Figura 7). In realtà, la testa della Stewart era stata poggiata sul corpo, decisamente più magro, di una modella.

Un tipo di alterazione molto frequente è l'accostamento di persone, come nel falso di Tydings. Un esempio moderno è quello del Febbraio 2004, quando il senatore John Kerry fu accostato all'attivista Jane Fonda durante un comizio contro la guerra del Vietnam (vedi Figura 8).

Le manipolazioni non riguardano solo il mondo dello spettacolo o della politica, ma sono presenti anche in contesti più pericolosi, come nel caso delle immagini di guerra. Un esempio è quello dell'Aprile 2003: si tratta della foto di un soldato Inglese in Basra, Iraq, che con dei gesti intima ai civili iracheni di abbassarsi (vedi Figura 9). Questa foto, scattata da





(a) Immagini originali.



(b) Fotomontaggio.

Figura 9 *Falso del militare che intima ai civili di abbassarsi, 2003.***Figura 10** *Falso (a destra) che mostra il fumo dei bombardamenti, 2006.*

Brian Walski, venne pubblicata sulla copertina del Los Angeles Times dopo l'invasione dell'Iraq da parte degli Stati Uniti. Quando l'editore scoprì che la foto in realtà era la composizione di altre due foto scattate dallo stesso fotografo non esitò a licenziarlo nonostante fosse un veterano nel suo campo con 30 anni di esperienza quale fotografo.

Un altro esempio è quello dell'Agosto 2006, quando l'agenzia Reuters pubblicò sul proprio sito una foto scattata da Adnan Hajj, un fotografo libanese, che ritraeva l'esito di un bombardamento aereo di una città del Libano da parte degli israeliani (vedi Figura 10). La foto fu poi eliminata del sito quando divenne evidente che era stata manipolata, infatti era stato aumentato il fumo dovuto ai bombardamenti e reso più scuro con lo scopo di esagerare l'impatto dell'attacco israeliano. Reuters eliminò dal suo archivio oltre 1000 foto scattate da Hajj.



I casi esposti poc'anzi sono solo una piccola parte di tutte le foto che sono state manipolate e diffuse nel corso della storia. In tutti i casi, l'obiettivo è diffondere una notizia falsa o gettare infamia su un evento, una persona o una nazione. Diversi studi hanno dimostrato che le immagini manipolate restano impresse nella mente per molto tempo, infatti sono anche capaci di creare o alterare i ricordi di coloro che le guardano.

Pochi giorni prima delle elezioni presidenziali americane del 2004, fu chiesto ad un votante per chi avesse votato. Tra le varie ragioni che espose per giustificare il suo voto per George W. Bush, disse che non riusciva a togliersi dalla mente l'immagine di John Kerry e Jane Fonda insieme al congresso anti-guerra. Quando gli fu ricordato che l'immagine in realtà era un falso, il votante rispose «Lo so, ma non riesco a togliere quell'immagine dalla mia mente».

In uno studio, ai partecipanti furono fatte vedere delle immagini, sia autentiche che false, di memorabili eventi pubblici ai quali erano presenti. Le immagini manipolate, che mostravano ad esempio una folla più estesa o una maggiore violenza, riuscirono a cambiare il modo in cui i partecipanti ricordavano quegli eventi.



*La vita è breve e la conoscenza infinita.
Riassumere è dunque un male necessario.
Il compito di colui che riassume è fare bene
tale lavoro, che sebbene sia in sé pessimo,
è pur sempre meglio che niente.*

ALDOUS HUXLEY

Capitolo 1

RILEVARE LE CONTRAFFAZIONI MEDIANTE IL PRNU

1.1 PROCESSO DI ACQUISIZIONE DIGITALE

Il cuore di ogni fotocamera digitale è il sensore preposto alla cattura delle immagini. Esso consiste in un gran numero di piccoli rilevatori di luce, detti *foto-ricettori*, realizzati in silicio tramite la tecnologia CCD, JFET oppure CMOS. I ricettori catturano la luce grazie all'effetto foto-elettrico, accumulando gli elettroni che sono liberati per mezzo dei fotoni incidenti. La carica raccolta è trasferita fuori dal sensore, amplificata e quindi convertita in un segnale digitale. Questo è infine codificato in un formato binario grezzo che esprime la luminosità della scena nelle diverse posizioni spaziali.

L'immagine grezza è sottoposta a varie operazioni di *enhancement*, per compensare i difetti che inevitabilmente vengono introdotti durante la fase di cattura. Le elaborazioni tipicamente eseguite sono: riduzione del rumore, interpolazione dei colori (per costruire le bande RGB dell'immagine), bilanciamento del bianco (per compensare le sorgenti luminose che non emettono una luce perfettamente bianca), correzione gamma (per ottenere una migliore visualizzazione sui monitor) e altri tipi di filtraggi che migliorano la qualità visiva dell'immagine.

Il passo conclusivo prevede il salvataggio dell'immagine nella memoria interna della fotocamera. Questo potrebbe richiedere un'ulteriore elaborazione, tipicamente una compressione nel formato JPEG.

1.1.1 CATTURA DI UN'IMMAGINE

La cattura di un'immagine è il processo che permette di convertire la luce in un segnale bi-dimensionale che misura l'intensità luminosa in ogni posizione spaziale. Questa operazione è fisicamente realizzata dal *sensore d'immagini*, un reticolo $N \times M$ di foto-ricettori impiantati sul wafer di silicio.

Un *foto-ricettore* è un dispositivo semi-conduttore che accumula una carica elettrica proporzionale all'energia della luce incidente, secondo la relazione integrale espressa di seguito [12]:

$$I = T \int_{\Lambda} B(\lambda) \cdot \eta(\lambda) d\lambda \quad (1.1)$$

dove T è il tempo di esposizione, B è l'energia della luce incidente, η è la densità di carica per unità di energia e Λ è lo spettro del visibile. Il parametro η dipende dalla quantità di fotoni che il silicio è in grado di assorbire [19]. La Figura 1.1-a mostra la sensibilità del silicio alle diverse lunghezze d'onda. Il grafico evidenzia che un foto-ricettore presenta un comportamento di tipo monocromatico. La Figura 1.1-b, invece, mostra la capacità di penetrazione dei fotoni nel silicio, evidenziando che una radiazione incidente è in grado di arrivare tanto più in profondità quanto più è grande la sua lunghezza d'onda.

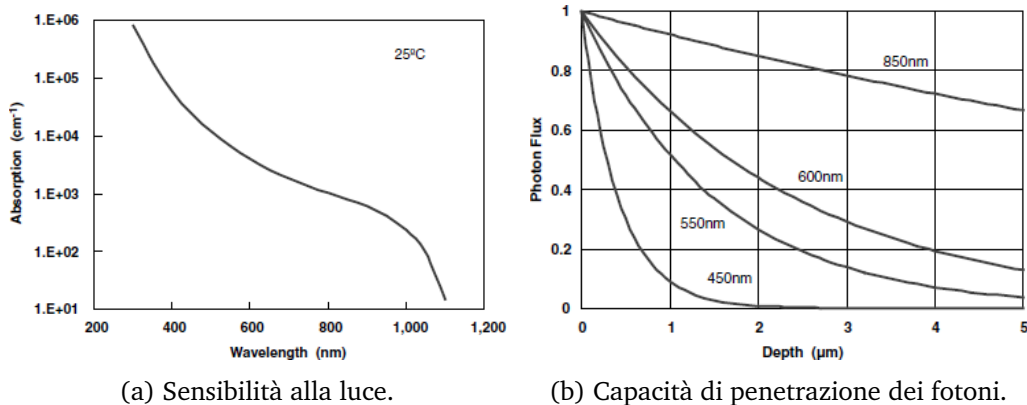


Figura 1.1 Proprietà fisiche del silicio.



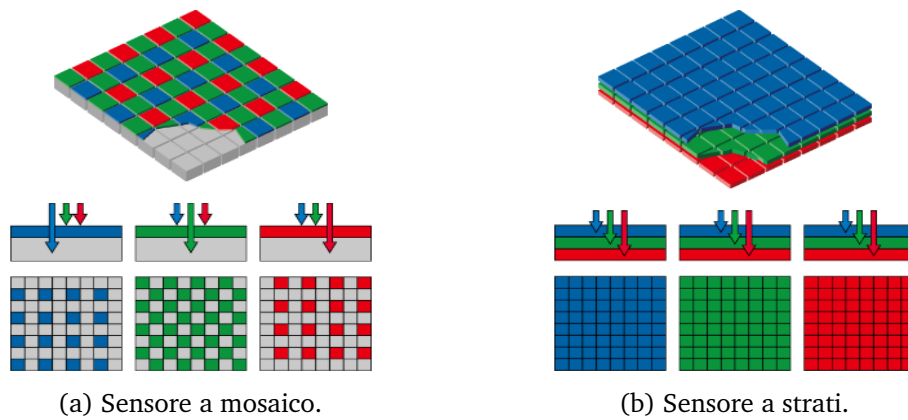


Figura 1.2 *Cattura dei colori.*

Le suddette proprietà del silicio rendono possibili due diversi tipi di sensori:

1. **cattura a mosaico.** I foto-ricettori sono impiantati sulla superficie del silicio secondo una struttura puramente bi-dimensionale. Per separare le bande di colore, il reticolo foto-sensibile viene ricoperto con uno strato di materiale in grado di filtrare selettivamente le componenti spettrali nella banda del visibile. Tale maschera è suddivisa in modo che ogni blocco 2x2 di ricettori sia esposto alla cattura di un pixel rosso, 2 pixel verdi e un pixel blu. La formazione di un'immagine RGB completa richiede una successiva operazione di *demosaicking*, che consiste nel combinare le uscite dei ricettori adiacenti e ricostruire le componenti di colore blu, verde e rosso in ogni posizione spaziale;
2. **cattura a strati.** I foto-ricettori sono impiantati nel silicio a tre diverse profondità, secondo una struttura tri-dimensionale. Per separare le bande di colore, si sfrutta la diversa capacità di penetrazione dei fotoni. Secondo questo principio, lo strato superiore cattura tutte le componenti spettrali del visibile. Lo strato intermedio, invece, cattura le componenti di colore verde e rosso, ma non il blu, che viene filtrato dallo stesso silicio. Analogamente, lo strato inferiore è esposto alla sola componente di colore rosso. Combinando linearmente le uscite dei tre strati di ricettori, è possibile ricavare le componenti di colore blu, verde e rosso in ogni posizione spaziale.



1.1.2 MODELLO DI RUMORE

La cattura di un'immagine inizia quando la luce esterna colpisce il sistema ottico. Quest'ultimo è composto da una lente esterna anti-riflesso, che filtra i raggi infra-rossi, e una matrice di micro-lenti posizionate sul reticolo fotosensibile (il sensore) per guidare le onde visibili verso i singoli ricettori. Queste lenti producono distorsioni geometriche e aberrazioni cromatiche, ma questi disturbi sono ignorati nella costruzione del modello di rumore.

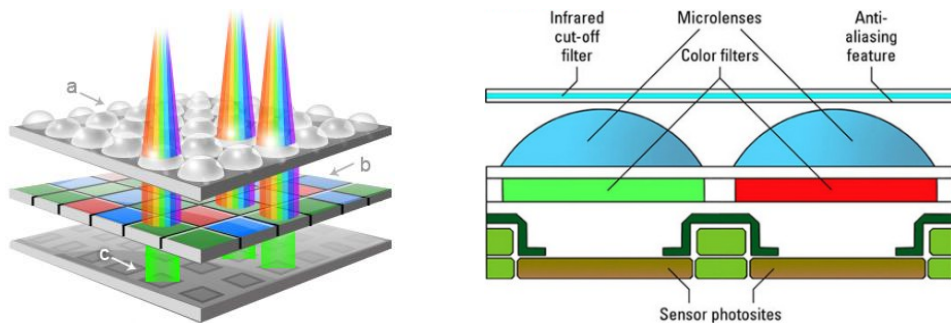


Figura 1.3 Funzionamento del sistema ottico.

La luce incidente attraversa il sistema ottico e raggiunge il *Color Filter Array*, un filtro che espone i diversi ricettori alla cattura di un singolo colore. Esso è impiegato solo nei sensori a mosaico, per consentire la separazione dei colori. Sebbene il CFA non introduca alcun disturbo, la cattura a mosaico richiede una successiva operazione di interpolazione dei colori, detta *demosaicking*. Essa è l'origine del cosiddetto *pattern di moiré*, un fenomeno di *aliasing* spaziale che si manifesta quando si tenta di ricostruire una struttura periodica (una texture presente nell'immagine) utilizzando un reticolo periodico (la griglia di foto-ricettori) caratterizzato da una diversa frequenza spaziale o un differente angolo. Il pattern di moiré è ridotto dal sistema ottico mediante un'operazione di *blurring* su quattro pixel contigui, al costo di rendere più *morbide* le immagini. Tutti questi disturbi non sono considerati nella costruzione del modello di rumore.

I raggi luminosi, una volta filtrati dal CFA, raggiungono il reticolo fotosensibile e colpiscono i singoli ricettori. Ogni foto-ricettore genera una corrente proporzionale al numero di fotoni incidenti [12], secondo una



legge di accumulazione della carica approssimabile con l'equazione 1.1. La risposta ideale del ricettore è alterata da diversi disturbi che danno luogo a una risposta rumorosa esprimibile tramite l'equazione:

$$I_N = (1 + K) \cdot I + N_{DC} + N_I \quad (1.2)$$

dove K , N_{DC} e N_I sono variabili aleatorie che modellano, rispettivamente, le seguenti componenti di rumore [12]:

1. **photo response non-uniformity.** Il processo di fabbricazione è soggetto a piccoli errori di imprecisione durante la costruzione del sensore. Questi errori conducono a piccole variazioni fisiche che alterano leggermente la risposta dei singoli ricettori. Pertanto, se il reticolo foto-sensibile è illuminato in modo uniforme, le cariche accumulate saranno leggermente diverse tra un ricettore e l'altro. Questa non uniformità di risposta è appunto chiamata *PRNU*. La variazione di carica rispetto al valore nominale I è modellata attraverso una v.a. K a media nulla e varianza σ_K^2 (uguale per tutti i ricettori);
2. **corrente al buio.** L'energia termica del silicio genera elettroni liberi che, mescolandosi con gli elettroni generati dalla luce incidente, accrescono la carica accumulata nei singoli foto-ricettori. Questo flusso di elettroni liberi è chiamato *corrente al buio*. Il nome è giustificato dal fatto che, in condizioni di buio totale, un ricettore accumula solo gli elettroni generati dall'eccitazione termica;
3. **rumore fotonico.** La natura quantizzata della luce introduce un certo grado di incertezza sulla carica accumulata nei singoli foto-ricettori. Il numero di elettroni effettivamente raccolti segue una distribuzione di Poisson. Il *rumore fotonico* rappresenta proprio questa incertezza e si modella come una v.a. a media nulla e varianza uguale al numero ideale di elettroni che un ricettore dovrebbe raccogliere (compresa la corrente al buio). Questo disturbo è una limitazione molto importante, poiché il rumore complessivo diventa *segnale-dipendente*.



Per generare l'immagine in formato grezzo, l'uscita del ricevitore viene prima convertita in tensione dall'amplificatore d'uscita e poi quantizzata dal convertitore analogico-digitale. Queste operazioni introducono degli ulteriori disturbi [12], esprimibili tramite l'equazione:

$$D = A \cdot I_N + N_A + N_Q \quad (1.3)$$

dove A è il guadagno dell'amplificatore, mentre N_A e N_Q sono v.a. a media nulla (indipendenti da I_N) che racchiudono gli effetti del rumore di amplificazione e quantizzazione, rispettivamente.

Le equazioni da 1.1 a 1.3 forniscono una caratterizzazione completa dell'uscita reale del foto-ricevitore. D'altra parte, il PRNU è un disturbo immutabile nel tempo, poiché dipende fortemente dalle caratteristiche fisiche del ricevitore. Questo permette di considerare il termine K come una grandezza deterministica, e quindi il modello di rumore del ricevitore può essere riformulato nel seguente modo:

$$D = \underbrace{I_A + K \cdot I_A + E_{DC}}_{\mu} + \underbrace{N_S + N_T + N_A + N_Q}_{N_G} \quad (1.4)$$

dove I_A è il valore amplificato della luminosità, $E_{DC} = A \cdot E\{N_{DC}\}$ è il valore medio della corrente al buio (amplificata), $N_T = A \cdot N_{DC} - E_{DC}$ è il rumore termico, mentre $N_S = A \cdot N_I$ è il rumore fotonico amplificato.

Il modello afferma che il pixel osservato D può essere suddiviso in una componente deterministica μ , linearmente dipendente dalla luminosità della scena, e una componente aleatoria N a media nulla che esprime gli effetti del rumore. Quest'ultima è la somma di una v.a. N_S di Poisson e una combinazione di disturbi N_G assimilabili ad una v.a. gaussiana [6]. La varianza totale del rumore additivo è $\sigma_N^2 = \sigma_S^2 + \sigma_G^2$, poichè N_S e N_G sono indipendenti tra loro (si noti che solo N_S dipende da μ). Le singole varianze, invece, si possono esprimere come:

$$\sigma_S^2 = A^2 \cdot \sigma_I^2 \quad \sigma_I^2 = (1 + K) I + E\{N_{DC}\} \quad \sigma_G^2 = \sigma_T^2 + \sigma_A^2 + \frac{q^2}{12} \quad (1.5)$$



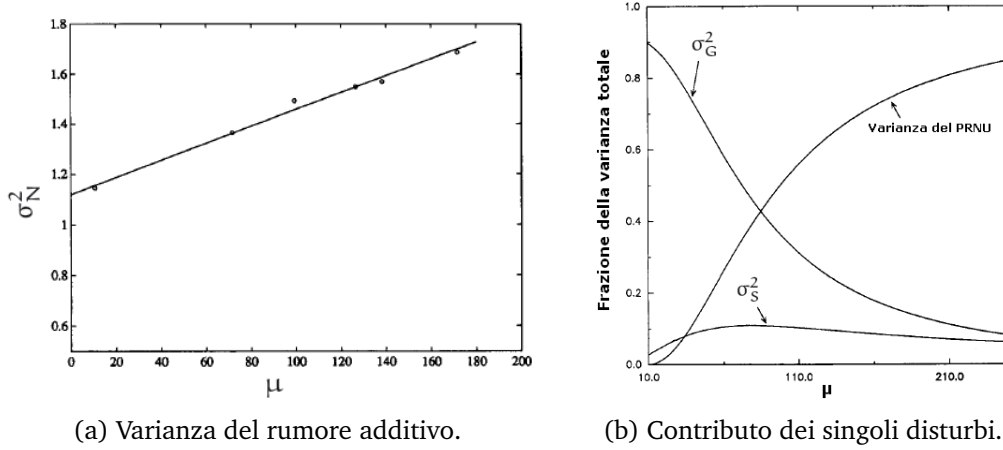


Figura 1.4 Proprietà delle componenti di rumore.

Le equazioni 1.5 dimostrano che il rumore additivo N è un disturbo *segnale dipendente*, come si vede anche nella Figura 1.4-a. Inoltre, la Figura 1.4-b mostra che il rumore complessivo è dominato dal termine $K \cdot I$ nelle regioni di forte luminosità ($I \gg 0$), con una varianza $\sigma^2 = I^2 \sigma_K^2$, mentre è dominata dal rumore gaussiano N_G nelle regioni di bassa luminosità [12].

I parametri del modello di rumore possono essere studiati con una procedura di *calibrazione* [12]. Sotto l'ipotesi che il pixel osservato D sia linearmente dipendente dalla luminosità I_A , la calibrazione fornisce una stima attendibile della corrente al buio e del PRNU. Queste possono essere impiegate per realizzare il cosiddetto *flat-fielding*, cioè la rimozione delle componenti deterministiche del rumore:

$$D_C = \frac{D - \hat{E}_{DC}}{1 + \hat{K}} \approx I_A + \frac{N_S + N_G}{1 + \hat{K}} \quad (1.6)$$

Il vantaggio del flat-fielding è che il PRNU viene soppresso. La nuova immagine D_C risulta dominata dal rumore fotonico nelle regioni di forte luminosità, invece che dal termine $K \cdot I$, mentre il rumore gaussiano continua a dominare le regioni di bassa luminosità. Tuttavia, questa operazione può essere realizzata correttamente solo sulle immagini grezze, dove il pixel osservato D è linearmente dipendente dalla luminosità I_A .



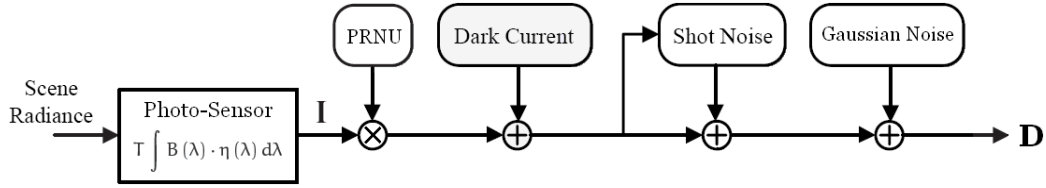


Figura 1.5 Modello lineare di rumore.

Il flat-fielding viene solitamente realizzato sulle immagini astronomiche, dove il PRNU è molto accentuato a causa dell'elevato tempo di esposizione. Le fotocamere di consumo, invece, si limitano a rimuovere la corrente al buio [17], che può essere stimata semplicemente catturando un'immagine in condizioni di buio totale. Il modello finale è dunque il seguente:

$$D_C = D - \hat{E}_{DC} = \underbrace{(1 + K) \cdot I_A}_{I_K} + \underbrace{N_S + N_G}_N \quad \sigma_N^2 = a \cdot I_A + b \quad (1.7)$$

dove $a, b \in \mathbb{R}$ esprimono il grado di dipendenza tra segnale e rumore.

Il processo di acquisizione termina con la fase di *post-processing*. Questa prevede innanzitutto la rimozione del *dark-frame*. Successivamente, l'immagine grezza viene sottoposta all'operazione di demosaicking, per ricostruire le bande RGB dell'immagine. Ai fini del modello di rumore, l'effetto di tale operazione è quello di correlare il rumore, a causa dell'interpolazione eseguita tra i pixel adiacenti. A questo punto, l'immagine viene sottoposta alle operazioni di bilanciamento del colore e correzione gamma, le quali si possono riassumere tramite una funzione non lineare detta *camera response function* [11, 16]. Infine, l'immagine viene sottoposta a compressione.

Il modello generale tiene conto anche delle non-linearità della fotocamera. Applicando lo sviluppo in serie di Taylor, l'uscita del foto-ricettore si può esprimere in termini della funzione CRF:

$$V = f_{CR}(I_K + N) + N_Q \approx f_{CR}(I_K) + \underbrace{f'_{CR}(I_K) \cdot N}_W + N_Q \quad (1.8)$$

e, in tal caso, la varianza del rumore diventa $\sigma_W^2 = [f'_{CR}(I_K)]^2 \cdot \sigma_N^2 + \sigma_Q^2$.



1.2 IMPRONTA DIGITALE DELLA FOTOCAMERA

Il sensore d'immagini è un reticolo di $m \times n$ foto-ricettori che catturano i fotoni incidenti e li convertono in elettroni. La quantità di elettroni raccolti, a parità di luce, è proporzionale alle dimensioni fisiche del ricettore. Il sensore è progettato in modo che tutti i ricettori abbiano le stesse dimensioni. Tuttavia, i processi di fabbricazione sono soggetti a piccoli errori di imprecisione, i quali conducono a leggere variazioni che alterano la dimensione dei ricettori, modificando la loro sensibilità alla luce [7].

L'imprecisione dei processi produttivi può essere rappresentata con una matrice K di $m \times n$ elementi, che esprime le variazioni fisiche dei singoli ricettori rispetto alla dimensione ideale. Quando il sensore è illuminato in modo uniforme con una intensità X , i diversi foto-ricettori non registrano una luminosità costante X , ma rispondono secondo una relazione del tipo:

$$Y[i] = X[i] + K[i] \cdot X[i] + N[i] \quad i \in \mathbb{Z}^2$$

Il termine $K \cdot X$ è il cosiddetto *Photo Response Non-Uniformity*, ovvero la “non uniformità di risposta” causata dalle piccole alterazioni fisiche nei ricettori. Il termine N , invece, è un rumore di tipo gaussiano-poissoniano che racchiude i numerosi disturbi introdotti durante il processo di acquisizione.

Il PRNU è un disturbo di tipo moltiplicativo. Questo rende il termine $K \cdot X$ molto debole nelle aree scure dell'immagine, dove $X \approx 0 \Rightarrow K X \approx 0$. Inoltre, nelle zone totalmente sature ($X \approx 255$), i ricettori lavorano al massimo della loro capacità, producendo un segnale costante che non conserva alcuna traccia del PRNU o di qualsiasi altro tipo di rumore.

La matrice K è la principale responsabile dell'*impronta digitale della fotocamera*, una grandezza che racchiude tutti i difetti sistematici del sensore. Un ulteriore contributo arriva dai *pixel guasti* [7], cioè i ricettori che producono costantemente il valore massimo o minimo di luminosità. L'impronta racchiude anche i *disturbi a bassa frequenza* [17], come la luce rifratta sulle particelle di polvere e i difetti delle lenti. Se la fotocamera non effettua la sottrazione del *dark frame*, entra in gioco pure la corrente al buio [7].



Il PRNU è una componente relativamente stabile del sensore, nonostante la sua natura aleatoria. Questo rende la matrice K un utile strumento forense, essendo una grandezza che gode delle seguenti proprietà [7]:

1. **unicità.** Il carattere e l'origine del PRNU rendono improbabile che le impronte digitali di sensori differenti siano correlate tra loro, anche se provengono dallo stesso wafer di silicio [17];
2. **universalità.** Tutti i sensori esibiscono il PRNU, poiché i processi di fabbricazione basati su silicio sono molto simili tra loro, indipendentemente dalla tecnologia utilizzata (CCD, JFET o CMOS);
3. **generalità.** Il PRNU è presente in ogni immagine, indipendentemente dall'ottica, i settaggi della fotocamera e il contenuto della scena (ad eccezione delle aree scure e le zone completamente sature);
4. **stabilità.** Il PRNU è stabile nel tempo e sotto un'ampia gamma di condizioni ambientali, essendo legato solo a caratteristiche fisiche (e quindi immutabili) del sensore;
5. **robustezza.** Il PRNU sopravvive a molte elaborazioni digitali (compressione JPEG, filtraggi, etc. . .), grazie alla sua natura moltiplicativa.

L'impronta digitale può essere usata in diverse attività forensi [3]. L'*identificazione* è il processo che permette di risalire alla particolare fotocamera che ha catturato una certa immagine. A tale scopo, basta verificare che l'impronta della fotocamera sia *presente* nell'immagine sotto analisi.

Viceversa, il *collegamento* è il processo che permette di stabilire se due immagini sono state catturate dalla stessa fotocamera. Questo richiede di *confrontare* le impronte nelle due immagini e stabilire se sono uguali.

Infine, la *verifica di integrità* è il processo che permette di scoprire se una certa immagine è stata manipolata. Per realizzare questo compito, è necessario verificare che l'impronta della fotocamera origine sia *assente* in alcune parti dell'immagine sotto analisi, sintomo di qualche contraffazione.



1.2.1 STIMA DEL PRNU

L'impronta digitale del sensore può essere stimata sperimentalmente senza troppe difficoltà. L'unico requisito è disporre di numerose immagini scattate dalla fotocamera o, meglio ancora, possedere direttamente la fotocamera. Sotto queste ipotesi, infatti, qualunque tecnica di stima ragionevole è capace di fornire un buon risultato, grazie alla natura deterministica del PRNU.

Un modo semplice per stimare l'impronta digitale consiste nel calcolare la media su tante immagini di una superficie uniformemente illuminata. Tale operazione permette di isolare le componenti deterministiche (PRNU, difetti sistematici del sensore) e sopprimere le componenti aleatorie (contenuto della scena, rumore additivo). Date le immagini $Y_1, Y_2, \dots, Y_M$, risulta:

$$\hat{K} = \frac{1}{M} \sum_{m=1}^M (Y_m - E\{Y_m\})$$

dove $E\{Y_m\}$ è l'intensità media dell'immagine Y_m .

Il contenuto della scena è la componente più forte dell'immagine, di gran lunga superiore al PRNU, che a confronto diventa un termine trascurabile. Affinché il PRNU possa emergere con chiarezza, dunque, il contenuto della scena deve essere soppresso. L'operazione di media aritmetica è un modo semplice, ma piuttosto inefficiente, per raggiungere questo scopo. Infatti, il numero di immagini richieste per sopprimere il segnale inutile (e rivelare così il PRNU) può diventare eccessivo, soprattutto quando non si possiede la fotocamera e le immagini disponibili sono in numero limitato.

Un modo efficace per sopprimere il contenuto della scena è utilizzare un filtro di *denoising*. L'idea consiste nel prendere l'immagine originale Y e sottrarre da essa l'immagine filtrata $D(Y) = \hat{X}$. Il risultato è il cosiddetto *residuo di rumore*, un segnale che contiene i disturbi $KX + N$ e gli artefatti introdotti dal denoising, ma non il contenuto della scena X :

$$W = Y - D(Y) = KX + N' \quad D(Y) = \hat{X}$$

dove il termine $N' = N + (X - \hat{X})$ ingloba anche le imperfezioni del denoising.



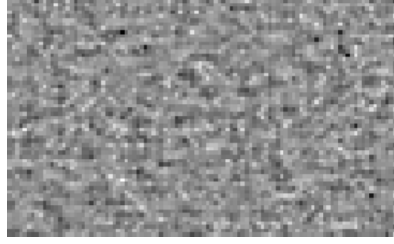


Figura 1.6 *Impronta digitale di una fotocamera.*

Il PRNU emerge con chiarezza nel residuo di rumore. Assumendo che il rumore N' sia trascurabile e la scena X sia illuminata in modo uniforme, il residuo $W = KX + N'$ è un'ottima stima della matrice K . Tuttavia, queste sono ipotesi molto favorevoli. Nei casi più realistici, il residuo è influenzato da diversi fattori (artefatti del denoising, cattiva luminosità della scena, disturbi del sensore) che tendono ad offuscare il PRNU. Una stima affidabile si ottiene facendo la media di un certo numero di residui, in modo da isolare le componenti deterministiche (PRNU, difetti sistematici) e sopprimere le componenti aleatorie (rumore additivo, artefatti del denoising):

$$\hat{K} = \frac{1}{M} \sum_{m=1}^M W_m$$

Un metodo più accurato per stimare il PRNU è basato sul concetto di *massima verosimiglianza* [3, 13]. L'idea consiste nel massimizzare la funzione di log-verosimiglianza dell'osservazione W_m/Y_m dato K . Nel caso di rumore AWGN, ciò equivale a minimizzare l'errore quadratico medio:

$$\hat{K} = \underset{K}{\operatorname{argmin}} \sum_{m=1}^M (W_m - Y_m K)^2 = \frac{1}{\alpha} \sum_{m=1}^M W_m \cdot Y_m \quad \alpha = \sum_{m=1}^M (Y_m)^2$$

La stima ML è una media pesata dei residui di rumore. Il termine Y_m riduce il contributo di W_m nelle zone scure dell'immagine, dove il PRNU è più attenuato e quindi poco utile per effettuare una stima affidabile.

La Figura 1.6 mostra una porzione ingrandita dell'impronta di una fotocamera, calcolata su 120 immagini in scala di grigi. Da notare la diversa luminosità dei pixel, causata dalla “non uniformità di risposta” dei ricettori.



1.2.2 PRE-ELABORAZIONE DEL PRNU

La stima $\hat{K}$ dell'impronta digitale include tutte le componenti che sono sistematicamente presenti in ogni immagine. Oltre ai difetti del sensore, di cui il PRNU è la grandezza dominante, sopravvivono anche gli artefatti causati dal demosaicking, dalla compressione JPEG e da alcune operazioni interne del sensore. Questi disturbi sono condivisi da tutte le fotocamere della stessa linea produttiva e, quindi, introducono una leggera correlazione tra le loro impronte, riducendo l'affidabilità dell'analisi forense [3].

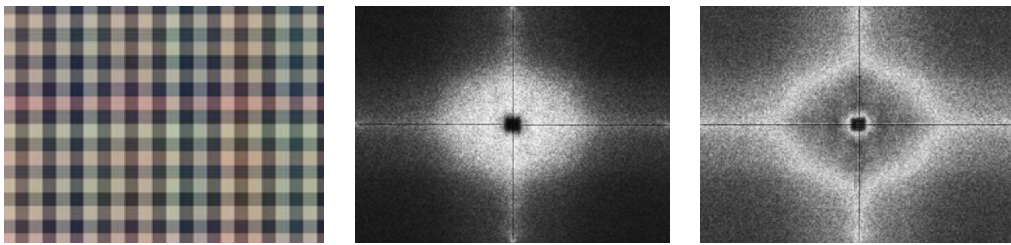
Gli artefatti introdotti dal demosaicking e dal sensore si manifestano come un pattern periodico nelle intensità medie delle righe e colonne di $\hat{K}$. Tale disturbo può essere soppresso tramite un'operazione di *zero mean* [3], che consiste nell'azzerare il valor medio delle singole righe e colonne di $\hat{K}$:

$$\hat{K}_{ZM}(i, j) = \hat{K}(i, j) - \frac{1}{M} \sum_{i=1}^M \hat{K}(i, j) - \frac{1}{N} \sum_{j=1}^N \hat{K}(i, j) + \frac{1}{N \cdot M} \sum_{i=1}^M \sum_{j=1}^N \hat{K}(i, j)$$

Viceversa, gli artefatti introdotti dalla compressione JPEG si manifestano tramite l'effetto *blocchizzazione*. Se il termine $\hat{K}_{ZM}$ presenta tali disturbi nella forma di pattern identificabili visivamente, esso viene elaborato nel dominio di Fourier mediante un filtraggio di Wiener [3]:

$$\hat{K}_{WF} = \mathfrak{F}^{-1} \left\{ F_K - W[F_K] \right\} \quad F_K = \mathfrak{F} \left\{ \hat{K}_{ZM} \right\}$$

dove $W[\cdot]$ è il filtro di Wiener 3x3 con varianza calcolata su $|F_K|$.



(a) Medie di riga e colonna. (b) Spettro senza Wiener. (c) Spettro dopo Wiener.

Figura 1.7 Artefatti presenti nell'impronta digitale.



1.3 VERIFICA DI INTEGRITÀ

Le fotocamere digitali aggiungono involontariamente un lieve disturbo nelle immagini catturate, detto *Photo Response Non Uniformity*. Il PRNU può essere considerato una sorta di impronta digitale, essendo unico per ogni fotocamera, costante nel tempo e indipendente dalla scena. Tra le altre cose, esso permette di scoprire se un'immagine è stata manipolata. L'idea consiste nell'individuare le regioni in cui è *assente* il PRNU della fotocamera che ha catturato l'immagine. Infatti, quando si manipola una regione, inevitabilmente si cancella anche il PRNU in essa contenuta.

Data un'immagine $Y = X + K \cdot X + N$, la *verifica d'integrità* consiste nell'accertare la presenza/assenza del PRNU nel residuo di rumore [3]:

$$W = Y - D(Y) = K \cdot X + N' \quad N' = N + (X - \hat{X})$$

dove N' ingloba sia il rumore AWGN che gli artefatti del denoising $D(Y) = \hat{X}$. Per ogni pixel di W , si formula un *test d'ipotesi* sul blocco $B \times B$ che circonda il pixel sott'analisi. L'esito del test è attribuito solo al pixel centrale. Il risultato globale si ottiene aggregando le decisioni locali di tutti i test.

L'aspetto più critico della verifica d'integrità è l'estrazione del PRNU dall'immagine Y . Infatti, se la scena X fosse illuminata in modo uniforme, il residuo W fornirebbe un'ottima stima del PRNU. Naturalmente, non è questo il caso. Anche nell'ipotesi di realizzare un denoising perfetto, il residuo di rumore è influenzato dal contenuto della scena (a causa della natura moltiplicativa del PRNU), e quindi solo debolmente correlato con l'impronta della fotocamera. Nel caso più generale, insomma, il massimo che si può sperare è un certo grado di correlazione tra i termini W e $\hat{K}$.

L'unico modo per affrontare questo problema è incrementare le dimensioni del blocco su cui si realizza il test d'ipotesi, ad esempio 128×128 . In questo modo, si migliora l'affidabilità del test, poiché aumentano le informazioni disponibili. Naturalmente, questo peggiora la risoluzione del risultato globale, poiché si riduce la probabilità di riconoscere le regioni contraffatte molto piccole, come quelle composte da meno di $B/2 \times B/2$ pixel.



1.3.1 TEST D'IPOTESI

Il *test d'ipotesi* è l'operazione che permette di stabilire se l'impronta della fotocamera è presente in un blocco $B \times B$ del residuo di rumore (preso dall'immagine sotto analisi). Dal punto di vista statistico, il blocco $\mathbf{w} = (w_1, \dots, w_{B^2})$ è modellato come la realizzazione di un vettore aleatorio $\mathbf{W}$. Non si conosce l'esatta distribuzione di $\mathbf{W}$, ma solo che questa è soggetta a due ipotesi, note a priori. L'ipotesi H_0 afferma che il ve.a. $\mathbf{W}$ è un segnale di puro rumore, cioè il PRNU è *assente* dal blocco $\mathbf{w}$. Viceversa, l'ipotesi H_1 sostiene che il ve.a. $\mathbf{W}$ è un segnale deterministico immerso nel rumore, ossia il PRNU è *presente* nel blocco $\mathbf{w}$. Sinteticamente, si può scrivere [3]:

$$\begin{array}{ll} H_0 : \mathbf{w} = \mathbf{n} & \Longleftrightarrow H_0 : \mathbf{W} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}) \\ H_1 : \mathbf{w} = \mathbf{k} + \mathbf{n} & \Longleftrightarrow H_1 : \mathbf{W} \sim N(\mathbf{k}, \sigma^2 \mathbf{I}) \end{array}$$

L'obiettivo del test è scegliere l'ipotesi che meglio descrive la distribuzione del ve.a. $\mathbf{W}$ basandosi sull'osservazione $\mathbf{w}$. A tale scopo, si fissa una soglia opportuna γ e si decide in base all'esito del *test statistico* [14]:

$$\begin{aligned} \frac{f(\mathbf{w}; H_1)}{f(\mathbf{w}; H_0)} &= \frac{\exp \left[-\frac{(\mathbf{w} - \mathbf{k})^T (\mathbf{w} - \mathbf{k})}{2\sigma^2} \right]}{\exp \left[-\frac{\mathbf{w}^T \mathbf{w}}{2\sigma^2} \right]} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \\ T(\mathbf{w}) &= \mathbf{w}^T \mathbf{k} \underset{H_0}{\overset{H_1}{\gtrless}} \underbrace{\frac{\|\mathbf{k}\|^2}{2} + \sigma^2 \log \gamma}_{\gamma'} \end{aligned}$$

La statistica $T(\mathbf{w})$ può essere considerata la realizzazione di una variabile aleatoria $T(\mathbf{W})$ con distribuzione soggetta a due ipotesi, note a priori. L'ipotesi H_0 afferma che $T(\mathbf{W})$ è concentrata attorno allo zero, mentre l'ipotesi H_1 sostiene che $T(\mathbf{W})$ è centrata in un valore non nullo:

$$\begin{array}{ll} H_0 : T(\mathbf{w}) = \mathbf{k}^T \mathbf{n} & \Longleftrightarrow H_0 : T(\mathbf{W}) \sim N(0, \sigma_0) \\ H_1 : T(\mathbf{w}) = \|\mathbf{k}\|^2 + \mathbf{k}^T \mathbf{n} & \Longleftrightarrow H_1 : T(\mathbf{W}) \sim N(\|\mathbf{k}\|^2, \sigma_1) \end{array}$$



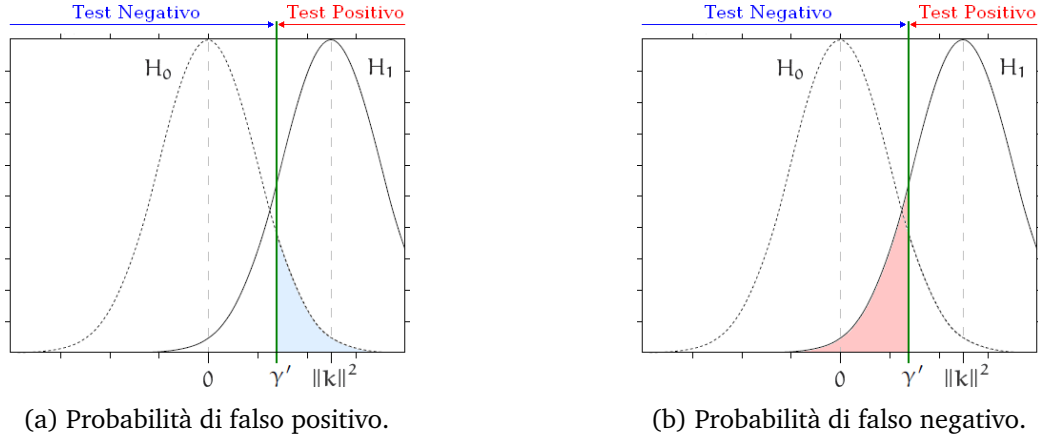


Figura 1.8 Test d'ipotesi.

Il test statistico $T(\mathbf{w}) \leq \gamma'$ suddivide lo spazio delle osservazioni in due regioni di decisione: nei punti $R_0 = \{\mathbf{w} : T(\mathbf{w}) < \gamma'\}$ si decide per H_0 , e nei punti $R_1 = \{\mathbf{w} : T(\mathbf{w}) > \gamma'\}$ si decide per H_1 . Il test può commettere due tipi di errori: si decide per H_1 quando H_0 è vera (*falso positivo*, si dichiara la presenza del PRNU quando in realtà è assente) o viceversa (*falso negativo*). Per determinare la soglia γ' , si fissa la probabilità di falso positivo a una costante arbitraria (criterio *Neyman-Pearson*). Siccome $T(\mathbf{W})$ è una v.a. gaussiana, il valore della soglia è il seguente [14]:

$$P_{FP} = \Pr\{T(\mathbf{W}) > \gamma'; H_0\} = Q\left(\frac{\gamma'}{\sigma_0}\right) \implies \gamma' = \sigma_0 \cdot Q^{-1}(P_{FP})$$

dove $Q(x)$ indica la *Q-function*. Ad esempio $\gamma' = 2.3 \sigma_0$ se $P_{FP} = 1\%$.

La statistica $T(\mathbf{w}) = \mathbf{w}^T \mathbf{k}$ dipende fortemente dal modulo di $\mathbf{w}$ e $\mathbf{k}$, essendo una misura basata sul prodotto scalare. Questo significa che il test d'ipotesi è sensibile alle variazioni di luminosità nelle immagini. Il problema si risolve normalizzando i vettori $\mathbf{w}$ e $\mathbf{k}$, dopo aver sottratto le rispettive medie. La nuova statistica è detta *coefficiente di correlazione* [3]:

$$\text{corr}(\mathbf{w}, \mathbf{k}) = \frac{(\mathbf{w} - \mu_w)^T (\mathbf{k} - \mu_k)}{\|\mathbf{w} - \mu_w\| \cdot \|\mathbf{k} - \mu_k\|}$$

dove μ_w e μ_k indicano la media dei vettori $\mathbf{w}$ e $\mathbf{k}$.



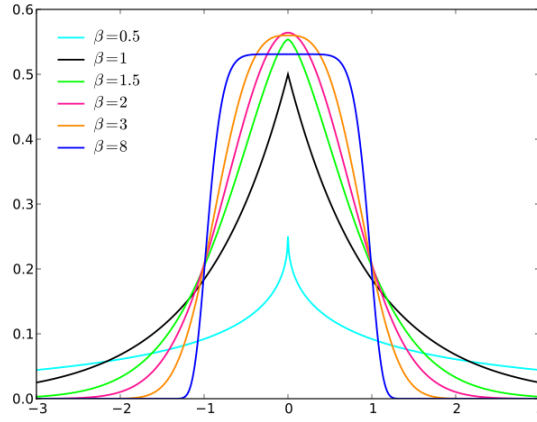


Figura 1.9 *Gaussiana generalizzata.*

La statistica $\text{corr}(\mathbf{w}, \mathbf{k})$ può essere considerata la realizzazione di una variabile aleatoria P con distribuzione soggetta a due ipotesi, note a priori. L'ipotesi H_0 afferma che P è concentrata attorno allo zero, cioè il residuo di rumore è incorrelato con il PRNU di riferimento. Viceversa, l'ipotesi H_1 sostiene che P è centrata in un valore non nullo, ossia il residuo di rumore è polarizzato dal PRNU di riferimento. Sinteticamente, si può scrivere:

$$\begin{aligned} H_0 : P &\sim \text{GG}(0, \alpha_0, \beta_0) \\ H_1 : P &\sim \text{GG}(\mu_1, \alpha_1, \beta_1) \end{aligned} \quad \text{corr}(\mathbf{w}, \mathbf{k}) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

Le distribuzioni delle ipotesi sono modellate attraverso la *gaussiana generalizzata*, una famiglia di pdf con tre gradi di libertà:

$$\text{GG}(\mu, \alpha, \beta) = \frac{\beta}{2 \alpha \Gamma\left(\frac{1}{\beta}\right)} \exp \left[- \left(\frac{|x - \mu|}{\alpha} \right)^\beta \right] \quad \sigma^2 = \alpha^2 \frac{\Gamma\left(\frac{3}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}$$

dove μ e σ^2 sono la media e la varianza della pdf. Il parametro β permette di controllare le code della distribuzione. Il valore di riferimento è $\beta = 2$, dove la pdf diventa gaussiana $\text{GG}(\mu, \alpha, 2) = \text{N}(\mu, \frac{\alpha^2}{2})$. Le distribuzioni con valori $\beta < 2$ hanno code più larghe, mentre quelle con valori $\beta > 2$ hanno code più strette. Infatti, la distribuzione diventa esponenziale quando $\beta = 1$, mentre tende alla distribuzione uniforme per $\beta \rightarrow \infty$ (vedi Figura 1.9).



La soglia λ non può essere calcolata senza conoscere la pdf dell'ipotesi H_0 . Questa può essere stimata sperimentalmente senza troppe difficoltà. Innanzitutto, si calcola la correlazione tra il PRNU di riferimento e il residuo di rumore estratto da numerose immagini catturate con altre fotocamere. I risultati, essendo realizzazioni della v.a. $H_0 : P \sim GG(0, \alpha_0, \beta_0)$, si utilizzano per stimare i parametri α_0 e β_0 , dai quali è possibile calcolare la soglia:

$$P_{FP} = \int_{\lambda}^{+\infty} GG(0, \alpha_0, \beta_0) dx \quad \Rightarrow \quad \lambda = 2 \cdot \sigma_0 = 2 \sqrt{\alpha_0^2 \cdot \frac{\Gamma(3/\beta_0)}{\Gamma(1/\beta_0)}}$$

La scelta $\lambda = 2 \cdot \sigma_0$ fissa la probabilità di falso positivo a $P_{FP} \approx 1\%$.

I parametri della distribuzione gaussiana generalizzata si possono stimare attraverso il *metodo dei momenti*. Supponendo di avere N realizzazioni di una v.a. $P \sim GG(\mu, \alpha, \beta)$, gli stimatori dei parametri μ , α e β sono [17]:

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^N p[n] \quad \hat{\beta} = \psi^{-1} \left(\frac{\hat{m}^2}{\hat{\sigma}^2} \right) \quad \hat{\alpha} = \hat{m} \cdot \frac{\Gamma(1/\hat{\beta})}{\Gamma(2/\hat{\beta})}$$

dove $\psi(x) = \frac{\Gamma^2(2/x)}{\Gamma(1/x) \cdot \Gamma(3/x)}$ ed i momenti di 1° e 2° ordine sono:

$$\hat{m} = \frac{1}{N} \sum_{n=1}^N |p[n] - \hat{\mu}| \quad \hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N (p[n] - \hat{\mu})^2$$

La funzione inversa $\psi^{-1}(y)$ non esiste in forma analitica. Una possibile soluzione consiste nell'approssimare $\psi(x)$ con una funzione invertibile:

$$\psi(x) = \begin{cases} \left(\sqrt{3}\right)^{\frac{x-6}{x}} \cdot 2^{\frac{4-x}{x}} & \text{se } x \in [0, I_1) \\ a_1 x^2 + a_2 x + a_3 & \text{se } x \in [I_1, I_2) \\ \frac{b_1 \cdot x}{1 + b_2 x + b_3 x^2} & \text{se } x \in [I_2, I_3) \\ \frac{3}{4} - c_1 \cdot \exp[-c_2 x + c_3 x^2] & \text{se } x \in [I_3, \infty) \end{cases}$$

Si rimanda all'articolo [10] per ulteriori approfondimenti.



1.3.2 RIDUZIONE DEI FALSI NEGATIVI

Il test d'ipotesi accerta la presenza del PRNU in un blocco $B \times B$ del residuo di rumore. La decisione è basata sulla correlazione tra il blocco sotto analisi e il rispettivo blocco del PRNU. Se la correlazione è maggiore di una certa soglia λ_0 , l'esito del test è *positivo*, cioè si dichiara la presenza del PRNU. La scelta della soglia λ_0 è guidata dal criterio Neyman-Pearson. Questo permette di fissare la probabilità dei falsi positivi a un valore arbitrario.

Una volta stabilita la soglia λ_0 , la probabilità dei falsi negativi è fissata al valore $P_{FN} = \Pr\{\text{corr}(\mathbf{W}, \mathbf{K}) < \lambda_0; H_1\}$. In alcune circostanze, si desidera controllare anche questa probabilità. A tale scopo, si può introdurre una seconda soglia $\lambda_1 < \lambda_0$, come mostrato in Figura 1.10-b. Se la correlazione è minore di λ_1 , l'esito del test è *negativo*, cioè si dichiara l'assenza del PRNU. Naturalmente, esiste anche una terza possibilità. Se la correlazione è compresa tra le 2 soglie, l'esito del test è *indefinito* (non si prende alcuna decisione, per evitare di sbagliare). L'inconveniente di questo metodo è la scelta non facile della seconda soglia. Una soluzione consiste nell'usare il criterio Neyman-Pearson sull'ipotesi H_1 . L'alternativa è fissare λ_1 in modo tale che la decisione sia presa solo se: $\min\{C_{01} \cdot f(\rho; H_1), C_{10} \cdot f(\rho; H_0)\} < t$, dove C_{ij} è il costo per decidere H_i quando H_j è vera.

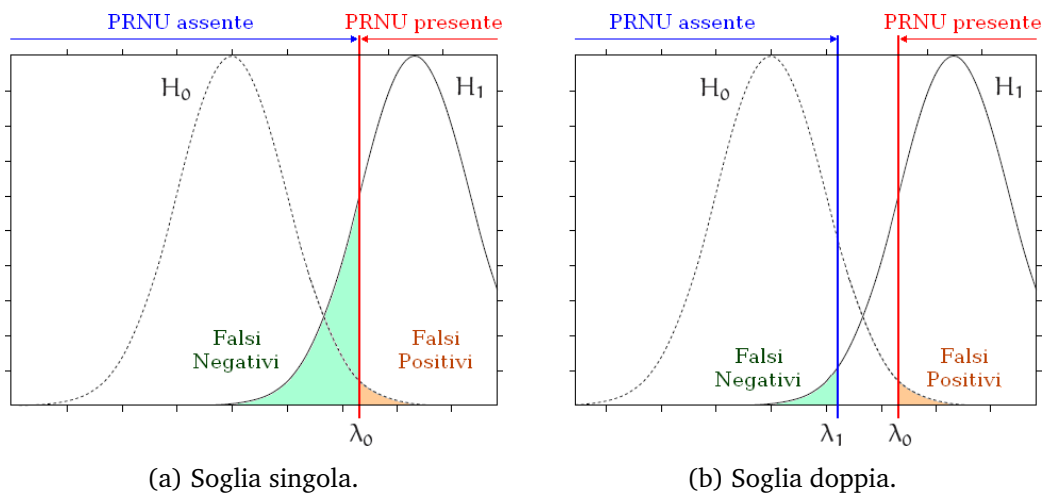


Figura 1.10 Test d'ipotesi.



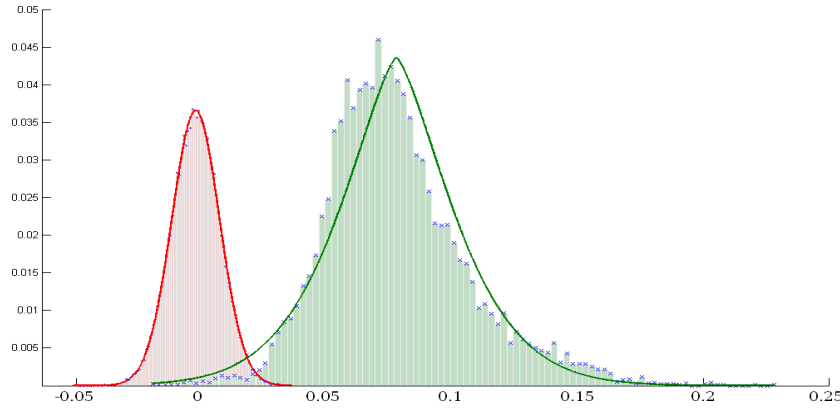


Figura 1.11 Stima delle ipotesi.

L'ipotesi H_0 può essere facilmente stimata calcolando le correlazioni tra il PRNU di riferimento e i residui di rumore presi da altre fotocamere. Per stimare l'ipotesi H_1 , invece, anche i residui di rumore devono essere presi dalla fotocamera di riferimento. La figura 1.11 mostra l'istogramma della statistica $\rho = \text{corr}(\mathbf{w}, \mathbf{k})$ nelle ipotesi H_0 (barre rosse) e H_1 (barre verdi). Come si può vedere, la distribuzione $f(\rho; H_0)$ si presta bene ad essere modellata come una gaussiana generalizzata (linea rossa), mentre la distribuzione $f(\rho; H_1)$ non si adatta al modello gaussiano (linea verde).

Il problema dell'ipotesi H_1 è che il contenuto della scena influenza fortemente il risultato della correlazione. I blocchi che presentano un basso valore della correlazione si trovano tipicamente: nelle zone scure (dove il PRNU è debole a causa della sua natura moltiplicativa), nelle zone piatte (dove il PRNU è debole a causa della saturazione dei colori o della forte compressione JPEG) e nelle zone strutturate (dove il filtro di denoising non rimuove bene il contenuto della scena). Questo rende molto difficile trovare un semplice modello statistico per $f(\rho; H_1)$. Infatti, la stima in Figura 1.11 è troppo conservativa nelle zone *favorevoli* (dove il PRNU è forte, e quindi ρ è alto) e troppo ottimistica nelle zone *sfavorevoli* (dove il PRNU è debole).

La soluzione è fornita dal cosiddetto *predittore*, una funzione $\hat{\rho}$ che stima la correlazione $\rho = \text{corr}(\mathbf{w}, \mathbf{k})$ del blocco $\mathbf{w}$ basandosi su tre attributi dell'immagine $\mathbf{y} = \mathbf{x} + \mathbf{w}$: intensità, struttura e uniformità. L'idea consiste nel modellare l'errore di predizione $v = \rho - \hat{\rho}$ come una v.a. $GG(0, \alpha_v, \beta_v)$.



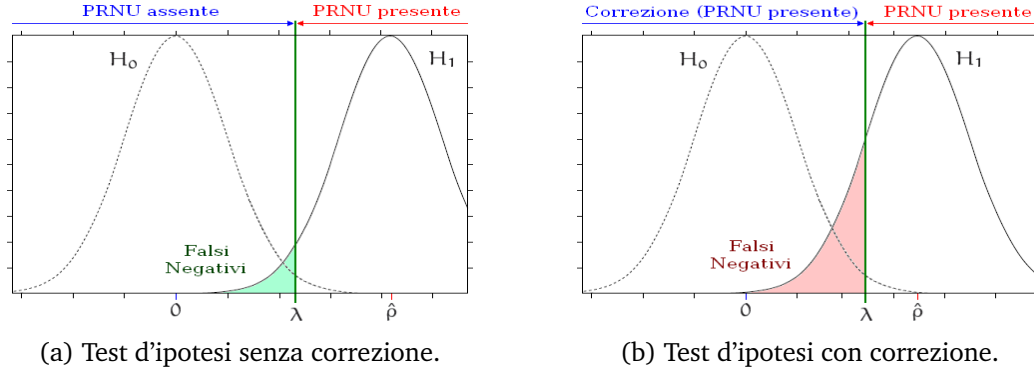


Figura 1.12 Modello adattativo dell'ipotesi H_1 .

La Figura 1.13-a mostra l'istogramma dell'errore v e il relativo modello gaussiano (linea rossa), mentre la Figura 1.13-b riporta le coppie $(\hat{\rho}, \rho)$.

Il predittore consente di adattare l'ipotesi H_1 al contenuto dell'immagine. Infatti, la distribuzione $f(\rho; H_1)$ può essere modellata come $GG(\hat{\rho}, \alpha_v, \beta_v)$, dove $\hat{\rho}$ è la predizione fatta con i soli attributi dell'immagine (non del residuo di rumore). L'idea è giustificata dal fatto che la varianza dell'errore di predizione non dipende dal contenuto dell'immagine (Figura 1.13-b).

Il test d'ipotesi fornisce una risposta negativa (PRNU *assente* dal pixel sotto analisi) quando $\rho < \lambda$. Tuttavia, nelle zone sfavorevoli dell'immagine, la correlazione ρ assume un valore basso anche sotto l'ipotesi H_1 . Per raggiungere una decisione conservativa ed evitare di etichettare un pixel autentico come contraffatto, si calcola la probabilità di falso negativo:

$$P_{FN} = \int_{-\infty}^{\lambda} GG(\hat{\rho}, \alpha_1, \beta_1) dx$$

Se tale probabilità supera una certa soglia γ , la decisione viene corretta e si fornisce una risposta positiva (PRNU *presente* nel pixel sotto analisi).

La correzione riduce l'occorrenza dei falsi negativi. Infatti, grazie al modello adattativo dell'ipotesi H_1 , il termine P_{FN} è piccolo nelle zone favorevoli dell'immagine (dove $\hat{\rho}$ è alto, vedi Figura 1.12-a), ma è grande nelle zone sfavorevoli (dove $\hat{\rho}$ è basso, vedi Figura 1.12-b). Questo permette di riconoscere le zone dove è più facile generare un falso negativo.



1.3.3 PREDITTORE

Il *predittore* è una funzione che stima la correlazione del residuo di rumore $W = Y - D(Y)$ usando tre attributi dell'immagine (calcolati su blocchi $B \times B$):

1. **intensità.** La correlazione diventa più forte all'aumentare della luminosità (natura moltiplicativa del PRNU), ma superata una soglia Y_{th} , essa inizia a calare fino ad annullarsi in $Y = 255$ (saturazione della dinamica). Una possibile misura dell'intensità è la seguente [3]:

$$x_i = \frac{1}{B^2} \sum_{j \in W_B} att(Y[j]) \quad att(Y) = \begin{cases} \exp \left[-\frac{(Y - Y_{th})^2}{\tau} \right] & Y > Y_{th} \\ \frac{Y}{Y_{th}} & Y \leq Y_{th} \end{cases}$$

Una scelta ragionevole dei parametri è $Y_{th} = 250$ e $\tau = 6$;

2. **struttura.** La correlazione tende ad essere più bassa nelle zone fortemente strutturate, a causa delle limitazioni del denoising. Questo attributo può essere calcolato sulle componenti ad alta frequenza, cioè la trasformata inversa delle 6 bande di dettaglio nei primi due livelli di decomposizione wavelet. Una possibile misura è la seguente [3]:

$$x_t = \frac{1}{B^2} \sum_{j \in W_B} \frac{1}{1 + var_5(F[j])}$$

dove F è l'immagine passa-alto, mentre $var_5(F[j])$ indica la varianza campione del blocco 5×5 centrato nel pixel $F[j]$;

3. **uniformità.** Le operazioni a bassa frequenza (come la compressione JPEG) si concentrano soprattutto nelle regioni piatte, causando una maggiore attenuazione del PRNU e, quindi, un calo della correlazione. Una possibile misura dell'uniformità è la seguente [3]:

$$x_u = \frac{1}{B^2} \left| \{ j \in W_B : \sigma[j] < c \cdot Y[j] \} \right|$$

dove $c = 0.03$, e $\sigma[j]$ è la varianza del blocco 5×5 centrato in $Y[j]$.



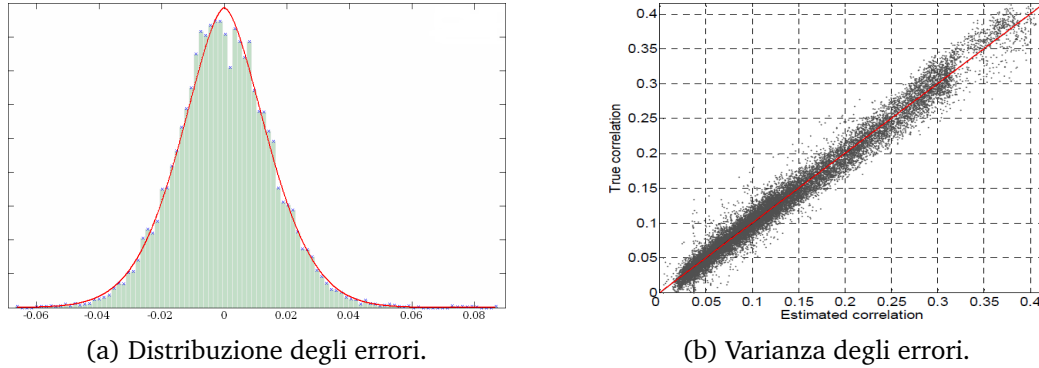


Figura 1.13 Errore di predizione.

Una volta definiti gli attributi, il predittore può essere costruito usando una qualsiasi tecnica di *machine learning*. L'idea consiste nell'*apprendere* la relazione funzionale tra gli attributi e la correlazione utilizzando un insieme di esempi significativi. Un *esempio* è una coppia $(\mathbf{x}, \rho)$ che contiene gli attributi $\mathbf{x} = (x_i, x_t, x_u)^T$ e la correlazione $\rho = \text{corr}(\mathbf{w}, \mathbf{k})$ di un blocco.

Una tecnica di addestramento piuttosto diffusa è la cosiddetta *regressione multipla polinomiale*, che consiste nel modellare la correlazione come una combinazione lineare degli attributi e dei loro termini del secondo ordine:

$$\rho = \hat{\rho} + \nu \quad \hat{\rho} = \boldsymbol{\omega}^T \mathbf{y}$$

dove $\mathbf{y} = (1, x_i, x_t, x_u, x_i^2, x_t^2, x_u^2, x_i \cdot x_t, x_i \cdot x_u, x_t \cdot x_u)^T$. L'obiettivo dell'addestramento è determinare il vettore $\boldsymbol{\omega}$ che minimizza l'errore quadratico medio sugli n esempi $(\mathbf{y}_1, \rho_1), (\mathbf{y}_2, \rho_2), \dots, (\mathbf{y}_n, \rho_n)$ [13]:

$$\hat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\text{argmin}} \sum_{j=1}^n (\rho_j - \boldsymbol{\omega}^T \mathbf{y}_j)^2 = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\rho}$$

dove $\boldsymbol{\rho} = (\rho_1, \dots, \rho_n)^T$, mentre $\mathbf{H}$ è una matrice le cui righe sono $\mathbf{y}_1^T, \dots, \mathbf{y}_n^T$. Il termine $(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$ è la matrice *pseudo-inversa* di $\mathbf{H}$. Sotto l'ipotesi che la matrice $\mathbf{H}$ sia quadrata e invertibile, la pseudo-inversa di $\mathbf{H}$ è proprio l'inversa $\mathbf{H}^{-1}$, e dunque $\hat{\boldsymbol{\omega}}$ è la soluzione del sistema lineare $\mathbf{H}\boldsymbol{\omega} = \boldsymbol{\rho}$.



Capitolo 2

TECNICHE DI DENOISING

2.1 RESTAURO DI IMMAGINI

Le immagini digitali spesso contengono distorsioni che possono causare un *degrado* della qualità visiva. Il principale obiettivo del restauro è quello di *riparare* un'immagine degradata, cercando di ripristinare il contenuto informativo e visuale. Le tecniche di restauro si fondano sulla conoscenza a priori del fenomeno che ha causato il degrado, nel tentativo di riuscire a generare il processo inverso e ricostruire quindi l'immagine originale. Da questo punto di vista, il restauro si può dividere in due categorie [9]:

1. **deblurring**. L'obiettivo è ridurre i difetti di nitidezza dell'immagine, che si manifestano come un effetto di *sfocatura*. Questo disturbo può essere causato da un movimento durante la cattura dell'immagine oppure da un errato settaggio della messa a fuoco.
2. **denoising**. L'obiettivo è ridurre i difetti di luminosità dell'immagine, che si presentano come una *granulosità* o *puntinatura cromatica*. Questo disturbo, detto genericamente *rumore*, è sempre presente nelle immagini, perché ha origine nei processi di acquisizione e trasmissione.

Per quanto riguarda il denoising, il modello di degrado più diffuso è quello di rumore bianco gaussiano che si *aggiunge* al segnale originale:

$$Y[i] = X[i] + N[i] \quad i \in \mathbb{Z}^2$$

dove X è l'immagine originale e N è un processo bianco a media nulla.

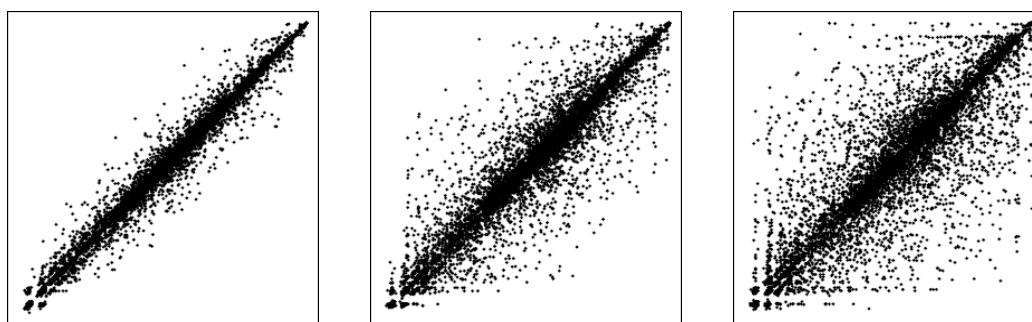
2.2 TECNICHE NEL DOMINIO SPAZIALE

Il denoising è l'operazione che permette di trovare una *buona* stima dell'immagine originale, data l'osservazione rumorosa. L'insieme di tutte le possibili immagini è enorme, ma di queste solo un ristretto numero ha un'alta probabilità di occorrenza. Affinché il denoising abbia successo, quindi, è essenziale che i pixel dell'immagine siano modellati nel migliore dei modi. A questo proposito, si possono fare diverse ipotesi semplificative:

1. **condizione di Markov.** La pdf di un pixel, condizionata dai pixel nel vicinato, è indipendente da tutti gli altri pixel dell'immagine;
2. **stazionarietà.** La pdf di un gruppo adiacente di pixel è indipendente dalla posizione assoluta di tale gruppo all'interno dell'immagine;

Si noti che la stazionarietà non è ben definita ai bordi dell'immagine, e perciò essi richiedono un trattamento *speciale*.

Un'ulteriore proprietà delle immagini è che i pixel adiacenti tendono ad assumere valori di intensità molto simili tra loro. Questo fatto è confermato sperimentalmente nei grafici di Figura 2.1, che riportano numerose coppie di pixel situati a distanza fissa [1]. Il comportamento più evidente è che i pixel adiacenti sono *fortemente correlati* (quando l'intensità di un pixel è alta, anche l'intensità del vicinato tende ad essere alta), ma questo effetto diminuisce all'aumentare della distanza.



(a) Pixel a distanza $\Delta x = 1$. (b) Pixel a distanza $\Delta x = 3$. (c) Pixel a distanza $\Delta x = 8$.

Figura 2.1 Correlazione tra i pixel adiacenti.



Gli algoritmi nel dominio spaziale sono tecniche di denoising che lavorano direttamente sui pixel dell'immagine, senza ricorrere ad alcuna trasformata. Data un'immagine rumorosa $Y[i] = X[i] + N[i]$, il risultato del filtro $\hat{X}[i]$, per ogni posizione i , è una combinazione lineare dei pixel nell'intorno di $Y[i]$:

$$\hat{X}[i] = Y[i] * h[i] = \sum_{j \in V_i} Y[j] \cdot h[i-j]$$

dove V_i è una finestra rettangolare centrata nel pixel $Y[i]$, mentre i pesi $h[i]$ rappresentano la *maschera* del filtro. Il funzionamento della tecnica è schematizzato nella Figura 2.2.

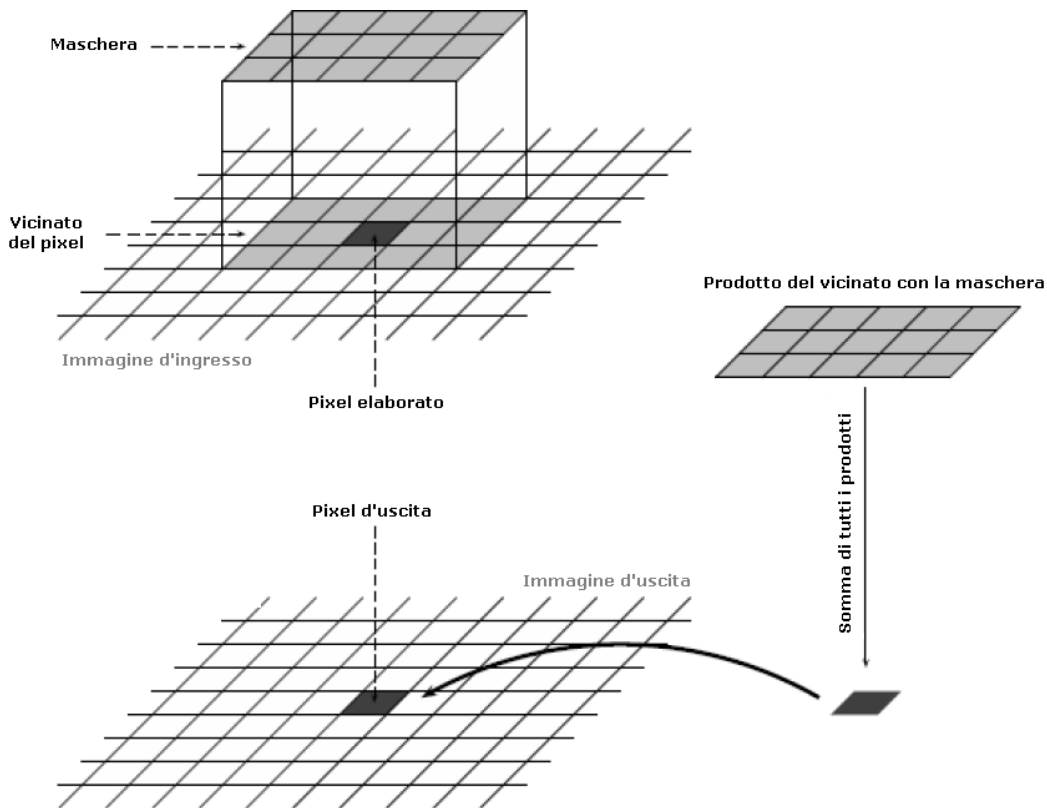


Figura 2.2 Filtraggio locale.



2.2.1 FILTRO A MEDIA PESATA

Il *filtro a media aritmetica* si fonda sull'ipotesi che i pixel adiacenti siano fortemente correlati. Infatti, il generico vicinato V_i di un'immagine è modellato come un segnale costante immerso nel rumore:

$$Y[j] = x + N[j] \quad \forall j \in V_i$$

dove x è una quantità *deterministica* ignota. L'uscita del filtro diventa:

$$\hat{X}[i] = \sum_{j \in V_i} Y[j] \cdot h[i-j] = x \cdot \sum_{j \in V_i} h[i-j] + \sum_{j \in V_i} N[j] \cdot h[i-j]$$

La strategia ottima è la stima non polarizzata a minima varianza. La condizione di stimatore non polarizzato ($E\{\hat{X}\} = x$) conduce a:

$$E\{\hat{X}[i]\} = x \cdot \sum_{j \in V_i} h[i-j] \quad \xRightarrow{E\{\hat{X}\}=x} \quad \sum_{m=1}^M h[m] = 1$$

dove M è la grandezza del vicinato. Viceversa, la varianza dello stimatore è:

$$\text{Var}\{\hat{X}[i]\} = \text{Var}\left\{\sum_{j \in V_i} h[i-j] \cdot N[j]\right\} = \sigma_n^2 \cdot \sum_{m=1}^M h^2[m]$$

da cui si ricava la maschera ottima del filtro:

$$\begin{cases} \min \sum_{m=1}^M h^2[m] \\ \sum_{m=1}^M h[m] = 1 \end{cases} \quad \Rightarrow \quad h[m] = \frac{1}{M}$$

Il filtro a media aritmetica produce un effetto eccessivo di *sfocatura*, a causa del suo comportamento passa-basso. Il problema può essere affrontato pesando i coefficienti in base alla distanza dal centro. A tal proposito, una buona soluzione è fornita dal filtro gaussiano $h[i] = \frac{1}{\sqrt{2\pi}k} \exp\left(-\frac{i^2}{2k^2}\right)$.



2.2.2 FILTRO DI WIENER

Il *filtro di Wiener* è lo stimatore lineare $\hat{X}[i] = Y[i] * h[i]$ che minimizza l'errore quadratico medio $E\{D^2[i]\}$, con $D[i] = \hat{X}[i] - X[i]$. Assumendo che l'immagine originale $X[i]$ sia un processo stazionario a *media nulla*, la risposta impulsiva $h[i]$ è il segnale che rende l'errore $D[i]$ *incorrelato* (principio di ortogonalità) con l'immagine rumorosa $Y[i]$:

$$\begin{aligned} E\{D[i] \cdot Y[i]\} &= 0 \\ E\{X[i] \cdot Y[i]\} &= E\{\hat{X}[i] \cdot Y[i]\} \\ &= h[i] * E\{Y[i] \cdot Y[i]\} \\ R_{xy}[i] &= h[i] * R_y[i] \end{aligned}$$

Se l'immagine originale $X[i]$ è incorrelata con il rumore $N[i]$, abbiamo che $R_{xy}[i] = R_x[i]$ e $R_y[i] = R_x[i] + R_n[i]$, da cui segue:

$$H(\omega) = \frac{S_{xy}(\omega)}{S_y(\omega)} = \frac{S_x(\omega)}{S_x(\omega) + S_n(\omega)} = \frac{\eta(\omega)}{\eta(\omega) + 1}, \quad \eta(\omega) = \frac{S_x(\omega)}{S_n(\omega)}$$

Nel caso di rumore additivo, il filtro di Wiener attenua le armoniche in proporzione al rapporto segnale-rumore $\eta(\omega)$. La soppressione si concentra maggiormente nelle frequenze in cui il segnale è debole, lasciando quasi inalterate le frequenze dove il segnale è forte (vedi Figura 2.3).

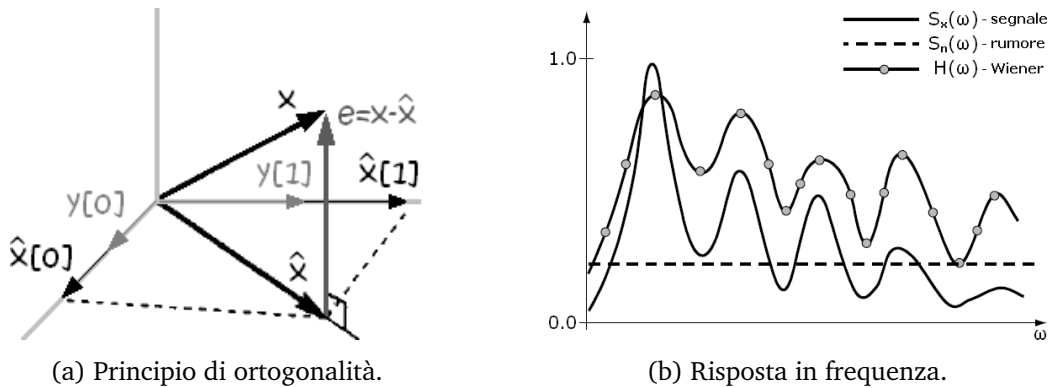


Figura 2.3 Filtro di Wiener.



Il filtro di Wiener esegue un'operazione spazio-invariante, ma in realtà le statistiche dell'immagine variano da regione a regione. Per adattare l'elaborazione alle caratteristiche locali, il filtro è applicato *separatamente* in ogni regione, stimando localmente lo spettro $S_x(\omega)$. Questo conduce a un filtro spazio-variante, dove la risposta $H(\omega)$ varia da regione a regione in funzione delle statistiche locali. Una regione V_i dell'immagine originale è modellata come un segnale a media e varianza costanti [15]:

$$X[j] = \mu_x + \sigma_x \cdot B[j] \quad \forall j \in V_i$$

dove $B[i]$ è un processo bianco a varianza uno. Applicando il filtro di Wiener nella sola regione V_i , l'ipotesi che $X[i]$ è localmente bianco implica:

$$H(\omega) = \frac{S_x(\omega)}{S_x(\omega) + S_n(\omega)} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2}$$

$$\hat{X}[i] = \mu_x + [h[i] * (Y[i] - \mu_x)], \quad h[i] = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} \delta[i]$$

La sottrazione e l'addizione della media serve per soddisfare l'ipotesi (del filtro di Wiener) di processo a media nulla. Siano $\mu_x[i]$ e $\sigma_x^2[i]$ le statistiche locali del blocco centrato in $X[i]$. Osservando che $\mu_y[i] = \mu_x[i]$ e $\sigma_y^2[i] = \sigma_x^2[i] + \sigma_n^2$, il filtro adattativo si può esprimere come:

$$\begin{aligned} \hat{X}[i] &= \mu_x[i] + \frac{\sigma_x^2[i]}{\sigma_x^2[i] + \sigma_n^2} (Y[i] - \mu_x[i]) \\ &= \mu_y[i] + \frac{\sigma_y^2[i] - \sigma_n^2}{\sigma_y^2[i]} (Y[i] - \mu_y[i]) \\ &= Y[i] - \frac{\sigma_n^2}{\sigma_y^2[i]} (Y[i] - \mu_y[i]) \end{aligned}$$

L'elaborazione svolta dal filtro adattativo è molto semplice. Per ogni regione dell'immagine, la media locale resta invariata e la varianza locale è scalata in base all'ampiezza relativa di σ_x^2 e σ_n^2 . L'attenuazione si concentra maggiormente nelle regioni con $\sigma_x^2 \ll \sigma_n^2$ (dove la varianza locale è alta a causa del rumore), lasciando quasi inalterate le regioni con $\sigma_x^2 \gg \sigma_n^2$.



2.3 TECNICHE NEL DOMINIO WAVELET

Le tecniche di denoising nel dominio trasformato si basano sull'ipotesi che i pixel adiacenti del rumore siano fortemente incorrelati. In tal caso, grazie alla correlazione tra i pixel dell'immagine, una qualsiasi trasformata *decorrelante* permette di compattare l'energia del segnale in pochi coefficienti e, allo stesso tempo, disperdere il rumore su molti coefficienti. Il risultato è una rappresentazione incorrelata e sparsa dell'immagine rumorosa che: (1) evidenzia maggiormente le differenze tra il segnale e il rumore, (2) permette di elaborare *scalarmente* i coefficienti trasformati.

La *trasformata wavelet* è basata sul concetto di decomposizione tempo-frequenza. A differenza della trasformata di Fourier (localizzata solo in frequenza) o della trasformata identità (localizzata solo nel tempo), la trasformata wavelet è ben localizzata sia nel tempo che in frequenza. Essa tiene conto che i segnali ad alta frequenza hanno una durata più breve e i segnali a bassa frequenza hanno una durata più lunga. Questo le permette di rappresentare bene sia le componenti stazionarie (sfondi, zone omogenee) che le discontinuità (bordi degli oggetti, zone strutturate) di un'immagine.

La trasformata wavelet suddivide il segnale in due componenti, banda di approssimazione (basse frequenze) e banda di dettaglio (alte frequenze), e decompone ricorsivamente il segnale approssimato fino ad ottenere una rappresentazione multi-scala completa. Tale operazione è facilmente realizzabile attraverso un banco di filtri. Questo la rende uno strumento ideale per le tecniche di denoising, che sono tipicamente organizzate in tre passi:

1. trasformata wavelet dell'immagine rumorosa;
2. denoising *scalare* dei coefficienti $y = x + n$ nelle bande di dettaglio:

$$\hat{x} = f(y)$$

dove $f(\cdot)$ è la cosiddetta funzione di *shrinkage*;

3. anti-trasformata wavelet del risultato.

Lo shrinkage è l'operazione chiave, da cui dipende l'efficacia della tecnica.



2.3.1 SHRINKAGE MARGINALE

Le tecniche di *shrinkage marginale* sono caratterizzate dal fatto che l'operazione $\hat{x} = f(y)$ dipende solo dal coefficiente y , ignorando eventuali correlazioni con i pixel adiacenti. La forma più semplice di shrinkage è il cosiddetto *hard-thresholding* [5], che consiste nell'azzerare i coefficienti più piccoli di una certa soglia T :

$$f(y) = \begin{cases} y & \text{se } |y| > T \\ 0 & \text{se } |y| \leq T \end{cases}$$

L'idea alla base è che il rumore nel dominio trasformato tende a disperdersi su tutti i coefficienti wavelet, per cui una strategia in cui si azzerano i valori sotto-soglia elimina il rumore, ma preserva l'informazione sui bordi dell'immagine. Una forma alternativa di shrinkage è il *soft-thresholding*, che azzerà i coefficienti sotto-soglia e comprime linearmente quelli sopra-soglia:

$$f(y) = \begin{cases} y + T & \text{se } y < -T \\ 0 & \text{se } -T < y < T \\ y - T & \text{se } y > T \end{cases}$$

In entrambi i metodi, la scelta della soglia riveste un ruolo fondamentale, poiché da essa dipendono le prestazioni globali della tecnica. Infatti, una soglia piccola fornisce un risultato ancora rumoroso, mentre una soglia troppo elevata tende ad annullare anche il segnale utile, introducendo un effetto di sfocatura. La soluzione più semplice è utilizzare la cosiddetta *soglia universale*, cioè una soglia uguale per tutte le bande di dettaglio:

$$T_u = \sigma_n \cdot \sqrt{2 \cdot \log(M)} \quad \hat{\sigma}_n = \frac{\text{median}\{HH_1(i)\}}{0.6475}$$

dove $M = M_1 \cdot M_2$ è la dimensione dell'immagine. Una soluzione più sofisticata è utilizzare una *soglia adattativa*, cioè variabile in funzione della banda. Si rimanda alla letteratura scientifica per ulteriori approfondimenti.



2.3.2 SHRINKAGE CONTESTUALE

Le tecniche di *shrinkage contestuale* sono caratterizzate dal fatto che l'operazione $\hat{x} = f(y)$ non dipende solo dal coefficiente y , ma anche dal suo vicinato. I coefficienti non rumorosi $X[i]$ sono modellati come v.a. gaussiane a media nulla e varianza $\sigma_x^2[i]$ spazio-variante [18]. La funzione di shrinkage è lo stimatore lineare che minimizza l'errore quadratico medio:

$$\hat{X}[i] = \alpha \cdot Y[i] \quad \alpha = \operatorname{argmin} E \left\{ (X[i] - \alpha Y[i])^2 \right\}$$

Derivando rispetto alla variabile α e ponendo il risultato a zero, si ottiene:

$$E \left\{ (X[i] - \alpha Y[i]) Y[i] \right\} = 0 \quad \Rightarrow \quad \alpha = \frac{\sigma_x^2[i]}{\sigma_x^2[i] + \sigma_n^2}$$

avendo osservato che $Y[i] = X[i] + N[i]$ e $E\{X[i] \cdot N[i]\} = 0$.

Le varianze $\sigma_x^2[i]$ sono stimate sfruttando le *informazioni contestuali*. A tale scopo, il segnale $X[i]$ è modellato come un processo localmente bianco:

$$X[j] = \sigma_x[i] \cdot B[j] \quad \forall j \in V_i$$

dove $B[i]$ è un processo bianco a varianza unitaria. Il modello afferma che i coefficienti $X[j]$ inclusi nel vicinato V_i sono v.a. indipendenti e identicamente distribuite, ossia $X[j] \sim G(0, \sigma_x^2[i])$ per ogni $j \in V_i$. Sotto tale ipotesi, la stima a *massima verosimiglianza* della varianza $\sigma_x^2[i]$ si esprime come:

$$\hat{\sigma}_x^2[i] = \operatorname{argmax} \prod_{j \in V_i} \Pr \{ Y[j] \mid \sigma_x^2[i] \} = \max \{ 0, \hat{\sigma}_y^2[i] - \sigma_n^2 \}$$

dove $\hat{\sigma}_y^2[i]$ è calcolata nel vicinato V_i di dimensione M centrato in $Y[i]$:

$$\hat{\sigma}_y^2[i] = \frac{1}{M} \sum_{j \in V_i} Y^2[j]$$

La tecnica esposta non è altro che il filtro di Wiener applicato nel dominio wavelet. In questo lavoro di tesi, essa è chiamata *filtro di Mihcak* [18].



2.3.3 SHRINKAGE IN DUE PASSI

Lo *shrinkage in due passi* è una tecnica che unisce lo shrinkage marginale e il filtro di Wiener. Il primo passo impiega il thresholding (hard o soft) per effettuare una stima grossolana dell'immagine non rumorosa, la quale è usata nel secondo passo per stimare le varianze dell'immagine originale, che però vengono adoperate per effettuare lo shrinkage LMMSE sull'immagine rumorosa di partenza [8]. La tecnica è illustrata in Figura 2.4.

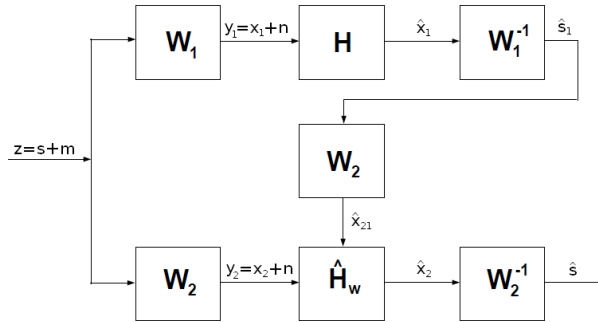


Figura 2.4 Funzionamento dello shrinkage in due passi.

L'algoritmo prevede tre operazioni distinte. Il primo passaggio consiste nel trasformare l'immagine rumorosa con una base wavelet W_1 , effettuare il thresholding e anti-trasformare il risultato, in modo da ottenere una prima stima dell'immagine senza rumore. Il secondo passaggio prevede di stimare le varianze dell'immagine originale come $\sigma_x^2[i] = \hat{x}^2[i]$, dove $\hat{x}[i]$ è la stima al passo 1 trasformata con una base wavelet W_2 . Il terzo passaggio, infine, consiste nel trasformare l'immagine rumorosa con la base wavelet W_2 , effettuare la stima LMMSE dei coefficienti trasformati (usando le varianze calcolate al passo 2) e anti-trasformare il risultato, in modo da ottenere la stima finale dell'immagine senza rumore.

A questo punto resta da affrontare una sola questione: come scegliere le trasformate W_1 e W_2 . L'esperienza mostra che il risultato finale è abbastanza insensibile alla scelta, purché W_1 e W_2 siano trasformate appropriate per lo shrinkage. Inoltre, i risultati sperimentali mostrano che W_1 e W_2 devono essere diverse tra di loro, altrimenti l'algoritmo non offre prestazioni migliori rispetto alle tecniche di shrinkage adoperate singolarmente.



2.4 TECNICHE NON LOCALI

L'inefficacia di molte tecniche di denoising risiede nell'inadeguatezza del modello assunto per l'immagine. Infatti, la strategia adottata da svariati algoritmi è basata sull'ipotesi che il rumore abbia uno spettro piatto e l'immagine originale abbia componenti spettrali significative solo alle basse frequenze. Seguendo questo ragionamento, il rumore può essere soppresso attenuando le alte frequenze e lasciando inalterate quelle basse.

Il risultato prodotto è poco soddisfacente, per diversi motivi. Innanzitutto, le componenti in alta frequenza del segnale vengono sopprese insieme al rumore, non essendo possibile distinguere tra le due. Ne consegue che le forti discontinuità o i bordi degli oggetti, essendo concentrati alle alte frequenze, non sono ricostruiti correttamente. Inoltre, nell'immagine ricostruita permangono le componenti di rumore collocate alle basse frequenze, poiché queste non sono alterate durante l'elaborazione.

Le tecniche di denoising fin qui analizzate impiegano un approccio di tipo *locale*, nel senso che si limitano ad operare esclusivamente in prossimità del pixel da ricostruire. Tale strategia manifesta i suoi limiti nell'incapacità di ricostruire correttamente le texture e le piccole caratteristiche delle immagini. Da diversi anni, la ricerca sta concentrando i propri sforzi sullo sviluppo di tecniche che cercano di superare questo limite.

Un'idea innovativa proviene da uno studio sulla sintesi di texture, che prevede di sintetizzare il generico pixel usando le informazioni presenti nell'intera immagine, oltre a quelle ricavabili dai pixel nelle immediate vicinanze. Grazie al fatto che tutte le immagini naturali presentano al loro interno texture ed elementi ripetitivi, il concetto può essere esteso alle tecniche di denoising, applicando processi analoghi a quelli per il restauro di texture. Infatti, l'elevato grado di ridondanza delle immagini può essere sfruttato per ricostruire i dettagli più fini e le strutture ripetitive, possibilmente senza creare artefatti. Sulla base di questo ragionamento, le tecniche di denoising *non locale* elaborano ciascun pixel sfruttando le informazioni situate in tutta l'immagine, ma senza trascurare la correlazione presente nelle immediate vicinanze del pixel sotto elaborazione.



2.4.1 ALGORITMO NL-MEANS

L'algoritmo *non-local means* filtra l'immagine rumorosa $Y[i] = X[i] + N[i]$ attraverso un'operazione di media aritmetica [2]. A differenza delle tecniche nel dominio spaziale, l'elaborazione del pixel $Y[i]$ coinvolge un'estesa porzione dell'immagine rumorosa, detta *finestra di ricerca*, che indichiamo con S_i e che al limite può anche coincidere con l'intera immagine. I pixel che ricadono nella finestra di ricerca S_i (centrata in $Y[i]$) sono pesati in base alla somiglianza del proprio vicinato (una finestra quadrata V_j centrata nel pixel $Y[j] \in S_i$) con il vicinato di $Y[i]$:

$$\hat{X}[i] = \sum_{j \in S_i} w(i, j) \cdot Y[j]$$

dove i pesi della sommatoria sono espressi come:

$$w(i, j) = \frac{1}{Z(i)} \exp \left[-\frac{d_{i,j}}{h^2} \right] \quad Z(i) = \sum_{j \in S_i} \exp \left[-\frac{d_{i,j}}{h^2} \right]$$

Il termine $d_{i,j}$ indica la distanza euclidea tra i vicinati di $Y[i]$ e $Y[j]$ moltiplicati per un kernel gaussiano con deviazione standard k :

$$d_{i,j} = \| Y[V_i] - Y[V_j] \|_k^2$$

Viceversa, il termine h controlla la velocità di decadimento della funzione esponenziale, e dunque determina l'*intensità* del filtraggio complessivo.

L'unico peso che viene calcolato in maniera differente è quello associato al pixel centrale $Y[i]$. Infatti, se fosse calcolato normalmente, esso risulterebbe il più grande di tutti e il suo vicinato avrebbe un'importanza maggiore degli altri, producendo un effetto simile a un filtro locale. Per evitare questo problema, si pone $w(i, i)$ al massimo peso stimato per gli altri pixel:

$$w(i, i) = \max_{j \in S_i} \{ w(i, j) : j \neq i \}$$

Si rimanda all'articolo originale per la scelta ottimale dei vari parametri.



2.4.2 ALGORITMO BM3D

L'algoritmo BM3D è una tecnica di denoising che impiega l'approccio non locale nel dominio trasformato [4], in modo da ottenere una rappresentazione molto sparsa dei dati e, quindi, una separazione più netta tra il segnale e il rumore. L'algoritmo è organizzato in due passi che, singolarmente, effettuano tre operazioni: raggruppamento, filtraggio collaborativo e aggregazione. L'elaborazione complessiva dell'algoritmo è illustrata in Figura 2.5, dove si evidenziano le differenze tra i due passi.

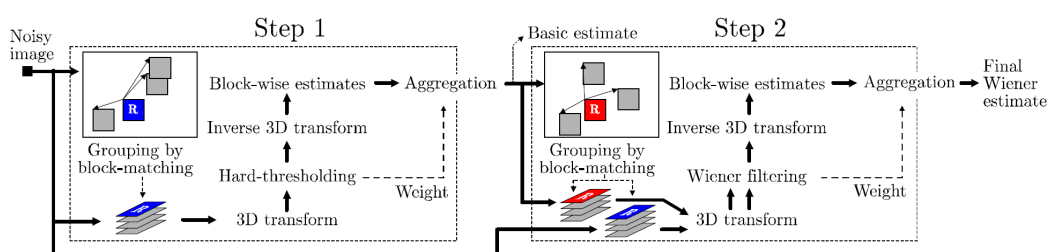


Figura 2.5 Funzionamento dell'algoritmo BM3D.

RAGGRUPPAMENTO La prima operazione dell'algoritmo è il *raggruppamento*, che consiste nel trovare i blocchi simili dell'immagine e costruire con essi una struttura tridimensionale chiamata *gruppo*. In entrambi i passi, la misura di *non similarità* è la distanza euclidea, calcolata tra il blocco di riferimento e tutti i blocchi (scorrevoli) compresi in un'ampia zona di ricerca. Un blocco si considera *simile* (a quello di riferimento) quando la distanza è inferiore ad una certa soglia, nel caso contrario è scartato. La dimensione massima del gruppo è limitata ad un certo numero di blocchi, disposti in ordine crescente di distanza. L'unica differenza sostanziale tra i due passi è che nel primo le distanze sono calcolate sull'immagine rumorosa, mentre nel secondo sono calcolate sul risultato del primo passo.

FILTRAGGIO COLLABORATIVO Il risultato del raggruppamento è sottoposto al *filtraggio collaborativo*, che prevede la trasformata 3D del gruppo, lo shrinkage nel dominio trasformato e l'anti-trasformata 3D del risultato. La



la differenza tra i due passi è che nel primo si esegue il *thresholding* e nel secondo si applica il filtro di Wiener (stima LMMSE). Il termine *collaborativo* è dovuto al fatto che ogni blocco del gruppo partecipa al filtraggio di tutti gli altri. Questa tecnica risulta essere molto efficace sulle immagini naturali, perché sfrutta la correlazione tra i pixel dello stesso blocco e, cosa più importante, la correlazione tra i diversi blocchi del gruppo. Ne consegue una rappresentazione molto sparsa nel dominio trasformato.

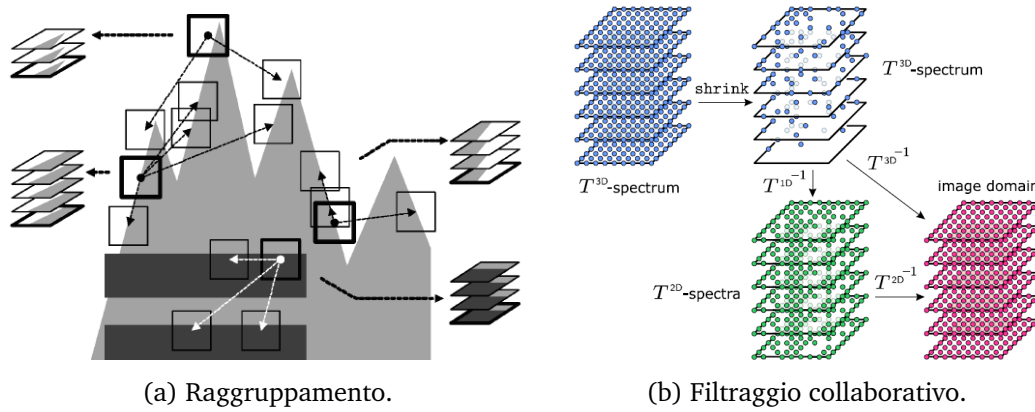


Figura 2.6 Passi intermedi dell'algoritmo BM3D.

AGGREGAZIONE Nella fase di raggruppamento, i blocchi selezionati possono avere delle sovrapposizioni e, inoltre, possono entrare a far parte di diversi gruppi. Questa è una forma di ridondanza, nel senso che esiste più di una stima per alcuni pixel dell'immagine. L'*aggregazione* consiste nell'effettuare la media pesata delle diverse stime di un pixel, al fine di ottenere una stima finale più accurata. Il peso è uguale per tutti i blocchi di ogni gruppo, e può variare solo tra blocchi appartenenti a gruppi diversi. La scelta dei pesi è tale da assegnare un'importanza minore alle stime *più rumorose*. In entrambi i passi, essi sono calcolati come il reciproco della varianza totale dei campioni del gruppo, stimata nel dominio trasformato. Nel calcolo della varianza totale del gruppo, si ipotizza l'indipendenza del rumore tra i vari blocchi, per motivi di semplicità. In realtà, questa ipotesi è vera solamente quando i blocchi del gruppo sono disgiunti.



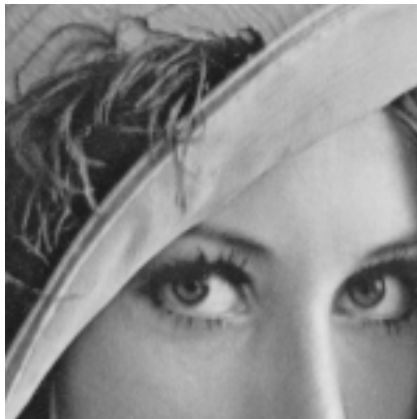
2.5 CONFRONTO DELLE PRESTAZIONI

Le figure 2.7 e 2.9 mostrano i risultati delle tecniche di denoising sulle immagini *Lena* e *Barbara*, riassunti nella Tabella 2.1. La superiorità di BM3D è abbastanza evidente, soprattutto sull'immagine *Barbara*, dove l'approccio non locale è favorito dall'abbondanza di texture e strutture ripetitive. Sull'immagine *Lena*, le tecniche locali sono meno penalizzate, data l'assenza di pattern periodici. La tecnica *Wiener ideale* (shrinkage LMMSE nel dominio wavelet non decimato che usa le varianze stimate dall'immagine senza rumore) è un limite superiore per le tecniche locali.

Tabella 2.1 Rapporto segnale-rumore di picco per le tecniche di denoising.

TECNICHE DI DENOISING		DEVIAZIONE STD DEL RUMORE			
		10	15	20	25
LENA	<i>Hard-Thresh. universale</i>	31.48	29.89	28.73	27.86
	<i>Filtro di Wiener</i>	33.59	31.13	28.99	27.19
	<i>Soft-Thresh. adattativo</i>	33.42	31.36	29.87	28.67
	<i>Shrinkage contestuale</i>	34.39	32.38	30.96	29.84
	<i>Shrinkage in due passi</i>	34.41	32.53	31.15	30.09
	<i>NL-means</i>	34.81	32.72	31.51	30.36
	<i>BM3D</i>	35.93	34.27	33.05	32.08
	<i>Wiener ideale (SWT)</i>	37.12	35.26	34.02	33.06
BARBARA	<i>Hard-Thresh. universale</i>	27.92	25.75	24.41	23.62
	<i>Filtro di Wiener</i>	31.35	28.58	26.67	25.19
	<i>Soft-Thresh. adattativo</i>	31.25	28.72	27.05	25.81
	<i>Shrinkage contestuale</i>	32.67	30.26	28.65	27.43
	<i>Shrinkage in due passi</i>	32.08	29.67	28.04	26.82
	<i>NL-means</i>	33.72	31.67	30.38	29.19
	<i>BM3D</i>	34.98	33.11	31.78	30.72
	<i>Wiener ideale (SWT)</i>	35.23	32.99	31.49	30.37





(a) Originale.



(b) Rumorosa.



(c) Shrinkage marginale.



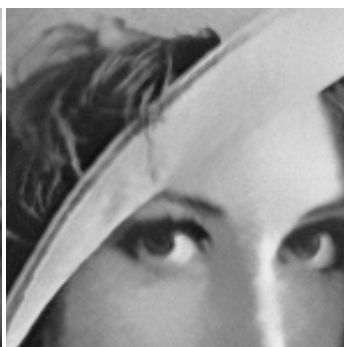
(d) Filtro di Wiener.



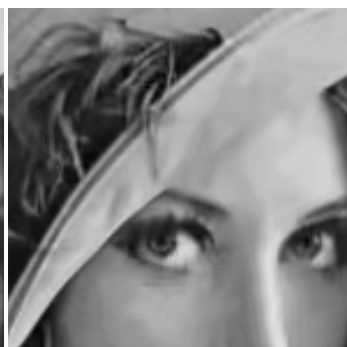
(e) Shrinkage contestuale.



(f) Shrinkage in due passi.



(g) NL-means.



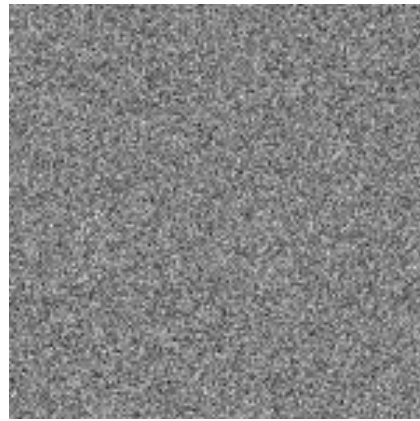
(h) BM3D.

Figura 2.7 Denoising dell'immagine "Lena" ($\sigma_n = 20$).

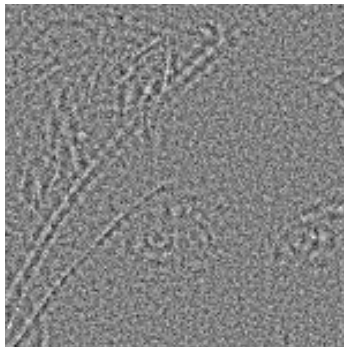




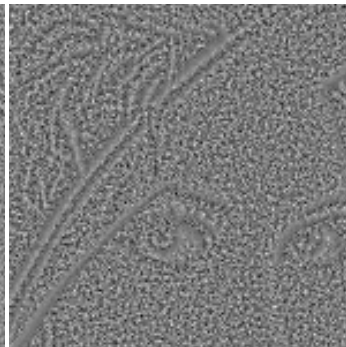
(a) Originale.



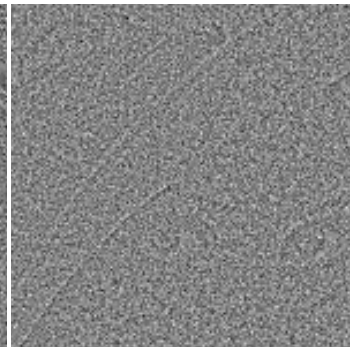
(b) Rumore AWGN.



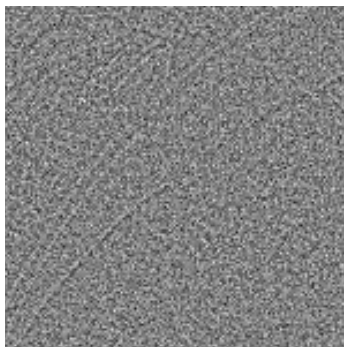
(c) Shrinkage marginale.



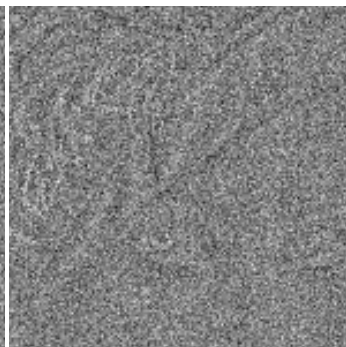
(d) Filtro di Wiener.



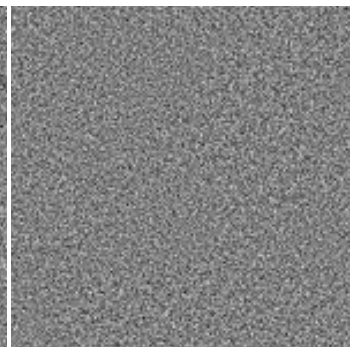
(e) Shrinkage contestuale.



(f) Shrinkage in due passi.



(g) NL-means.



(h) BM3D.

Figura 2.8 Residui di rumore dell'immagine "Lena" ($\sigma_n = 20$).



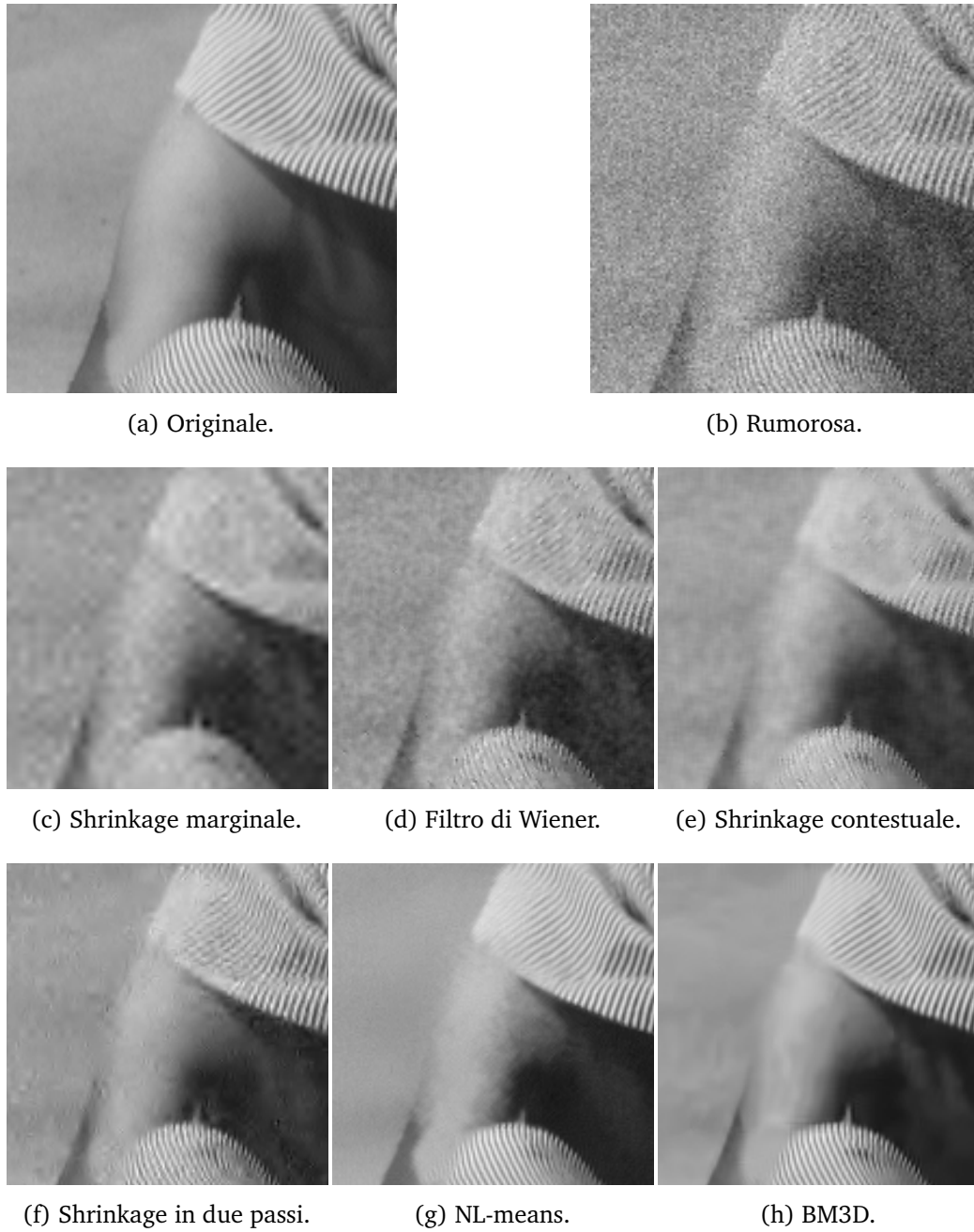


Figura 2.9 Denoising dell'immagine "Barbara" ($\sigma_n = 20$).



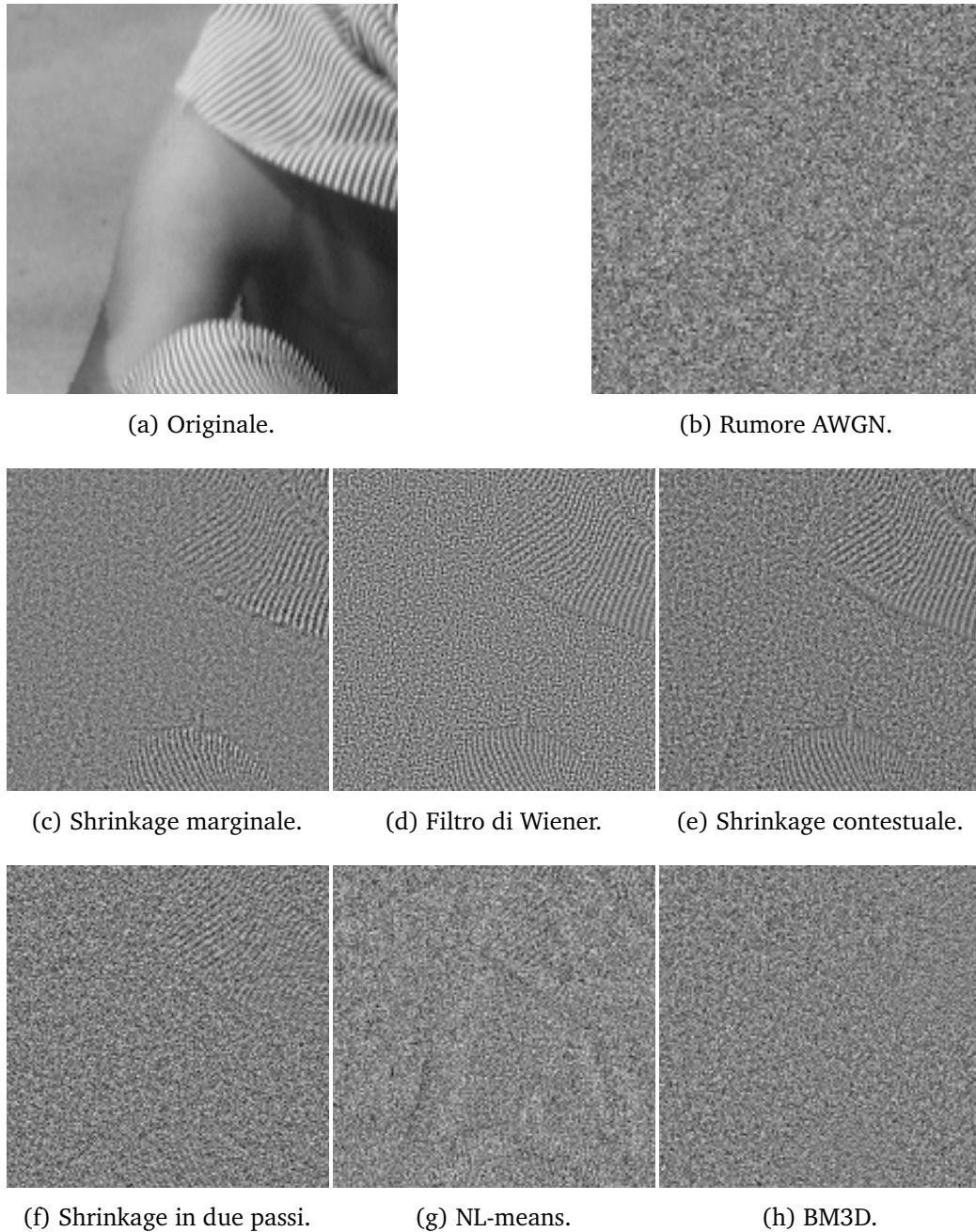


Figura 2.10 Residui di rumore dell'immagine "Barbara" ($\sigma_n = 20$).



*La teoria è quando si sa tutto e nulla funziona.
La pratica è quando tutto funziona e non
si sa il perché. Noi abbiamo messo insieme
la teoria e la pratica: non c'è niente che
funzioni... e nessuno sa il perché!*

ALBERT EINSTEIN

Capitolo 3

RISULTATI SPERIMENTALI

3.1 INTRODUZIONE

A causa della “non uniformità di risposta” del sensore, le fotocamere digitali introducono un lieve disturbo nelle immagini catturate, detto *Photo Response Non Uniformity*. Essendo unico per ogni fotocamera, costante nel tempo e indipendente dalla scena, il PRNU può essere considerato una sorta di *impronta digitale*. Tra le altre cose, esso permette di scoprire se un'immagine è stata manipolata. L'idea è molto semplice: un'immagine si può considerare autentica se ogni pixel esibisce il PRNU della fotocamera origine. La natura moltiplicativa di questo disturbo, infatti, rende molto difficile manomettere un'immagine senza alterare il PRNU nella regione contraffatta. Ne consegue che la mancata rivelazione del PRNU è un forte suggerimento circa la presenza di qualche forma di contraffazione.

Il procedimento che permette di stabilire l'integrità di un'immagine è organizzato in due fasi:

1. stima del PRNU della fotocamera (una sola volta per ogni fotocamera);
2. ricerca del PRNU all'interno dell'immagine sotto analisi.

Il denoising è un passo fondamentale in entrambe le fasi, poiché consente di rimuovere le componenti aleatorie dell'immagine, e rivelare quindi il PRNU. L'obiettivo di questo capitolo è analizzare l'influenza del denoising sulle prestazioni globali dell'algoritmo.

3.2 PREPARAZIONE DELLA FOTOCAMERA

La prima fase dell'algoritmo consiste nel raccogliere le informazioni necessarie per realizzare la verifica d'integrità. Supponendo di conoscere la fotocamera che ha catturato l'immagine sotto analisi, l'algoritmo di verifica richiede tre dati: (1) l'impronta digitale della fotocamera, (2) la soglia per il test d'ipotesi e (3) i pesi del predittore. Il procedimento che permette di stimare le grandezze suddette, detto *preparazione della fotocamera*, è realizzato nel Programma 3.1.

Programma 3.1 *Preparazione della fotocamera.*

```

1 %
2 % Created on: 09/06/10 - Giovanni Chierchia
3 %
4 % Lo script prepara una fotocamera eseguendo tre operazioni:
5 %   1. stima del PRNU su 200 immagini
6 %   2. stima dell'ipotesi H0 su 50,000 correlazioni (da 200 immagini)
7 %   3. addestramento del predittore su 10,000 correl. (20 immagini)
8
9 target = 'canonEOS10D';      % fotocamera "target"
10 denoising = {                % filtri di denoising
11     'mihcak_filter(X,3)'
12     'bm3d_filter(X,5)'
13 };
14 radius = 64;                % dimensione dei blocchi (129x129)
15 params = [250 6 0.03];      % parametri per estrarre gli attributi
16
17
18 % fotocamere disponibili
19 cameras = {'canonEOS10D', 'olympusE30', 'sonyDSCV3'};
20
21 % immagini per stimare il PRNU
22 prnu_dir = ['Foto\' target '\prnu'];
23 prnu_images = discover_images( prnu_dir, 'jpg', 200 );
24
25 % immagini per stimare l'ipotesi H0
26 others_dir = strcat('Foto\'', setdiff(cameras, target), '\prnu');
27 H0_images = discover_images( others_dir, 'jpg', 200 );

```



```

28
29 % immagini per addestrare il predittore
30 pred_dir = ['Foto\' target '\predittore'];
31 H1_images = discover_images( pred_dir, 'jpg', 20 );
32
33
34 %%%%%%%%%%%%%%% esegui il test %%%%%%%%%%%%%%%
35 for idx = 1:length(denoising)
36
37     % funzione che estrae il residuo di rumore
38     fetch = inline( ['X - ' denoising{idx}] );
39
40     % stima del PRNU
41     pattern = estimate_prnu( prnu_images, fetch );
42
43     % stima dell'ipotesi H0
44     corr_H0 = random_blocks( H0_images, pattern, 50000, radius );
45     ggd_H0 = ggfit( corr_H0(:) );
46
47     % addestramento del predittore (ipotesi H1)
48     [weights ggd_H1] = estimate_predictor( H1_images, pattern,
49                                         10000, radius, params);
50
51     % salva i risultati
52     camera.pattern = pattern;
53     camera.hypotesis = struct('forged', ggd_H0,
54                             'authentic', ggd_H1,
55                             'radius', radius
56                             );
57     camera.predictor = struct('weights', weights, 'params', params);
58     save( ['mat\' target '_' num2str(idx)], 'camera' );
59 end

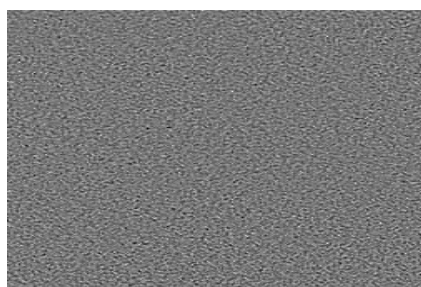
```

Il Programma 3.1 effettua la preparazione della fotocamera usando due diversi filtri di denoising: mihcak e bm3d. In entrambi i casi, la funzione `estimate_prnu` stima il PRNU usando 200 immagini provenienti dalla fotocamera, la funzione `ggfit` stima la distribuzione dell'ipotesi H_0 , mentre la funzione `estimate_predictor` addestra il predittore.

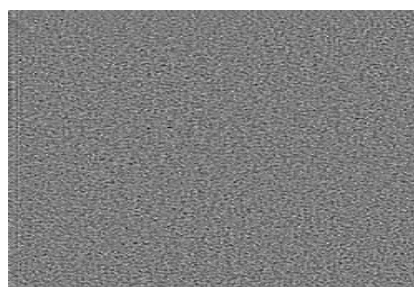


3.2.1 STIMA DEL PRNU

L'impronta digitale della fotocamera è calcolata usando lo stimatore a massima verosimiglianza descritto nella sezione 1.2.1 (e implementato nel Programma 3.2). La Figura 3.1 illustra i risultati che si ottengono con i filtri *mihcak* e *bm3d*. La correlazione tra le due impronte è del 97%. Questo significa che il denoising *non* è fondamentale per ottenere una buona stima del PRNU. Ciò è dovuto all'operazione di media aritmetica, che sopprime le componenti aleatorie (compresi gli artefatti del denoising).



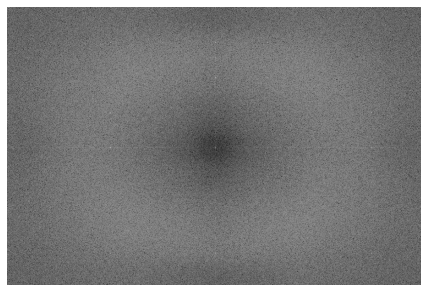
(a) Filtro *bm3d*.



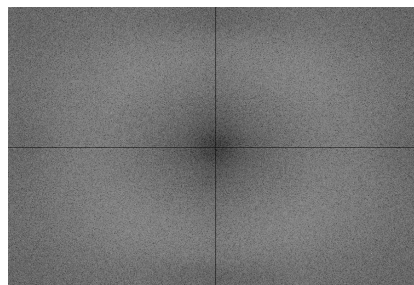
(b) Filtro *mihcak*.

Figura 3.1 Stima sperimentale del PRNU.

Il risultato ottenuto dallo stimatore è sottoposto all'operazione di *zero-mean*, per diminuire la correlazione tra le impronte che provengono da fotocamere della stessa linea produttiva. La Figura 3.2 mostra gli effetti dell'operazione sullo spettro del PRNU.



(a) Prima dell'operazione *zero-mean*.



(b) Dopo l'operazione *zero-mean*.

Figura 3.2 Spettro del PRNU.



Programma 3.2 *Stima del PRNU.*

```

1 function pattern = estimate_prnu( images, fetch_residue )
2 %
3 % La funzione effettua la stima del PRNU su N immagini.
4 %
5 % PARAMETRI D'INGRESSO:
6 %   | parametro | tipo | commento
7 % ---|-----|-----|-----
8 % 1 | images    | cella 1xN | lista delle immagini da elaborare
9 % 2 | fetch_residue | handler | funzione che estrae il residuo
10
11 % scorri le immagini
12 for n=1:length(images)
13
14     % carica l'immagine
15     if ischar(images{n})
16         X = double( imread(images{n}) );
17     else
18         X = images{n};
19     end
20
21     % estrai il residuo di rumore
22     W = fetch_residue(X);
23
24     % accumula il residuo di rumore
25     if n == 1
26         num = W .* X;
27         den = X .^ 2;
28     else
29         num = num + W .* X;
30         den = den + X .^ 2;
31     end
32 end
33
34 % pre-processing del PRNU
35 den( den==0 ) = 1; % impedisce la divisione per zero
36 prnu = zero_mean(num ./ den);
37
38 % costruisci l'uscita
39 pattern = struct('prnu', prnu, 'fetch', fetch_residue);

```



3.2.2 STIMA DELLE IPOTESI

Le distribuzioni del test d'ipotesi sono stimate attraverso il metodo dei momenti descritto nella sezione 1.3.1 (e implementato nel Programma 3.3). L'ipotesi H_0 viene calcolata usando le correlazioni tra il PRNU della fotocamera e i residui di rumore presi da altre fotocamere. Per calcolare l'ipotesi H_1 , invece, i residui di rumore devono essere presi dalla fotocamera origine. La Figura 3.3 mette a confronto i risultati che si ottengono con i filtri *mihcak* e *bm3d*. Come si può vedere, il filtro *bm3d* permette di separare meglio gli istogrammi delle ipotesi H_0 e H_1 . Questo significa che il denoising svolge un ruolo fondamentale per isolare il PRNU all'interno di una singola immagine.

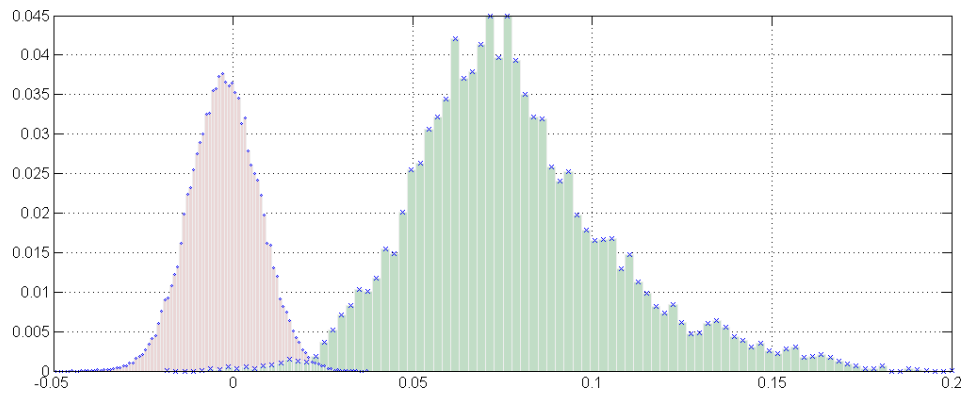
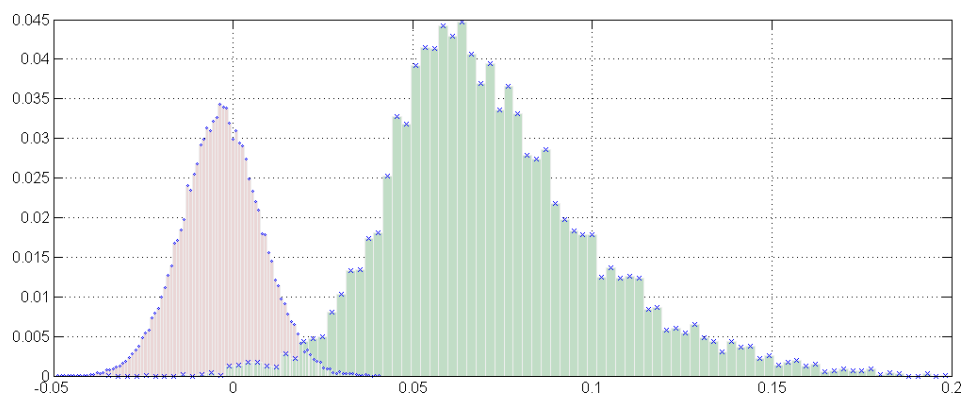
(a) Filtro *bm3d*.(b) Filtro *mihcak*.

Figura 3.3 Stima sperimentale delle ipotesi.



Programma 3.3 *Stima della gaussiana generalizzata.*

```

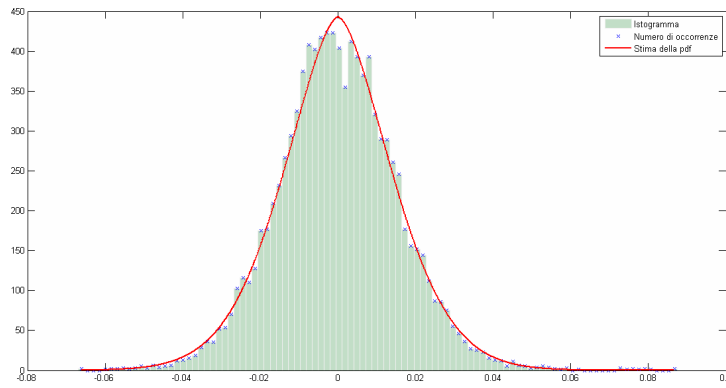
1 function y = ggfit(x, opt)
2 %
3 % La funzione stima i parametri (m,a,b) della PDF gauss. general.
4 %
5 % Il parametro "opt" permette di scegliere tra due stimatori:
6 %   - 'mme' = stima con il metodo dei momenti (DEFAULT)
7 %   - 'mle' = stima a massima verosimiglianza
8 %
9 % L'uscita è il vettore y = [m a b] dei parametri stimati.
10
11 % parametro di default
12 if nargin < 2
13     opt = 'mme';
14 end
15
16 % controllo dei parametri
17 if numel(x) ~= length(x)
18     error('l\'ingresso deve essere un vettore riga o colonna');
19 end
20
21 % stima MME del valor medio
22 m = mean(x);
23
24 % stima MME del fattore di forma
25 num = mean( abs(x-m) );
26 den = mean( (x-m).^2 );
27 b = ggr_inv( num^2 / den );
28
29 % stima MME del coefficiente di scala
30 a = num * gamma( 1 / b ) / gamma( 2 / b );
31
32 % costruisci l'uscita
33 if strcmpi( 'mme', opt )
34     y = [m a b];
35 elseif strcmpi( 'mle', opt )
36     y = mle(x, 'pdf', @ggpdf, 'start', [m a b], 'lower', [-Inf 0 0]);
37 else
38     error('opzione non riconosciuta');
39 end

```

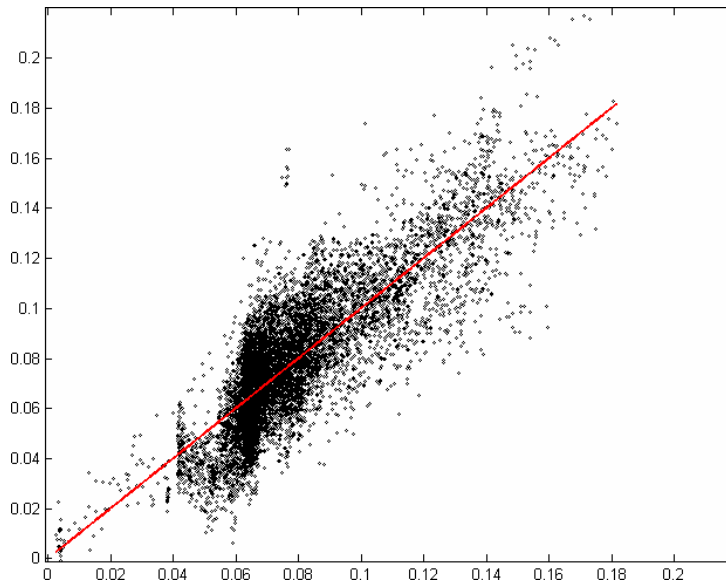


3.2.3 ADDESTRAMENTO DEL PREDITTORE

Per addestrare il predittore, si utilizzano 5000 esempi $(\mathbf{x}, \rho)$ estratti da 20 immagini molto variegiate della fotocamera. Gli attributi $\mathbf{x} = (x_i, x_t, x_u)$ e la correlazione $\rho = \text{corr}(\mathbf{k}, \mathbf{w})$ sono calcolati su blocchi scorrevoli di dimensioni 128×128 . La Figura 3.4 illustra i risultati dell'addestramento: la pdf degli errori di predizione (3.4-a) e la qualità della regressione (3.4-b). L'addestramento è implementato nel Programma 3.4.



(a) Distribuzione degli errori.

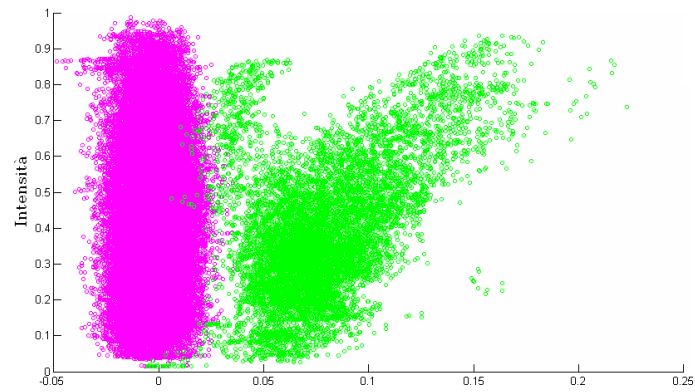


(b) Regressione.

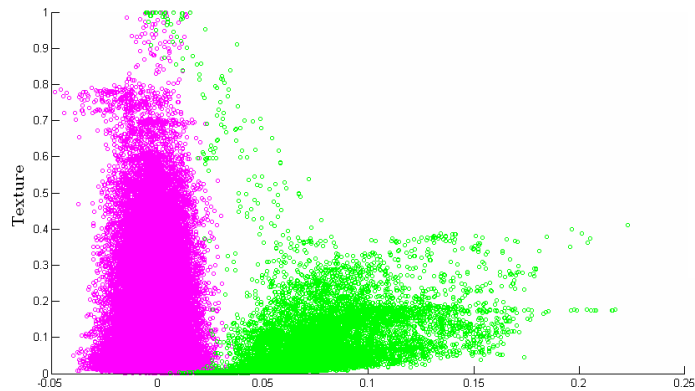
Figura 3.4 Risultati dell'addestramento.



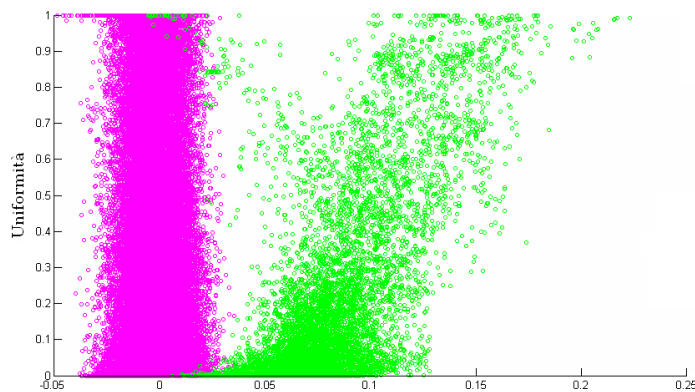
La Figura 3.5 illustra diverse migliaia di esempi che rappresentano l'ipotesi H_0 (punti rossi) e l'ipotesi H_1 (punti verdi). Naturalmente, l'addestramento del predittore è realizzato solo sugli esempi dell'ipotesi H_1 .



(a) Correlazione-Intensità.



(b) Correlazione-Struttura.



(c) Correlazione-Uniformità.

Figura 3.5 *Grafici scatter degli attributi.*



Programma 3.4 *Addestramento del predittore.*

```

1 function [weights ggd corr] = estimate_predictor(images, pattern,
    N_corr, radius, params)
2 %
3 % Created on: 01/06/10 - Giovanni Chierchia
4 %
5 % La funzione costruisce il predittore e stima la distribuzione degli
6 % errori di predizione. A tal fine, vengono calcolate le correlazioni
7 % tra blocchi (2r+1)*(2r+1) presi casualmente da: il PRNU e i residui
8 % di rumore estratti dalle immagini.
9 %
10 %
11 % PARAMETRI D'INGRESSO:
12 %      | parame. |   tipo   |          commento
13 % ---|-----|-----|-----
14 %  1 | images  | cell 1xN | lista delle immagini da elaborare
15 %  2 | pattern | struct | PRNU + funzione che estrae il residuo
16 %  3 | N_corr  | intero | numero totale di correlazioni
17 %  4 | radius  | intero | raggio dei singoli blocchi
18 %  5 | params  | vet. 1x2 | parametri per l'estrazione delle feature
19
20
21 % parametri di default
22 if nargin < 5
23     radius = 64;
24 end
25
26 % selezione casuale dei blocchi
27 [corr feat] = random_blocks(images, pattern, N_corr, radius, params);
28
29 % disponi le features sulle colonne
30 feat = reshape( feat, [size(feat,1)*size(feat,2) size(feat,3)] );
31
32 % regressione multipla con termini del 1° e 2° ordine
33 design = x2fx(feat, 'quadratic');
34 weights = regress( corr(:), design );
35
36 % stima la pdf degli errori di predizione
37 err = corr(:) - design*weights;
38 ggd = ggfit(err);

```



3.3 VERIFICA D'INTEGRITÀ

La verifica d'integrità è un algoritmo che si articola in diversi passi:

1. **statistiche locali.** Il primo passo consiste nel raccogliere la statistica sulla quale è basato il test d'ipotesi. Dopo aver estratto il residuo di rumore dall'immagine, si scorre una finestra 128×128 e si calcola la correlazione locale $\rho = \text{corr}(\mathbf{w}, \mathbf{k})$ tra il blocco $\mathbf{w}$ del residuo di rumore e il rispettivo blocco $\mathbf{k}$ del PRNU. Il risultato è una matrice con le stesse dimensioni dell'immagine sotto analisi;
2. **test d'ipotesi.** Il passo centrale prevede di effettuare il test d'ipotesi su ogni pixel dell'immagine. A tale scopo, i parametri α_0 e β_0 dell'ipotesi H_0 sono impiegati per ricavare la soglia del test, la quale è messa a confronto con le correlazioni locali. Il risultato è una maschera binaria che, per ogni pixel, indica l'esito del test;
3. **predizione.** Il passo successivo si occupa di correggere le decisioni del test d'ipotesi. L'obiettivo è ridurre la probabilità di etichettare un pixel autentico come contraffatto. Tale errore si verifica soprattutto nelle zone scure, piatte o strutturate dell'immagine. La funzione del predittore è adattare l'ipotesi H_1 al contenuto dell'immagine, in modo da correggere un'eventuale decisione negativa (PRNU assente) quando il pixel sotto analisi si trova in una delle suddette regioni;
4. **elaborazione finale.** Il passo conclusivo prevede di modificare la maschera dei pixel contraffatti. Innanzitutto, si rimuovono le regioni più piccole di 64×64 , perché la natura locale dell'algoritmo impone un limite inferiore sull'ampiezza delle regioni identificabili. Successivamente, si effettua una dilatazione morfologica, per compensare il fatto che i bordi delle regioni contraffatte non sono identificati correttamente, visto che le decisioni sono attribuite solo al pixel centrale.

L'algoritmo di verifica è implementato nel Programma 3.5.



Programma 3.5 Verifica d'integrità.

```

1 function mask = detect_forgery( img, pattern, hypotesis, predictor )
2 %
3 % PARAMETRI D'INGRESSO:
4 %      | parametro |      tipo      |      commento
5 % ----|-----|-----|-----
6 %  1 |   img   | vet. MxNxK | immagine da analizzare
7 %  2 | pattern | struct   | PRNU, funzione che estrae il residuo
8 %  3 | hypotesis | struct   | ipo. H0, ipo. H1, raggio dei blocchi
9 %  4 | predictor | struct   | predittore, param. per le features
10
11 % estrazione del residuo di rumore
12 residuo = pattern.fetch(img);
13 X = color2gray(residuo);
14
15 % calcolo delle correlazioni "locali"
16 Y = color2gray(pattern.prnu);
17 corr = extract_correlations(X, Y, hypotesis.radius);
18
19 % test d'ipotesi
20 a0 = hypotesis.forged(2);
21 b0 = hypotesis.forged(3);
22 thresh = 2 * sqrt( a0^2 * gamma(3/b0) / gamma(1/b0) );
23 mask = double(corr < thresh);
24
25 % predizione delle correlazioni
26 w = predictor.weights; p = predictor.params;
27 p_corr = predict_correlations(img, w, hypotesis.radius, p);
28
29 % correzione delle decisioni
30 a1 = hypotesis.authentic(2) * ones(Nr,Nc);
31 b1 = hypotesis.authentic(3) * ones(Nr,Nc);
32 prob = ggcdf(thresh, p_corr, a1, b1);
33 mask( prob > 0.01 ) = 0;
34
35 % rimozione delle regioni piccole
36 mask = bwareaopen( mask, hypotesis.radius^2 );
37
38 % dilatazione dei bordi
39 mask = imdilate( mask, strel('square',20) );

```



Programma 3.6 *Calcolo delle correlazioni.*

```

1 function Z = extract_correlations(X, Y, rad)
2
3 % finestra scorrevole
4 win = ones(2*rad+1);
5 n = numel(win);
6
7 % calcola le somme locali
8 Sx = imfilter(X, win);
9 Sy = imfilter(Y, win);
10
11 % calcola le correlazioni locali
12 Cxy = imfilter(X.*Y, win) - (1/n) * Sx .* Sy;
13
14 % calcola le varianze locali
15 Qx = imfilter(X.^2, win) - (1/n) * Sx.^2;
16 Qy = imfilter(Y.^2, win) - (1/n) * Sy.^2;
17
18 % normalizza
19 Z = Cxy ./ sqrt(Qx .* Qy);

```

Programma 3.7 *Predizione delle correlazioni.*

```

1 function corr = predict_correlations(X, weights, radius, params)
2
3 % estrai le features dall'immagine
4 F = extract_features( color2gray(X), radius, params );
5
6 % disponi le features lungo le colonne
7 Fx = reshape( F, [size(F,1)*size(F,2) size(F,3)] );
8
9 % aggiungi i termini del 2° ordine
10 feat = x2fx(Fx, 'quadratic');
11
12 % effettua la predizione
13 corr = feat * weights;
14
15 % disponi le correlazioni su una matrice
16 corr = reshape( corr, [size(F,1) size(F,2)] );

```



3.3.1 ESPERIMENTI SUL DENOISING

Gli istogrammi di Figura 3.3 (nella pagina 48) suggeriscono che l'algoritmo di verifica dovrebbe comportarsi meglio quando si utilizza il filtro *bm3d*. Questo fatto è dimostrato sperimentalmente nelle figure da 3.6 a 3.8, dove si mostrano i risultati dell'algoritmo al variare del filtro di denoising. La tabella 3.1 riporta, per ogni esperimento, le percentuali di accuratezza, falso negativo (frazione dei pixel autentici dichiarati contraffatti) e vero negativo (frazione dei pixel contraffatti dichiarati tali).

Gli esperimenti mettono in luce anche un secondo aspetto. La predizione riveste un ruolo importante quando si utilizza il filtro *mihcak*, mentre è essenzialmente inutile con l'algoritmo *bm3d*. Questo non dovrebbe sorprendere. Infatti, lo scopo della predizione è correggere le decisioni nelle zone scure, piatte o strutturate dell'immagine, dove la correlazione è sempre bassa (anche in assenza di contraffazioni). La maggiore accuratezza di *bm3d* permette di ottenere una stima migliore del PRNU nelle zone strutturate, e dunque viene meno la necessità di correggere la decisione in tali regioni. Naturalmente, la predizione è ancora necessaria nelle zone scure o piatte, dove la correlazione è bassa per motivi che non sono legati al denoising.

Tabella 3.1 Esperimenti sul denoising: accuratezza e falso/vero negativo.

		FILTRO MIHCAK		FILTRO BM3D	
		senza pred.	con pred.	senza pred.	con pred.
Foto 1	Acc (%)	95.91	96.59	98.18	98.18
	R _{FN} (%)	4.20	0.16	0.84	0.84
	R _{TN} (%)	96.56	68.62	89.77	89.77
Foto 2	Acc (%)	97.69	97.69	97.94	97.94
	R _{FN} (%)	0.0	0.0	0.0	0.0
	R _{TN} (%)	36.79	36.79	43.60	43.60
Foto 3	Acc (%)	98.59	89.41	97.93	97.93
	R _{FN} (%)	1.15	0.0	0.19	0.19
	R _{TN} (%)	96.44	0.0	82.13	82.13





(a) Originale.



(b) Contraffatta.

ACC=95.91 FNR=4.20 TNR=96.56

(c) Filtro *mihcak*, senza predizione.

ACC=96.59 FNR=0.16 TNR=68.62

(d) Filtro *mihcak*, con predizione.

ACC=98.18 FNR=0.84 TNR=89.77

(e) Filtro *bm3d*, senza predizione.

ACC=98.18 FNR=0.84 TNR=89.77

(f) Filtro *bm3d*, con predizione.

Figura 3.6 Esperimento sul denoising con il fotomontaggio delle persone.
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.





(a) Originale.



(b) Contraffatta.

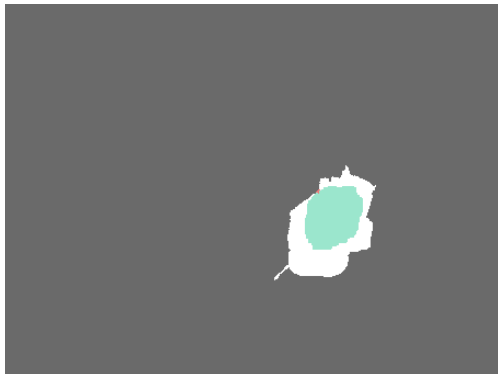
ACC=97.69 FNR=0.00 TNR=36.79

(c) Filtro *mihcak*, senza predizione.

ACC=97.69 FNR=0.00 TNR=36.79

(d) Filtro *mihcak*, con predizione.

ACC=97.94 FNR=0.00 TNR=43.60

(e) Filtro *bm3d*, senza predizione.

ACC=97.94 FNR=0.00 TNR=43.60

(f) Filtro *bm3d*, con predittore.

Figura 3.7 Esperimento sul denoising con il fotomontaggio delle navi.
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.





(a) Originale.



(b) Contraffatta.

ACC=98.59 FNR=1.15 TNR=96.44

(c) Filtro *mihcak*, senza predizione.

ACC=89.41 FNR=0.00 TNR=0.00

(d) Filtro *mihcak*, con predizione.

ACC=97.93 FNR=0.19 TNR=82.13

(e) Filtro *bm3d*, senza predizione.

ACC=97.93 FNR=0.19 TNR=82.13

(f) Filtro *bm3d*, con predizione.

Figura 3.8 Esperimento sul denoising con il fotomontaggio dei monoliti.
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.



3.3.2 ESPERIMENTI SULLA COMPRESSIONE JPEG

La Figura 3.9 illustra le maschere prodotte dall'algoritmo al variare della compressione JPEG (l'esperimento è condotto sul fotomontaggio delle persone). Le statistiche numeriche sono riportate nella Tabella 3.2.

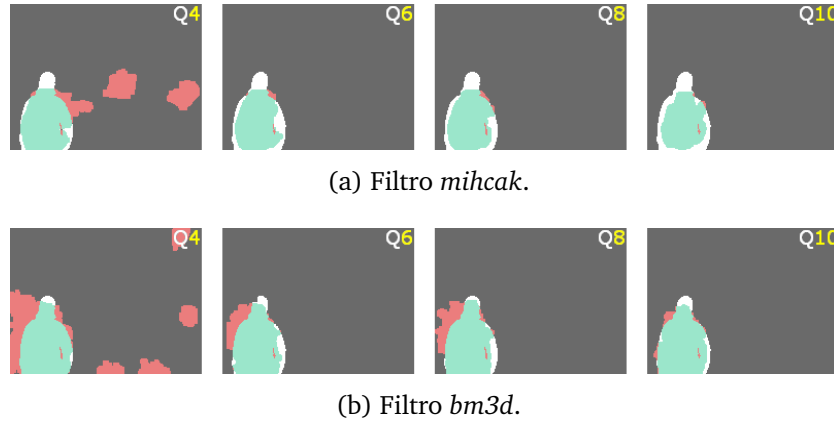


Figura 3.9 Esperimento sulla compressione JPEG.

Tabella 3.2 Esperimenti su JPEG: accuratezza e falso/vero negativo.

	Q	FOTO 1		FOTO 2	
		<i>mihcak</i>	<i>bm3d</i>	<i>mihcak</i>	<i>bm3d</i>
Acc (%)	4	91.70	89.39	76.11	73.25
	6	97.07	96.61	94.76	98.39
	8	97.17	96.26	97.63	98.16
	10	96.60	98.22	97.65	97.93
R _{FN} (%)	4	7.47	11.41	24.79	27.76
	6	0.27	2.31	4.26	0.51
	8	0.27	3.03	0.59	0.01
	10	0.15	0.78	0.00	0.00
R _{TN} (%)	4	84.51	96.34	100.00	100.00
	6	74.15	87.31	68.76	69.20
	8	75.17	90.20	50.60	49.64
	10	68.61	89.64	35.41	43.25



3.3.3 ESPERIMENTI CON LE RETI NEURALI

Una rete neurale è un sistema che apprende attraverso gli esempi. Un *esempio* è una coppia $(\mathbf{x}, \mathbf{y})$ che indica l'uscita desiderata $\mathbf{y} = (y_1, \dots, y_k)$ in corrispondenza dell'ingresso $\mathbf{x} = (x_1, \dots, x_m)$. Per diventare operativa, la rete neurale deve essere *addestrata* su un insieme di esempi $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)$, in modo da apprendere la relazione tra ingresso e uscita.

Una rete neurale consiste di tanti *neuroni* interconnessi tra loro. Il neurone è una relazione del tipo $y = f(\omega^T \mathbf{x})$, dove $f(\cdot)$ è la *funzione di attivazione*. La forma più semplice di rete neurale è il cosiddetto *perceptrone*. Esso è composto da un singolo strato di neuroni, ognuno dei quali è collegato a tutti gli ingressi e una singola uscita delle rete. L'addestramento consiste nel trovare, per ogni neurone, il vettore $\hat{\omega}$ che minimizza l'errore quadratico medio tra gli ingressi e la rispettiva uscita. Applicando il metodo del *gradiente discendente*, si ottiene un algoritmo iterativo chiamato *regola delta*:

$$\Delta \omega = \eta \cdot \sum_{n=1}^N \mathbf{x}_n \cdot f'(\omega^T \mathbf{x}_n) \cdot (y_n - f(\omega^T \mathbf{x}_n))$$

Il perceptrone è un sistema di tipo lineare. Una rete non lineare, invece, è il *perceptrone multi-livello*. Esso è formato da due (o anche più) strati di neuroni: gli ingressi sono collegati a tutti i neuroni del primo livello, detto *strato nascosto*, le cui uscite sono collegate a tutti i neuroni del secondo livello, detto *strato d'uscita*, che a sua volta è collegato alle uscite della rete. L'addestramento è realizzato con l'algoritmo *back-propagation*: (1) si valuta l'uscita della rete, (2) si calcola l'errore, (3) si modificano i pesi dello strato d'uscita, e infine (4) si modificano i pesi dello strato nascosto.

La rete neurale usata negli esperimenti (figure da 3.10 a 3.12) è un perceptrone multi-livello a 4 ingressi e 2 uscite. Gli ingressi $\mathbf{x} = (\rho, x_i, x_t, x_u)$ sono la correlazione e gli attributi su blocchi 128×128 , mentre le uscite $y_0 = P(H_0 | \mathbf{x})$ e $y_1 = P(H_1 | \mathbf{x})$ sono le probabilità a posteriori delle ipotesi. La maschera dei pixel contraffatti si costruisce ponendo $y_1 < 0.01$, in modo da mantenere la probabilità dei falsi negativi al di sotto del 1%.





(a) Originale.



(b) Contraffatta.

ACC=96.59 FNR=0.16 TNR=68.62

(c) Filtro *mihcak*, test d'ipotesi.

ACC=96.80 FNR=2.78 TNR=93.13

(d) Filtro *mihcak*, rete neurale.

ACC=98.18 FNR=0.84 TNR=89.77

(e) Filtro *bm3d*, test d'ipotesi.

ACC=98.49 FNR=0.65 TNR=91.09

(f) Filtro *bm3d*, rete neurale.

Figura 3.10 Esperimento con le reti neurali (fotomontaggio delle persone).
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.





(a) Originale.

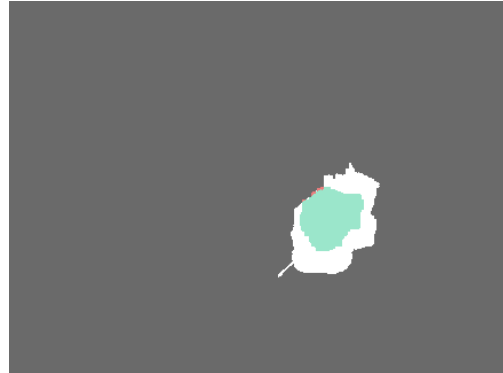


(b) Contraffatta.

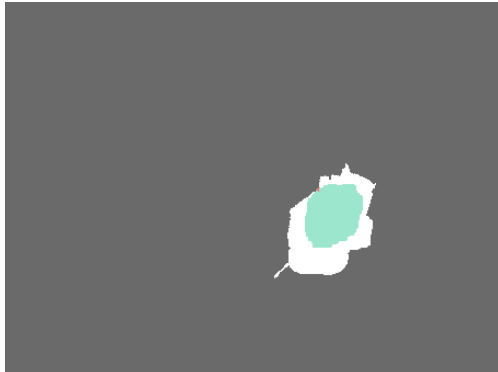
ACC=97.69 FNR=0.00 TNR=36.79

(c) Filtro *mihcak*, test d'ipotesi.

ACC=97.91 FNR=0.02 TNR=43.24

(d) Filtro *mihcak*, rete neurale.

ACC=97.94 FNR=0.00 TNR=43.60

(e) Filtro *bm3d*, test d'ipotesi.

ACC=98.87 FNR=0.30 TNR=76.96

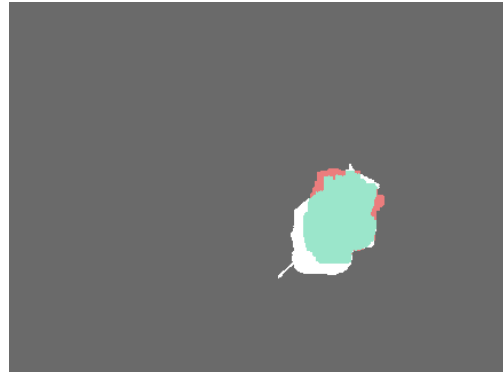
(f) Filtro *bm3d*, rete neurale.

Figura 3.11 Esperimento con le reti neurali (fotomontaggio delle navi).
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.





(a) Originale.



(b) Contraffatta.

ACC=89.41 FNR=0.00 TNR=0.00

(c) Filtro *mihcak*, test d'ipotesi.

ACC=97.54 FNR=0.56 TNR=81.48

(d) Filtro *mihcak*, rete neurale.

ACC=97.93 FNR=0.19 TNR=82.13

(e) Filtro *bm3d*, test d'ipotesi.

ACC=97.10 FNR=0.02 TNR=72.80

(f) Filtro *bm3d*, rete neurale.

Figura 3.12 Esperimento con le reti neurali (fotomontaggio dei monoliti).
 Legenda: TN = verde, FN = rosso, TP = nero, FP = bianco.
 Maschera reale = verde + bianco. Risultato = verde + rosso.





BIBLIOGRAFIA

- [1] A. BOVIK. *The Essential Guide to Image Processing*. Academic Press (Imprint of Elsevier), giugno 2009. (Citato a pagina 25).
- [2] A. BUADES, B. COLL e J. M. MOREL. “A Review of Image Denoising Algorithms, with a new one”. In: *SIAM Interdisciplinary Journal on Multiscale Modeling and Simulation* 4.2 (2005), pagine 490–530. (Citato a pagina 35).
- [3] M. CHEN, J. FRIDRICH, M. GOLJAN e J. LUKAS. “Determining Image Origin and Integrity Using Sensor Noise”. In: *IEEE Transaction on Information Forensics and Security* 3.1 (marzo 2008), pagine 74–90. (Citato alle pagine ii, 10, 12–16, 22).
- [4] K. DABOV, A. FOI, V. KATKOVNIK e K. EGIAZARIAN. “Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering”. In: *IEEE Transactions on Image Processing* 16.8 (agosto 2007), pagine 2080–2095. (Citato a pagina 36).
- [5] D. L. DONOHO e I. M. JOHNSTONE. “Ideal Spatial Adaptation by Wavelet Shrinkage”. In: *Biometrika* 81.3 (settembre 1994), pagine 425–455. (Citato a pagina 31).
- [6] A. FOI, M. TRIMECHE, V. KATKOVNIK e K. EGIAZARIAN. “Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data”. In: *IEEE Transaction on Image Processing* 17.10 (ottobre 2008), pagine 1737–1754. (Citato a pagina 6).

- [7] J. FRIDRICH. “Digital Image Forensics Using Sensor Noise”. In: *IEEE Signal Processing Magazine* 26.2 (marzo 2009), pagine 26–37. (Citato alle pagine 9, 10).
- [8] S. P. GHAEI, A. M. SAYEED e R. G. BARANIUK. “Improved Wavelet Denoising via Empirical Wiener Filtering”. In: *Proceedings of SPIE Technical Conference on Wavelet Applications in Signal Processing*. Volume 5. San Diego, CA, USA, 1997, pagine 300–303. (Citato a pagina 33).
- [9] R. C. GONZALEZ e R. E. WOODS. *Digital Image Processing*. 3^a edizione. Prentice Hall, agosto 2007. (Citato a pagina 24).
- [10] G. GONZÁLEZ-FARÍAS, J. A. DOMÍNGUEZ MOLINA e R. M. RODRÍGUEZ-DAGNINO. “Efficiency of the Approximated Shape Parameter Estimator in the Generalized Gaussian Distribution”. In: *IEEE Transaction on Vehicular Technology* 58.8 (ottobre 2009), pagine 4214–4223. (Citato a pagina 18).
- [11] M. D. GROSSBERG e S. K. NAYAR. “Modeling the Space of Camera Response Functions”. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 26.10 (ottobre 2004), pagine 1272–1282. (Citato a pagina 8).
- [12] G. E. HEALEY e R. KONDEPUDY. “Radiometric CCD Camera Calibration and Noise Estimation”. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 16.3 (marzo 1994), pagine 267–276. (Citato alle pagine 2, 4–7).
- [13] S. M. KAY. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, aprile 1993. (Citato alle pagine 12, 23).
- [14] S. M. KAY. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice-Hall, febbraio 1998. (Citato alle pagine 15, 16).
- [15] J. S. LIM. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, settembre 1989. (Citato a pagina 29).



- [16] C. LIU, R. SZELISKI, S. B. KANG, C. L. ZITNICK e W. T. FREEMAN. “Automatic Estimation and Removal of Noise from a Single Image”. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 30.2 (febbraio 2008), pagine 299–314. (Citato a pagina 8).
- [17] J. LUKAS, J. FRIDRICH e M. GOLJAN. “Digital Camera Identification from Sensor Pattern Noise”. In: *IEEE Transaction on Information Forensics and Security* 1.2 (giugno 2006), pagine 205–214. (Citato alle pagine 8–10, 18).
- [18] M. K. MIHÇAK, I. KOZINTSEV, K. RAMCHANDRAN e P. MOULIN. “Low Complexity Image Denoising Based on Statistical Modeling of Wavelet Coefficients”. In: *IEEE Signal Processing Letters* 6.12 (dicembre 1999), pagine 300–303. (Citato a pagina 32).
- [19] J. NAKAMURA. *Image Sensor and Signal Processing for Digital Still Camera*. CRC Press (Imprint of Taylor & Francis Group), agosto 2005. (Citato a pagina 2).





INDICE DELLE FIGURE

1	Falso di Abramo Lincoln, 1860.	vi
2	Falso del gen. Grant, 1864.	vi
3	Falso del gen. Blair, 1865.	vii
4	Falso di Mussolini, 1942.	vii
5	Falso di Browder e Tydings, 1950.	vii
6	Falso di Oprah Winfrey, 1989.	viii
7	Falso di Martha Stewart, 2005.	viii
8	Falso di John Kerry e Henry Fonda, 2004.	ix
9	Falso del militare che intima ai civili di abbassarsi, 2003. . .	x
10	Falso dei bombardamenti, 2006.	x
1.1	Proprietà fisiche del silicio.	2
1.2	Cattura dei colori.	3
1.3	Funzionamento del sistema ottico.	4
1.4	Proprietà delle componenti di rumore.	7
1.5	Modello lineare di rumore.	8
1.6	Impronta digitale di una fotocamera.	12
1.7	Artefatti presenti nell'impronta digitale.	13
1.8	Test d'ipotesi.	16
1.9	Gaussiana generalizzata.	17
1.10	Test d'ipotesi.	19
1.11	Stima delle ipotesi.	20

1.12	Modello adattativo dell'ipotesi H_1	21
1.13	Errore di predizione.	23
2.1	Correlazione tra i pixel adiacenti.	25
2.2	Filtraggio locale.	26
2.3	Filtro di Wiener.	28
2.4	Funzionamento dello shrinkage in due passi.	33
2.5	Funzionamento dell'algoritmo BM3D.	36
2.6	Passi intermedi dell'algoritmo BM3D.	37
2.7	Denoising dell'immagine "Lena" ($\sigma_n = 20$).	39
2.8	Residui di rumore dell'immagine "Lena" ($\sigma_n = 20$).	40
2.9	Denoising dell'immagine "Barbara" ($\sigma_n = 20$).	41
2.10	Residui di rumore dell'immagine "Barbara" ($\sigma_n = 20$).	42
3.1	Stima sperimentale del PRNU.	46
3.2	Spettro del PRNU.	46
3.3	Stima sperimentale delle ipotesi.	48
3.4	Risultati dell'addestramento.	50
3.5	Grafici scatter degli attributi.	51
3.6	Esperimento sul denoising (fotomontaggio delle persone).	57
3.7	Esperimento sul denoising (fotomontaggio delle navi).	58
3.8	Esperimento sul denoising (fotomontaggio dei monoliti).	59
3.9	Esperimento sulla compressione JPEG.	60
3.10	Esperimento con le reti neurali (falso delle persone).	62
3.11	Esperimento con le reti neurali (falso delle navi).	63
3.12	Esperimento con le reti neurali (falso dei monoliti).	64





INDICE ANALITICO

- Cattura, 3
 - a mosaico, 3
 - a strati, 3
- Color Filter Array, 4
- Demosaicking, 4, 8
- Denoising, 11, 24
 - bm3d, 36, 48, 56
 - mihcak, 32, 48, 56
- Foto-ricettore, 2, 4
 - uscita amplificata, 6
 - uscita flat-fielding, 7
 - uscita ideale, 2
 - uscita non uniforme, 5
 - uscita rumorosa, 6
- Modello di rumore, 6
 - lineare, 8
 - non lineare, 8
- Predittore
 - addestramento, 23
 - attributi, 22
- PRNU, 9
 - proprietà, 10
 - stima ML, 12
 - zero mean, 13
- Programmi MATLAB
 - addestramento predittore, 52
 - stima del PRNU, 46
 - stima delle ipotesi, 48
 - verifica d'integrità, 53
- Residuo di rumore, 11
- Sensore d'immagini, 2
 - corrente al buio, 5
 - prnu, 5
 - rumore fotonico, 5
- Test d'ipotesi, 15
 - correlazione, 16
 - gaussiana generalizzata, 17
 - metodo dei momenti, 18