

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Ingegneria Civile, Chimica, Ambientale e dei Materiali

*CORSO DI LAUREA MAGISTRALE
IN
INGEGNERIA PER L'AMBIENTE E IL TERRITORIO*

TESI DI LAUREA

in

GEOMATICA

**Applicazione in ambito termico
di un algoritmo di super risoluzione**

CANDIDATO
Francesco Corsini

RELATORE:
Chiar.mo Prof. Stefano Gandolfi

CORRELATORI:
Dott.Ing Luca Tavasci
Dott.Ing. Emanuele Mandanici

Anno Accademico 2016/2017

Sessione II

1. INTRODUZIONE	1
2. ALGORITMI DI SUPER RISOLUZIONE	4
2.1. <i>Struttura delle immagini digitali</i>	4
2.1.1. <i>Immagini termiche</i>	5
2.2. <i>Classificazione degli algoritmi di super risoluzione</i>	6
2.3. <i>Modello di immagine</i>	7
2.4. <i>Algoritmi di super risoluzione esistenti</i>	9
2.4.1. <i>Algoritmi del dominio di frequenza</i>	9
2.4.2. <i>Algoritmi del dominio spaziale</i>	10
2.5. <i>Registrazione delle immagini</i>	12
2.5.1. <i>Registrazione geometrica</i>	12
2.5.2. <i>Stima del blur</i>	21
2.6. <i>Tecniche di interpolazione spaziale</i>	22
2.6.1. <i>Nearest Neighbour (NN)</i>	23
2.6.2. <i>Inverse Distance Weighting (IDW)</i>	23
2.6.3. <i>Least Square Plane fitting (LSP)</i>	24
2.7. <i>Osservazioni LR</i>	25
3. TEST PRELIMINARI	27
3.1. <i>Generazione delle osservazioni a bassa risoluzione</i>	29
3.2. <i>Registrazione delle osservazioni LR</i>	30
3.2.1. <i>Cross correlazione normalizzata</i>	30
3.2.2. <i>Registrazione automatica intensity-based</i>	34
3.3. <i>Proiezione delle immagini su una griglia a risoluzione aumentata</i>	35
3.3.1. <i>Proiezione e media dei digital number</i>	35
3.3.2. <i>Proiezione di immagini esplose</i>	36
3.4. <i>Interpolazione tramite Inverse Distance Weighting</i>	38
3.5. <i>Verifiche di qualità delle immagini ricostruite</i>	42
3.5.1. <i>Confronto tra combinazioni di algoritmi</i>	42
3.5.2. <i>Filtro di smoothing</i>	47
3.5.3. <i>Posizione dei pixel nelle immagini esplose</i>	49
3.5.4. <i>Numero di immagini necessarie al processo di ricostruzione</i>	49
4. APPLICAZIONI	54
4.1. <i>Immagini digitali in scala di grigi</i>	54

4.1.1. <i>Canon EOS 6D</i>	54
4.1.2. <i>Fotogrammi da video in bassa risoluzione</i>	57
4.2. <i>Verifica di registrazione</i>	60
4.2.1. <i>Test chi quadro</i>	62
4.2.2. <i>Disuguaglianza di Čebyšëv</i>	67
4.2.3. <i>Verifiche seriali su set di immagini</i>	68
4.3. <i>Osservazioni termiche in bassa risoluzione</i>	72
5. CONCLUSIONI	77
6. FONTI BIBLIOGRAFICHE E SITOGRAFIA	80
APPENDICE A	81

1. INTRODUZIONE

Il passaggio dalla fotografia tradizionale alla fotografia digitale ha introdotto significativi cambiamenti sul livello di qualità ottenibile in una foto. La qualità delle prime immagini digitali infatti non era assolutamente paragonabile a quella delle immagini tradizionali, che rispetto alle prime presentavano un livello di dettaglio di gran lunga superiore. Fortunatamente l'evoluzione tecnologica ha portato a un netto miglioramento nel campo della fotografia digitale, unendone la praticità con la qualità di un'immagine tradizionale. Sfortunatamente, l'avvento della fotografia digitale non ha modificato il costo di una buona apparecchiatura che, anzi, si è visto aumentare.

A parte la fruizione della fotografia come semplice passatempo o lavoro, l'ottenimento di immagini di alta qualità riveste un aspetto fondamentale in numerosi altri campi di applicazione. La qualità di un'immagine, soprattutto per applicazioni tecniche, è strettamente legata alla risoluzione spaziale di quest'ultima, in altre parole la dimensione dell'impronta del singolo pixel sulla superficie osservata (ad esempio una risoluzione di 1 cm significa che il singolo pixel registra l'energia proveniente da un'area di $1 \times 1 \text{ cm}^2$). Per tale motivo la risoluzione spaziale di un'immagine dipende sia dal numero di detector che compongono il sensore, ma anche dalla distanza di presa e dalle caratteristiche dell'ottica.

Normalmente le immagini ad alta risoluzione si ottengono utilizzando sensori di grandi dimensioni e ad elevata precisione, in cui il numero di pixel è aumentato riducendo la dimensione delle celle fotosensibili, e quindi dei pixel stessi, permettendo di rappresentare una stessa scena con un numero maggiore di informazioni. Tali sensori, come anticipato, sono però molto costosi e in più vi sono anche delle limitazioni ottiche sulle dimensioni raggiungibili dei pixel, dato che un'eccessiva riduzione delle loro dimensioni porta a una diminuzione della quantità di luce che raggiunge le celle fotosensibili all'interno del sensore stesso, generando il cosiddetto "rumore" di immagine, ovvero pixel con una luce anomala nel loro contesto. In più, i pixel di piccole dimensioni sono più sensibili ad eventuali effetti di diffrazione della luce rispetto a quelli di grandi dimensioni.

Fondamentale è che l'implementazione di pixel di dimensioni minori porta alla necessità di avere una maggiore quantità di energia luminosa per unità di superficie per impressionare il singolo pixel (ad esempio, in condizioni di scarsa illuminazione è necessario aumentare l'ISO equivalente della macchina fotografica, simulando in tal modo di avere pixel a dimensioni maggiori e sgranando di conseguenza l'immagine).

Un ulteriore problema è rappresentato dalla compressione dei dati grezzi delle immagini (*raw data*) in vari protocolli lossy, come JPG, necessari a ridurre lo spazio richiesto per l'archiviazione e la velocità di trasmissione. Molti protocolli di compressione portano alla perdita di informazioni, dato che l'immagine viene divisa in blocchi poi processati indipendentemente, creando in tal modo problemi di continuità e distorsioni tra un blocco e l'altro a seguito della decompressione (l'effetto è ben visibile nel caso di immagini con sfondi chiari o scuri omogenei salvate in JPG, che presentano delle discontinuità proprio sulle porzioni omogenee). Parte di questo problema si può aggirare utilizzando protocolli di compressione intrinsecamente più pesanti ma robusti, generalmente detti lossless.

Come introdotto, il bisogno di avere immagini con la migliore risoluzione spaziale possibile riguarda vari settori, come imaging aereo o da satellite, sistemi di sorveglianza, imaging medico, scienze forensi e molti altri. È proprio per questi settori che sono stati sviluppati algoritmi di varia natura per approcciare il problema, volti al miglioramento della qualità di un'immagine, mantenendo allo stesso tempo un costo relativamente limitato.

La tipologia di algoritmi sviluppati è molto ampia; i più comuni sono quelli di interpolazione, rigenerazione e *rendering* di immagini, che presentano però delle limitazioni sulla qualità raggiungibile. Di tutte le tecniche, una in particolare ha attirato l'attenzione degli operatori dei settori sopra citati grazie alle sue potenzialità, la Super Risoluzione (da ora SR). Il concetto alla base è quello della fusione o "registrazione" di numerose immagini a bassa risoluzione (da ora LR) che descrivano la stessa scena, a meno di scostamenti tra l'una e l'altra a livello sub-pixel, per ottenere un'unica immagine ad alta risoluzione (da ora HR) contenente tutti i particolari delle immagini di partenza. Ciò significa che le immagini LR possono essere ottenute con una fotocamera spostata leggermente di volta in volta oppure considerando i frame di un video in lieve movimento che ritragga l'oggetto in esame. La necessità di scostamenti sub-pixel deriva dalla necessità che ogni immagine aggiunga informazioni rispetto alle altre; risulta infatti impossibile ottenere un aumento di risoluzione partendo da immagini identiche tra loro, dato che nessuna di queste aggiunge particolari rispetto alle altre.

Un settore estremamente condizionato dalla scarsa risoluzione spaziale delle immagini è quello delle analisi termografiche, nel quale vengono impiegate immagini ottenute da camere termiche. Queste ultime sono particolari camere sensibili alle radiazioni infrarosse, che generano mappe di temperatura delle superfici inquadrare. I sensori utilizzati in tali strumenti si differenziano da quelli tradizionali proprio a causa della diversa banda dello spettro

elettromagnetico da acquisire. L'energia che i corpi emettono nella banda dell'infrarosso termico è notevolmente inferiore a quella tipicamente riflessa nella banda del visibile: conseguenza è che i sensori delle camere termiche, a parità di altri fattori, devono avere celle (pixel) di dimensione maggiore rispetto a quelle dei comuni sensori delle fotocamere digitali. Quindi, per natura costruttiva dei sensori stessi, la risoluzione spaziale di un'immagine termica, anche se appartenente alla classe di "alta definizione", è di molto inferiore a quella ottenibile con una comune fotocamera. In quanto tecnica di analisi non distruttiva, ma allo stesso tempo molto efficace, la termografia risulta essere di grande importanza in molti ambiti, come quello medico o quello edile.

Se da un lato è vero che la ricerca sui sensori termici prosegue incessantemente e consente di un progressivo miglioramento della risoluzione spaziale delle termo-camere, dall'altro rimane il fatto che questi strumenti sono molto costosi, per cui non si può pensare di sostituirli frequentemente. Rimane in ogni caso il fatto che una maggiore risoluzione spaziale è auspicabile in molte applicazioni termografiche, soprattutto quando l'aspetto geometrico degli oggetti misurati è rilevante per il risultato finale.

Per questi motivi, nella presente tesi di laurea ci si è occupati di analizzare, implementare ed applicare algoritmi di SR ad immagini termiche. Dopo un'attenta analisi bibliografica volta a comprendere l'approccio generale alla base della SR, sono stati sviluppati in ambiente MATLAB una serie di algoritmi in grado di produrre immagini a risoluzione aumentata partendo da set di immagini di risoluzione inferiore. Particolare attenzione è stata rivolta all'analisi della qualità delle immagini HR ricostruite. In particolare, si è fatto inizialmente uso di *dataset* sintetici di immagini al fine di testare le performance dei diversi algoritmi, passando solo in un secondo momento alle applicazioni reali, sia nel campo del visibile sia nel campo delle immagini termiche.

2. ALGORITMI DI SUPER RISOLUZIONE

In questo capitolo viene introdotto il concetto di super risoluzione, descrivendone i principi di base e riportando alcuni dettagli riguardo alcuni degli algoritmi disponibili che implementano tale approccio. L'idea alla base è quella di ottenere un'immagine a risoluzione aumentata partendo da un insieme di immagini a risoluzione inferiore. Noti gli *shift* tra le varie immagini a bassa risoluzione a livello sub-pixel, la metodologia SR punta a utilizzare tutte le informazioni contenute al loro interno per ottenere un'unica immagine somma di esse.

Una situazione idealizzata di algoritmo SR è quella riportata in figura 2.1, in cui le immagini (b)-(c)-(d) sono traslate rispetto ad (a) rispettivamente di mezzo pixel in orizzontale, verticale e diagonale, senza alcuna rotazione o fattore di scala. Noti gli spostamenti di mezzo pixel per ogni immagine, queste sono proiettate su un'unica griglia a risoluzione raddoppiata lungo le due direzioni, costruendo di fatto una nuova immagine HR contenente tutti i particolari delle osservazioni LR.

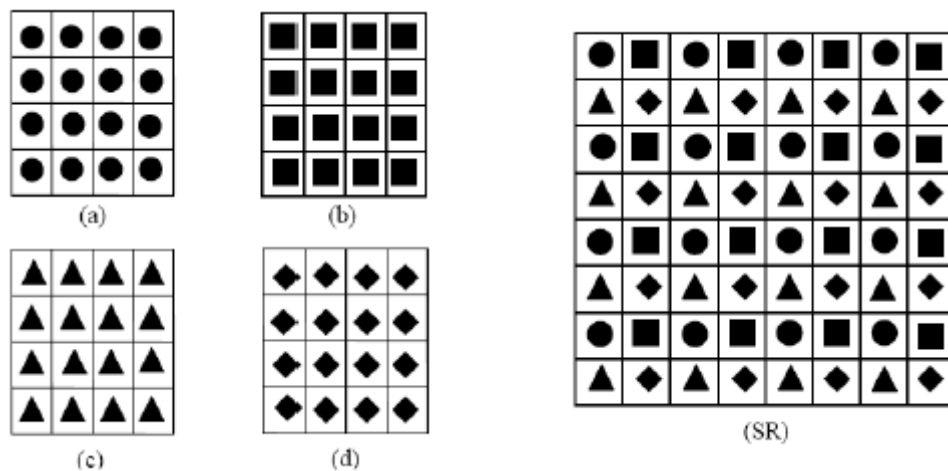


Fig. 2.1 – *Concetto della Super Risoluzione*

2.1. Struttura delle immagini digitali

Per comprendere al meglio il problema è necessaria una breve trattazione sulle immagini digitali e su cosa queste effettivamente siano. In passato un'immagine veniva ottenuta utilizzando una pellicola fotografica impressionata dalla radiazione luminosa catturata dall'obiettivo della camera. La differenza tra un'immagine ottica e un'immagine digitale sta nel processo di digitalizzazione e nell'entità della discretizzazione che ne deriva. La fotografia digitale è infatti un procedimento per l'acquisizione di immagini proiettate attraverso un

sistema ottico su un sensore sensibile alla luce, con successiva conversione in formato digitale e immagazzinamento su un supporto di memoria. Diversamente, nella fotografia ottica le immagini sono direttamente proiettate su una pellicola con una grana molto più fitta rispetto alla discretizzazione delle immagini digitali.

L'immagine digitale risulta quindi da una discretizzazione, più grossolana rispetto a quella della fotografia tradizionale, di una scena continua nello spazio che viene campionata e i cui valori di intensità sono quantizzati su un numero finito di livelli. Il campionamento è effettuato dalle celle fotosensibili presenti all'interno del sensore, che rilevano la radiazione luminosa e vi assegnano un valore di intensità.

Il risultato dell'operazione è una matrice di numeri di dimensioni $M \times N \times K$, dove M è il numero di righe di pixel, N di colonne e K rappresenta gli strati di colore da sovrapporre a seconda del modello utilizzato.

I modelli di colore sono modelli matematici per la rappresentazione numerica dei colori che tipicamente, a seconda della complessità, utilizzano tre o quattro componenti cromatiche. RGB (Red, Green, Blue) è il modello più utilizzato e per ogni strato rende disponibili valori di intensità interi da 0 a 255, con 8 bit/pixel. Il numero totale di colori a disposizione è dunque pari a $256 \times 256 \times 256$, a seconda delle combinazioni dei tre strati. Il modello di colore più semplice è invece la scala di grigi, o anche detta "bianco e nero", in cui l'immagine è descritta da una singola matrice di dimensione $M \times N$, al cui interno gli interi descrivono l'intensità del bianco (0 è nero assoluto e 255 è bianco).

2.1.1. Immagini termiche

Prima di procedere oltre è bene porre l'attenzione sulla struttura delle immagini termiche, che risultano essere di natura diversa da quelle tradizionali.

Un'immagine termica è il risultato della cattura della radiazione infrarossa emessa dai soggetti e dall'ambiente inquadrati da parte di una termocamera. Diversamente dalla fotografia tradizionale, ciò che viene catturato non è quindi la luce appartenente allo spettro visibile, ma la radiazione infrarossa, correlata alla temperatura dei soggetti stessi. La termografia permette infatti di visualizzare in matrici bidimensionali (in analogia con le immagini in scala di grigi) una misura dell'irraggiamento dei soggetti presenti. Ciò che viene registrato all'interno della matrice di un'immagine termica è la temperatura dei corpi

analizzati, ottenuta attraverso la misurazione dell'intensità di radiazione infrarossa che questi emettono. Tali misurazioni sono effettuabili su qualsiasi corpo emetta radiazione infrarossa, quindi con una temperatura superiore allo zero assoluto, indipendentemente dalla loro illuminazione nel campo del visibile. La quantità di energia emessa da un corpo q è proporzionale alla quarta potenza della temperatura assoluta T di un corpo, secondo la legge di Stefan-Boltzmann:

$$q = \sigma T^4$$

in cui σ è la costante di Stefan-Boltzmann e T è la temperatura assoluta del corpo emittente. Misurando la radiazione emessa è dunque possibile risalire alla temperatura del corpo emittente e registrarla all'interno della matrice costituente l'immagine.

2.2. Classificazione degli algoritmi di super risoluzione

Gli algoritmi per la super risoluzione possono essere classificati in base a diversi fattori, come il dominio in cui operano, il numero di immagini LR in esame e il metodo di ricostruzione utilizzato.

Partendo dal dominio, esso definisce la tipologia di approccio che si adotta nei confronti delle immagini; il dominio di frequenza considera le immagini come insieme di diverse frequenze, mentre il dominio spaziale opera direttamente sui campioni dell'immagine. Dei due, il dominio di frequenza è stato il punto di partenza per i primi algoritmi SR, che in tempi più recenti si sono però orientati verso il dominio spaziale, in cui è possibile collocarne la maggior parte.

Per quanto riguarda il numero di immagini LR utilizzate vi sono due classi: algoritmi a singola immagine (*single-image based*) e algoritmi a immagini multiple (*multiple-image based*). Gli algoritmi nel dominio delle frequenze si collocano praticamente tutti nella seconda categoria, mentre quelli nel dominio spaziale si suddividono in entrambe. Nel caso *multiple-image based*, si considera che vi siano degli scostamenti geometrici sub-pixel e fotometrici tra le osservazioni LR a disposizione e l'immagine HR che si vuole ottenere. In questi algoritmi, detti *reconstruction-based*, si vogliono individuare tali scostamenti per ricostruire l'immagine finale. Con tali condizioni la soluzione delle equazioni è più semplice, ma ogni immagine LR impone un set di vincoli ai valori di intensità dei pixel nella griglia HR, rendendo necessario un enorme numero di osservazioni LR per ottenere grandi aumenti di risoluzione. Per tale

motivo, in applicazioni che richiedano incrementi di risoluzione molto elevati sono stati sviluppati gli algoritmi *single-image based*, anche conosciuti come “*example-based super resolution*” o “*image hallucination*”. In essi le corrispondenze tra porzioni di immagini LR e HR sono dedotte e applicate tramite database di coppie di immagini LR e HR campione, generalmente con un fattore di scala pari a due. Ripetendo il procedimento è possibile ottenere incrementi di scala maggiori e ricreare dettagli altrimenti non catturati. In tal modo viene superato il limite sull’incremento di risoluzione, ma il rischio è che i dettagli, inizialmente sconosciuti, dedotti dal procedimento non siano effettivamente corrispondenti con la realtà.

2.3. Modello di immagine

Nella pratica un algoritmo di SR cerca di eliminare limiti tecnici, problematiche e disturbi introdotti durante l’operazione di campionatura e digitalizzazione delle immagini. Si introduce quindi un modello d’immagine rappresentativo della logica di un algoritmo SR, che descriva il procedimento con cui le immagini osservate LR vengono ottenute. Nel caso più semplice, è possibile considerare in modello lineare come segue:

$$g(m, n) = \frac{1}{q^2} \sum_{x=qm}^{(q+1)m-1} \sum_{y=qn}^{(q+1)n-1} f(x, y)$$

in cui g è un’osservazione LR, f è la scena HR originale, q è il fattore di riduzione della risoluzione (*downsampling*) che compare durante l’operazione di campionamento, uguale lungo entrambe le direzioni, x e y sono le coordinate nella griglia HR e m e n sono le coordinate nella griglia LR. La risoluzione spaziale delle immagini LR è pari a $M_1 \times M_2$, mentre quella dell’immagine HR è $N_1 \times N_2$, dove $N_1 = qM_1$ e $N_2 = qM_2$. Il concetto dell’equazione sopra descritta è che un singolo pixel di un’immagine LR sia ottenuto mediando le intensità dei pixel HR all’interno di un riquadro di dimensioni $q \times q$.

È possibile rendere il modello più realistico introducendovi anche altri parametri che concorrono durante il campionamento. A questo proposito si definiscono nuovi concetti:

- **blurring**, ovvero la sfocatura che si può generare in un’immagine, ad esempio a causa del movimento, dato che il processo di cattura è molto breve, ma non istantaneo, e quindi certi elementi possono muoversi al suo interno;
- **warping**, deformazioni di un’immagine a bassa risoluzione rispetto ad una successiva, dovute a traslazioni, rotazioni, fattori di scala e variazioni di prospettiva;

- **noise**, o rumore, un'interferenza nel segnale luminoso che colpisce le celle fotosensibili del sensore, che porta ad avere pixel con intensità anomale.

L'inclusione di tali elementi all'interno del modello d'immagine porta, per ogni immagine LR k-esima, a

$$g_k(m, n) = d \left(h_k \left(w_k (f(x, y)) \right) \right) + \eta_k(m, n)$$

dove w è la funzione di *warping*, h di *blurring*, d è l'operatore di *downsampling* (definisce come l'immagine LR è campionata rispetto all'HR) e η è il rumore d'immagine. In forma matriciale, l'ultima equazione può essere espressa come

$$g = Af + \eta$$

dove A rappresenta il fattore di degradazione, contenente al suo interno le funzioni appena elencate, eccetto il rumore. Quanto appena descritto è rappresentato visivamente in figura 2.2.

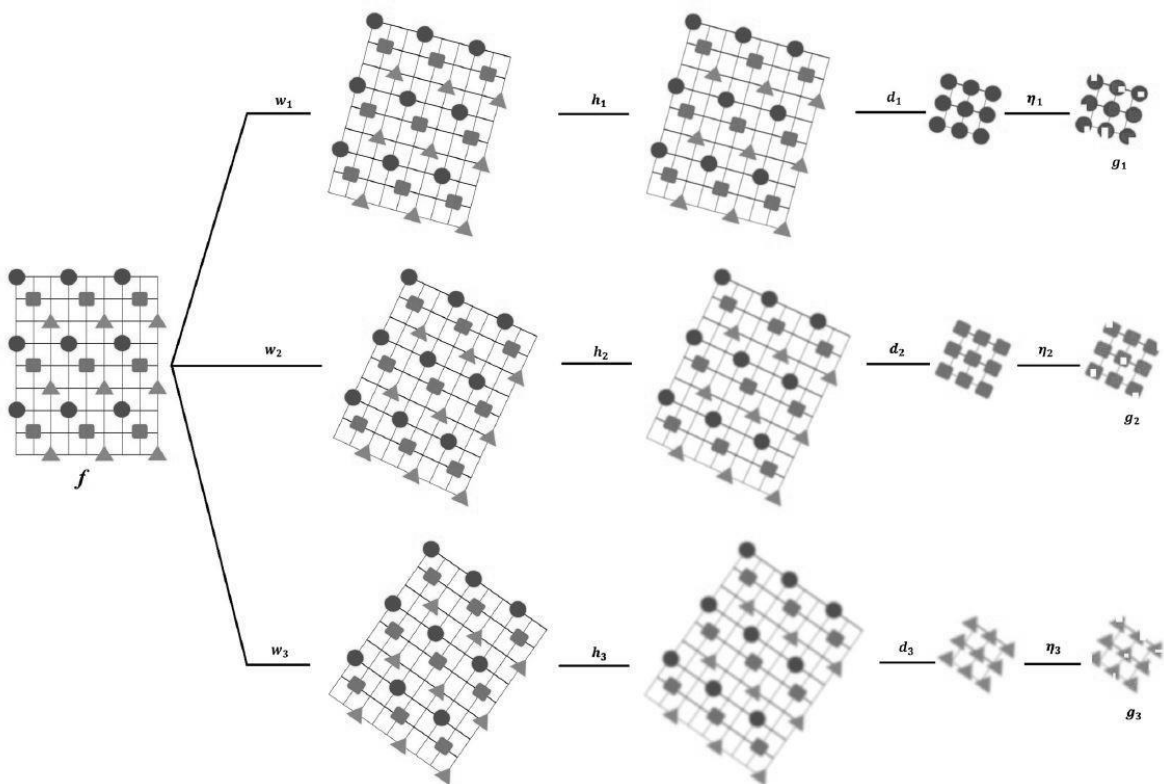


Fig. 2.2 – Schema di ottenimento di un'osservazione LR

La funzione di *warping* racchiude al proprio interno ogni trasformazione esistente tra l'osservazione LR e la scena HR. Ad esempio, se LR è spostato di un vettore (a, b) e ruotato di un angolo θ rispetto ad HR, la funzione di *warping* sarà pari a:

$$w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} x \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

A seconda del tipo di trasformazioni che intercorrono la matrice assume forme diverse.

Nella funzione di *blurring* compaiono vari possibili effetti, da quelli dovuti al sistema ottico fino ad eventuali effetti atmosferici o ad un movimento nella scena.

2.4. Algoritmi di super risoluzione esistenti

Per poter comprendere la mole di algoritmi esistenti, suddivisi tra dominio di frequenza e dominio spaziale, in questo paragrafo segue una breve trattazione dei due sottogruppi, con annesso un elenco degli algoritmi esistenti.

2.4.1. Algoritmi del dominio di frequenza

Gli algoritmi di questo gruppo convertono le matrici delle osservazioni LR nel dominio della frequenza, ottenendo così un segnale rappresentante l'immagine ad alta risoluzione da ricostruire. Infine quest'ultimo, ottenuto nel dominio di frequenza, è trasportato al dominio spaziale e riconvertito nell'immagine HR ricostruita. In base alla tipologia di trasformazioni impiegate per convertire le immagini in segnali all'interno del dominio di frequenza si possono distinguere due gruppi: metodi basati sulla trasformata di Fourier e metodi sulla trasformata *wavelet* (ondicelle). Il primo gruppo è storicamente il punto di partenza del filone della super risoluzione, utilizzato in algoritmi iterativi per estendere lo spettro dei segnali delle osservazioni oltre al limite di diffrazione per poi aumentarne la risoluzione. La prima applicazione di tale metodologia fu nell'ambito della ricostruzione di immagini acquisite dal satellite Landsat 4, che produceva set di immagini generalmente simili, ma traslate l'una dall'altra, della stessa porzione di superficie terrestre.

Il secondo gruppo di metodi rappresenta un'alternativa molto utilizzata alla trasformata di Fourier. Lo schema generale parte con la decomposizione dell'immagine input in sotto-immagini strutturalmente correlate, in modo da poter individuare somiglianze tra aree vicine. Ad esempio, l'osservazione LR è prima scomposta in sottobande e poi sia queste ultime che l'immagine ad alta frequenza originale sono utilizzate per migliorare la qualità delle

sottobande stesse interpolate. L'immagine HR finale è poi generata combinando tutte le sottobande, tramite una trasformazione inversa alla prima.

2.4.2. Algoritmi del dominio spaziale

In questo paragrafo viene fornito un riassunto degli algoritmi di super risoluzione del dominio spaziale (i *reconstruction based*), in particolare di quelli basati sull'utilizzo di immagini multiple, tralasciando gli algoritmi di allucinazione basati su una singola immagine. Questi ultimi presentano infatti caratteristiche estremamente diverse dai primi e sono utilizzati per ricostruire classi particolari di immagini che presentano fra loro statistiche più o meno comuni a prescindere dal soggetto, come ad esempio i volti.

2.4.2.1. Iterative Back Projection (IBP)

I metodi appartenenti a questa categoria sono i primi sviluppati all'interno del dominio spaziale. Definito il modello spaziale come al paragrafo 2.3, questi metodi sono volti alla minimizzazione del valore di $\|Af - g\|_2^2$ (par. 2.3). Per farlo viene generalmente effettuata per prima cosa una stima di partenza dell'immagine HR, per poi rifinirla. Tale stima di primo tentativo è effettuata registrando preliminarmente le osservazioni LR su una griglia HR, per poi farne una media. Per rifinire tale immagine di partenza l'equazione del modello di immagine viene utilizzata per simulare un set fittizio di osservazioni LR. Viene poi calcolato l'errore tra le osservazioni LR simulate e quelle reali, ripetendo iterativamente tutto il processo fino a convergenza (rappresentata da un criterio di accettazione), ottenendo in tal modo i parametri da inserire nel modello d'immagine utilizzato poi per rifinire la prima stima di immagine HR.

2.4.2.2. Iterative Adaptive Filtering (IAF)

L'IAF è stato sviluppato per la specifica applicazione nell'ambito video, dato che permette di generare video ad alta risoluzione partendo da video LR. La sua applicazione parte dall'utilizzo di filtri specifici di tale campo e tratta la ricostruzione come un problema di stima dei *frame* HR.

2.4.2.3. Metodi diretti

Dato un set di osservazioni LR, i primi algoritmi di questa tipologia seguono una serie di step, che iniziano dalla scelta di un'immagine, fra tutte, come riferimento. Le altre immagini sono poi registrate rispetto ad essa secondo l'algoritmo ritenuto più opportuno. Successivamente la risoluzione dell'immagine di riferimento è aumentata del fattore scelto e le osservazioni LR vengono registrate sopra ad essa utilizzando i parametri identificati in fase di registrazione. Poi, l'immagine HR è ottenuta dalla fusione di tutte le osservazioni LR, con un successivo ultimo step opzionale di *deblurring*. La fusione delle immagini è ottenuta tramite uno dei metodi di interpolazione elencati al paragrafo 2.6 od altri più o meno complessi. Questa tipologia di super risoluzione risulta più rapida rispetto al caso della IBP.

Al fine di ottenere una ricostruzione efficace, ma in tempi relativamente ristretti, non vi è una vera e propria canonicità nella classificazione dei metodi diretti di super risoluzione, poiché a causa degli elevati tempi computazionali richiesti per l'esecuzione di una singola ricostruzione i vari autori implementano modifiche all'interno dei codici per semplificare alcune fasi del processo e dunque ridurre i tempi necessari. Ad esempio, poiché spesso la fase più dispendiosa dal punto di vista computazionale è quella di registrazione, è possibile inserire come ipotesi di partenza che le trasformazioni tra le osservazioni LR di uno stesso set siano puramente traslazionali, con la necessità però di avere la certezza su tale ipotesi. In altri casi è possibile modificare la fase di proiezione dei pixel sulla griglia HR oppure eseguire parallelamente le operazioni di proiezione ed interpolazione.

2.4.2.4. Projection Onto Convex Sets (POCS)

La categoria POCS è un gruppo di metodi iterativi basati sulla definizione di una funzione di costo legata alla risoluzione del problema SR. Considerato il modello di immagine del paragrafo 2.3, viene assunto che ogni osservazione LR fornisca una conoscenza a priori della soluzione finale. Fornito il set di osservazioni LR correttamente elaborate, è possibile proseguire iterativamente con una fase di stima dell'immagine HR sconosciuta.

2.4.2.5. Metodi statistici

Un gruppo a parte di algoritmi di SR, basato sempre su più osservazioni LR, è rappresentato dai metodi statistici, come il *maximum likelihood* o il *maximum a posteriori* che introducono

al proprio interno funzioni di distribuzione della probabilità per giungere ad una stima probabilistica dell'immagine HR. Per tale categoria di metodi risulta fondamentale che il problema sia ben posto, che vi siano sufficienti osservazioni LR e che l'influenza di piccole variazioni sulle ipotesi non generino grandi cambiamenti dei risultati.

2.5.Registrazione delle immagini

Il processo di super risoluzione *reconstruction-based* è rivolto all'eliminazione, o almeno alla riduzione, degli effetti di cui sopra. Per poter ricreare i pixel mancanti in una griglia HR partendo da più immagini LR è necessario che queste ultime presentino tra loro delle differenze. Infatti, la ricostruzione è possibile solo se almeno uno dei parametri coinvolti nel modello di immagine differisce da un'immagine LR a un'altra, in modo tale che ogni immagine LR porti informazioni aggiuntive rispetto alle precedenti.

Proprio a causa di tali differenze, nella maggior parte degli algoritmi SR è necessaria una preliminare fase di ricomposizione, che si suddivide a sua volta in registrazione geometrica (o riposizionamento) e stima del *blur*.

2.5.1. Registrazione geometrica

In fase di registrazione geometrica vengono compensati i disallineamenti/movimenti tra le immagini LR, per poterle poi porre all'interno di una griglia HR comune. I disallineamenti sono il risultato di movimenti globali e locali: con i primi si intendono i movimenti di un oggetto o della camera, mentre i secondi sono quelli di natura non rigida, come un cambio di espressione facciale o il movimento delle foglie di una chioma.

La modellazione dei movimenti locali è particolare e complessa e in genere si basa sul determinare set di punti di controllo combinati tra loro tramite pesatura, mentre il movimento globale può essere schematizzato con la sovrapposizione di:

- un modello traslazionale;
- un modello affine;
- un modello proiettivo.

Negli algoritmi SR più semplici i modelli traslazionale e rotazionale tra la scena HR e le osservazioni LR sono del tipo:

$$x = x_k^t + q_x m \cos\theta_k - q_y n \sin\theta_k$$

$$y = y_k^t + q_x m \sin\theta_k - q_y n \cos\theta_k$$

in cui (x_k^t, y_k^t) è il vettore traslazione dell'immagine k-esima, θ_k è l'angolo di rotazione e q_x e q_y sono le frequenze di campionamento lungo x e y. Per trovare tali parametri, la bibliografia suggerisce di utilizzare gli sviluppi in serie di Taylor sulle immagini LR. Per farlo, si considerano due immagini LR g_1 e g_2 , traslate tra loro orizzontalmente e verticalmente rispettivamente di due quantità a e b e ruotate di un angolo θ :

$$g_2(m, n) = g_1(m \cos\theta - n \sin\theta + a, n \cos\theta + m \sin\theta + b)$$

Si sviluppano poi i termini $\sin\theta$ e $\cos\theta$ in serie di Taylor, fino al secondo termine:

$$g_2(m, n) = g_1\left(m + a - n\theta - \frac{m\theta^2}{2}, n + b + m\theta - \frac{n\theta^2}{2}\right)$$

Anche il termine g_1 è poi sviluppato in serie di Taylor fino al secondo termine:

$$g_2(m, n) = g_1(m, n) + \left(a - n\theta - \frac{m\theta^2}{2}\right) \frac{\partial g_1}{\partial m} + \left(b + m\theta - \frac{n\theta^2}{2}\right) \frac{\partial g_1}{\partial n}$$

Da tale espressione è poi possibile ricavare l'errore totale commesso nel processo di registrazione di tutte le immagini, che risulta pari a:

$$E(a, b, \theta) = \sum \left(g_1(m, n) + \left(a - n\theta - \frac{m\theta^2}{2}\right) \frac{\partial g_1}{\partial m} + \left(b + m\theta - \frac{n\theta^2}{2}\right) \frac{\partial g_1}{\partial n} - g_2(m, n) \right)^2$$

in cui la sommatoria è effettuata sull'area di sovrapposizione delle immagini. Il minimo dell'errore si ricava derivando rispetto ad a, b e θ e risolvendo il sistema così ottenuto rispetto ad essi.

Nelle applicazioni passate è stato dimostrato che il metodo, nella forma appena esposta, è valido e ben applicabile solo in caso di piccole traslazioni e rotazioni e per tale motivo è stato utilizzato come punto di partenza per lo sviluppo di algoritmi più fini ed efficaci.

Un altro dei punti deboli degli algoritmi di registrazione così redatti si trova nella quantità di tempo impiegato per la risoluzione di tutte le equazioni e i sistemi. Per ridurre la quantità di tempo impiegato alcuni autori suggeriscono di applicare un filtro di *smoothing* (una funzione di "pulizia dell'immagine") alle osservazioni LR prima di registrarle. Spesso la scelta ricade su un filtro gaussiano, tramite il quale ogni pixel dell'immagine viene rimpiazzato dalla media pesata dei pixel in un suo intorno, secondo una funzione di distribuzione gaussiana. Tramite

esso è possibile determinare di quanto effettivamente sfocare l'immagine, modificando di volta in volta la deviazione standard della campana da applicare. L'utilizzo di un filtro di questo tipo porta un triplice beneficio, in quanto esso riduce:

1. Bordi netti e piccoli dettagli, difficili da registrare efficacemente;
2. Bruschi cambiamenti nei valori di intensità dei pixel, quindi anche il rumore generato da artefatti ambientali, elettronici o altri effetti di disturbo;
3. Distorsioni introdotte durante il processo di compressione.

Considerando una geometria a simmetria circolare, la funzione di distribuzione di Gauss assume forma:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

dove x e y sono le distanze dal centro della gaussiana e σ è la deviazione standard della distribuzione.

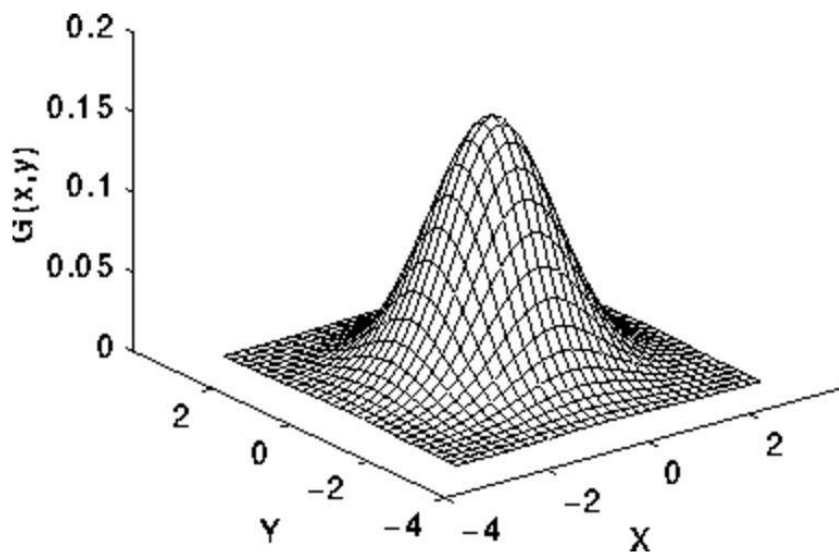


Fig. 2.3 – Curva gaussiana in tre dimensioni

Detta $G(x, y)$ l'intensità di pixel nella posizione (x, y) , il nuovo valore di intensità di questo è ottenuto tramite una media pesata dei pixel nell'intorno. Il pixel originale riceve il peso più alto, mentre quelli via via più lontani hanno peso minore.

Un esempio di risultato dell'applicazione del filtro appena descritto, con varianza 1.0 e media 0, è riportato nella figura 2.4, dove si può osservare come le distorsioni generate dalla

compressione in formato JPG sullo sfondo dell'immagine di sinistra risultino meno evidenti nell'immagine di destra.



Fig. 2.4 – Immagine prima (sinistra) e dopo (destra) l'applicazione di un filtro gaussiano

L'utilizzo di un filtro di *smoothing* permette anche di rendere quelli che sono grandi movimenti nell'immagine originale, movimenti di entità minore nelle immagini sfocate, in modo tale da poterle registrare più facilmente. Quanto appena descritto prende il nome di *optical flow* ed è una metodologia molto utile nel caso di immagini che abbiano al proprio interno oggetti non rigidi che compiano movimenti relativi tra loro, come ad esempio un viso con un cambio di espressione.

Per effettuare la registrazione geometrica di immagini vi sono a disposizione numerose tecniche, ognuna con i propri pro e contro, a seconda dell'applicazione:

- Estrazione di porzioni caratteristiche dell'immagine con successiva cross correlazione normalizzata (*Normalized Cross Correlation*, NCC). Si determinano preliminarmente zone comuni all'immagine di riferimento e al *target* contenenti elementi caratteristici, utilizzati poi come punti di riferimento per determinare le omografie tramite correlazione incrociata. Tale tecnica è utilizzata non solo nel campo delle immagini 2D, ma anche per immagini tridimensionali;
- Minimizzazione della somma del quadrato (*Sum of Squared Differences*, SSD) delle differenze sui bordi netti all'interno dell'immagine. Fornite un'immagine di riferimento e una *target*, esse sono divise in blocchi delle stesse dimensioni e ogni blocco dell'immagine di riferimento è confrontato con ogni blocco, entro un certo intorno, dell'immagine *target*. La corrispondenza migliore viene ricercata secondo

svariate possibili tecniche di minimizzazione e, trovata la migliore corrispondenza tra i blocchi, si ripete la procedura fino alla stima del vettore spostamento. Tale tecnica è poco efficace nel caso di immagini con grandi superfici di pixel con intensità simile;

- Modelli adattivi di similitudine per la registrazione di volti in sequenze di immagini ad alta frequenza;
- Utilizzo di algoritmi specifici per una prima stima del vettore spostamento tra immagine di riferimento e *target*. Successivamente vengono estratte delle regioni particolari nelle immagini per definire dei campi di spostamento, determinando similitudini e movimenti locali per raffinare il vettore di primo tentativo;
- Registrazione tramite trasformate di Fourier e trasformate discrete di Fourier;
- Registrazione automatica iterativa *intensity-based*;
- Selezione manuale di *control points* finalizzati a determinare i parametri della trasformazione, eventualmente anche non rigida.

A seconda della tipologia dell'immagine, dell'eventuale presenza di movimenti e della loro entità è possibile determinare la tecnica di registrazione più appropriata al contesto.

I metodi sopra descritti sono in generale globali, ovvero permettono di determinare campi di spostamento da applicare a tutti i pixel. In presenza di molteplici oggetti in movimento secondo vettori diversi all'interno della stessa scena, risulta necessario individuare metodi di registrazione specifici per il problema, che tengano conto anche di movimenti locali.

A seguire, una breve descrizione delle principali tecniche di registrazione delle immagini.

2.5.1.1. *Cross correlazione normalizzata*

La cross correlazione normalizzata è una particolare applicazione della cross correlazione.

Innanzitutto, per ottenere un risultato soddisfacente, è necessario selezionare porzioni caratteristiche delle immagini di riferimento e *target*, da confrontare tra loro. La cross correlazione è infatti una tecnica di ricerca degli *offset* per ogni punto di una porzione caratteristica dell'immagine *target* all'interno di un'immagine di riferimento, come riportato in figura 2.5.



Fig. 2.5 – Immagine di riferimento (sinistra) e finestra di ritaglio del target (destra)

La ricerca con passo pari ad un pixel rende la tecnica molto efficace; tuttavia, dato che ogni pixel viene processato di volta in volta, la correlazione risulta molto lenta, fattore che mal si addice alla necessità di uno *script* veloce. Il corretto posizionamento del *target* all'interno del *frame* di riferimento viene effettuato confrontando algebricamente i valori dei pixel presenti nel *target* e nel riferimento (per immagini in scala di grigio si considera l'unica matrice a disposizione, mentre nel caso di colori RGB, si focalizza l'attenzione su uno solo dei tre canali).

In figura 2.6 l'immagine di sinistra è il riferimento, mentre quella a destra il *target*. Sulla prima delle due è necessario scegliere una finestra di ricerca con cui confrontare il *target*, per ridurre la dimensione della matrice di correlazione e dunque il tempo computazionale.

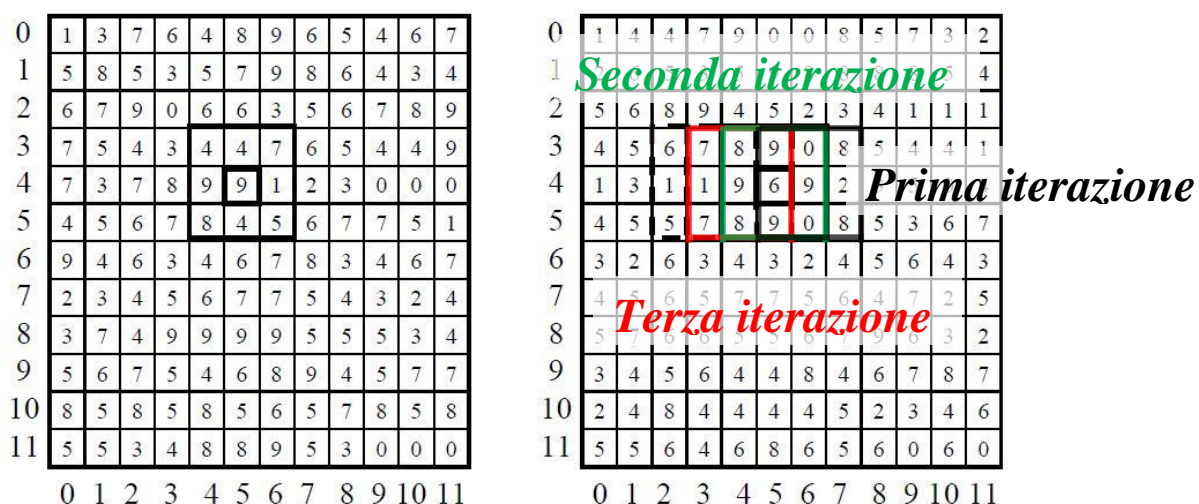


Fig. 2.6 – Matrici di immagini target e di riferimento con iterazioni di cross correlazione

Considerando una traslazione puramente orizzontale, un punto P di coordinate (x, y) preso sull'immagine di riferimento avrà un coniugato sul *target* di coordinate $(x - d, y)$, con d valore di *offset*. La ricerca della corretta posizione del *target* nel riferimento viene effettuata traslando di volta in volta l'ascissa del centro del primo all'interno del secondo. Una volta individuate tutte le possibili posizioni della finestra *target* all'interno della finestra di riferimento, la cross correlazione normalizzata determina la miglior corrispondenza possibile con:

$$C_{NCC}(x, y, d) = \max \left\{ \frac{\sum_{i,j} [I_1(x + i, y + j) - \bar{I}_1(x, y)] \cdot [I_2(x - d + i, y + j) - \bar{I}_2(x + d, y)]}{\sqrt{\{\sum_{i,j} [I_1(x + i, y + j) - \bar{I}_1(x, y)]^2 \cdot \sum_{i,j} [I_2(x - d + i, y + j) - \bar{I}_2(x + d, y)]^2\}}} \right\}$$

In essa gli indici i e j variano rispettivamente lungo la larghezza e l'altezza della finestra, $I_1(x, y)$ e $I_2(x, y)$ sono i valori di intensità del pixel (x, y) -esimo per la prima e la seconda immagine, d è il valore di *offset*, $\bar{I}_k(x, y)$, $k = 1, 2$, è la media, nella finestra delle dimensioni considerate, dei valori di intensità dei pixel dell'immagine k-esima.

La differenza tra cross correlazione e cross correlazione normalizzata sta nella normalizzazione, effettuata al denominatore, dei coefficienti di correlazione, che ricadono in tal modo tra 0 e 1 (normalmente l'intervallo sarebbe tra -1 e 1, ma nel presente caso i valori possono essere solo positivi, data la natura del *range* possibile della scala dei colori).

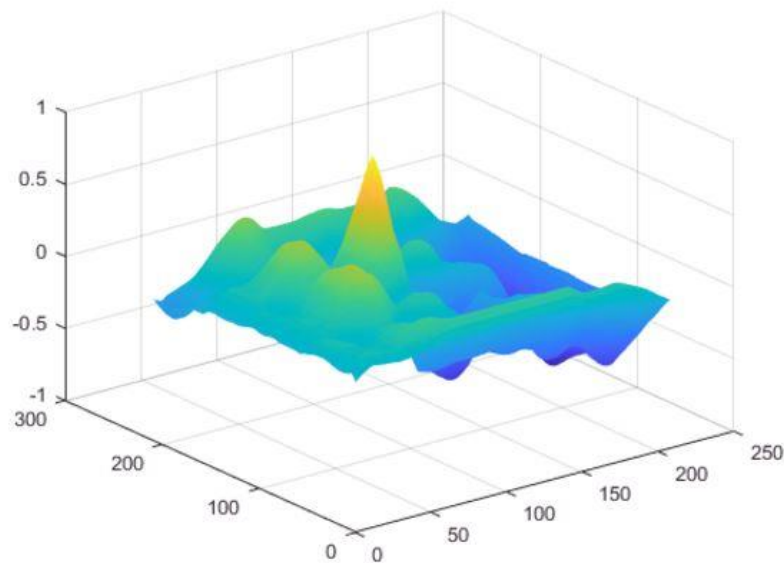


Fig. 2.7 – Esempio di picco di correlazione

Il risultato dell'operazione è una matrice di correlazione che presenta al proprio interno un picco, mostrato in figura 2.7, corrispondente alla corretta posizione del *target* all'interno del riferimento.

I parametri di *offset* che si ricavano per la finestra del *target* vengono poi applicati all'intera immagine *target* per ottenerne la registrazione.

2.5.1.2. Minimizzazione della somma delle differenze quadrate

Lo schema alla base di tale tecnica è lo stesso del caso della cross correlazione normalizzata. La differenza fra le due tecniche sta nell'operatore matematico utilizzato per trovare la miglior corrispondenza. Nel presente caso il criterio di ricerca è il seguente:

$$C_{SSD}(x, y, d) = \min \left\{ \sum_{i,j} [I_1(x + i, y + i) - I_2(x - d + i, y + j)]^2 \right\}$$

Il minimo di tale sommatoria indica la miglior corrispondenza tra *target* e riferimento e rappresenta l'*offset* da applicare all'intera immagine *target*.

Per poter utilizzare sia la cross correlazione normalizzata che la somma delle differenze quadrate è necessario introdurre preliminarmente una rotazione iterativa del *target* da registrare, dato che il prodotto di tali tecniche è un semplice coefficiente di correlazione, e non un intero set di parametri di registrazione.

2.5.1.3. Applicazione della trasformata discreta di Fourier (DFT)

Le trasformate di Fourier sono utilizzate in numerosi approcci per operare registrazioni accurate, considerando anche la presenza di piccoli *shift* sub-pixel. Generalmente le applicazioni sono successive ad una preliminare cross correlazione, per aumentare il grado di accuratezza nell'individuazione del picco ed effettuare una cosiddetta correlazione di fase.

Una tecnica del genere permette di registrare immagini in assenza di punti di controllo con una precisione fino a 0.01 pixel prendendo in considerazione trasformazioni di tipo rigido, ovvero traslazioni e rotazioni.

2.5.1.4. *Registrazione automatica intensity-based*

È la tipologia di registrazione utilizzata nel caso di algoritmi automatici, come nel caso delle funzioni matlab, esemplificata in figura 2.8.

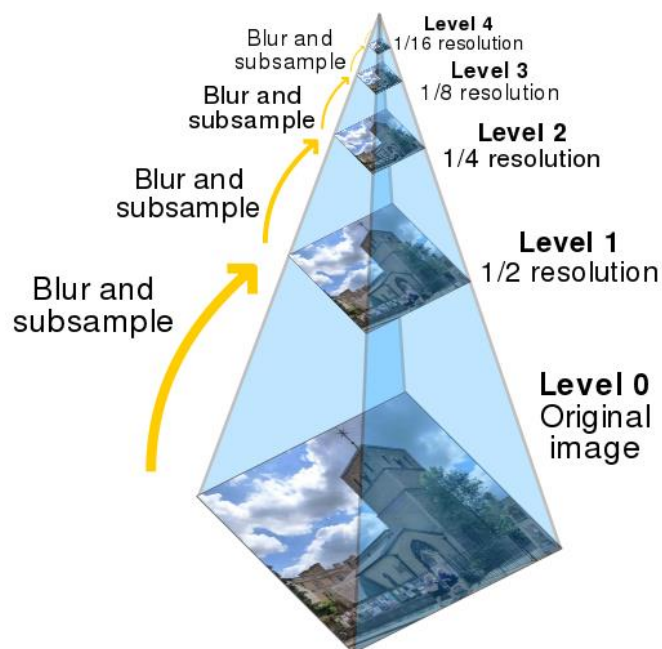


Fig. 2.8 – *Registrazione tramite piramide*

La risoluzione delle immagini da registrare è diminuita per ogni livello della piramide di un fattore pari a due in entrambe le direzioni. La registrazione tra riferimento e *target* viene poi effettuata, partendo dal livello a risoluzione più bassa, sfruttando svariate tecniche di interpolazione o registrazione specifiche. Il ciclo viene ripetuto per ogni livello della piramide, aumentando di volta in volta l'accuratezza del processo di registrazione e ottenendo alla fine una registrazione ottimale in tempi relativamente brevi.

2.5.1.5. *Registrazione tramite control points*

Per registrare due immagini con tale metodologia è necessario selezionare manualmente il maggior numero possibile di punti di controllo comuni ad entrambe le immagini, imbastendo un tal modo un set di equazioni lineari (con incogniti, ad esempio, fattori di traslazione, angolo di rotazione e fattore di scala). Si procede poi alla ricerca dei migliori parametri di trasformazione spaziale utilizzando tecniche di minimizzazione, per poi applicarli all'intera immagine ed ottenerne la registrazione. La necessità di selezionare manualmente caratteristiche comuni alle immagini (linee, vertici, spigoli e superfici) rende tale metodo

molto efficace, ma allo stesso tempo ne rappresenta il punto debole, dato che la selezione manuale rende impossibile la completa automatizzazione del processo e quindi impraticabile l'utilizzo nel caso di un elevato numero di immagini LR da registrare.

Per quanto appena descritto, questa tecnica si differenzia da quelle precedenti, appartenenti al dominio *intensity-based*, in cui la registrazione è effettuata comparando schemi di intensità comuni alle immagini *target* e di riferimento.

Definita la tecnica più appropriata, la stima dei movimenti globali tra tutte le immagini LR può essere effettuata secondo due modelli, uno differenziale ed uno cumulativo. Nel primo caso, meno utilizzato, i parametri di movimento vengono determinati per ogni coppia successiva di immagini, mentre nel secondo una singola immagine fra tutte è presa come riferimento e i parametri delle altre immagini sono calcolati relativamente a essa. Scegliendo una sola immagine come riferimento, è fondamentale che questa non presenti un eccessivo rumore e che non vi siano eccessivi disturbi, dato che una scelta errata renderebbe inefficace l'intero processo di super risoluzione. A tal fine, negli algoritmi più rigorosi vengono introdotti dei controlli di qualità sulle immagini LR per definire oggettivamente quale di esse sia maggiormente rappresentativa.

2.5.2. *Stima del blur*

La stima del *blur* è un procedimento volto alla compensazione della sfocatura presente all'interno delle osservazioni LR, con lo scopo di eliminare tale effetto dall'immagine SR obiettivo dell'algoritmo. Nella maggior parte dei casi gli effetti di sfocatura sono causati da imperfezioni intrinseche alla macchina fotografica, da effetti atmosferici o dal movimento. Per quanto riguarda il primo punto, nella pratica l'effetto è modellato da una cosiddetta *Point Spread Function* (PSF), generalmente una distribuzione di Gauss con valori standard di errore quadratico medio (regione 3×3 con $\sigma = 0.4$, 5×5 con $\sigma = 1.0$, etc.), e reso noto dal produttore. In caso contrario, essa può essere stimata con varie tecniche, a seconda della disponibilità o meno della macchina fotografica.

Il *blurring* introdotto dal movimento è fortemente dipendente dalla direzione e dalla velocità del movimento e dal tempo di esposizione.

In alcuni algoritmi conosciuti come metodi diretti, il processo di stima di una PSF è sostituito dall'applicazione di un filtro passo alto (un particolare filtro) successiva alla ricostruzione dell'immagine.

2.6. Tecniche di interpolazione spaziale

Il passo finale del processo di super risoluzione consiste, per la maggior parte degli algoritmi moderni, nell'interpolazione spaziale di tutti i dati a disposizione, per generare un'immagine della risoluzione desiderata. Il set di dati a disposizione per questa fase consiste in tutte le osservazioni a bassa risoluzione registrate all'interno dello stesso sistema di riferimento. A causa degli spostamenti intrinseci al processo di ottenimento descritto, determinati in fase di registrazione, ogni pixel delle immagini a bassa risoluzione occupa una posizione leggermente diversa rispetto agli altri e il primo passo per l'interpolazione consiste nel porre ogni pixel nella sua posizione registrata all'interno di una griglia delle dimensioni dell'immagine HR finale. Successivamente è possibile procedere con l'interpolazione vera e propria, che permette di ottenere le informazioni mancanti all'interno della griglia sfruttando i valori di intensità dei pixel a disposizione attorno a quelli vuoti. Il problema dell'interpolazione è dunque tridimensionale (la terza dimensione è l'intensità del pixel) e i dati a disposizione assumono forma $\{(x_i, y_i, I_i)\}_{i=1}^n$, dove $I_i = I_i(x_i, y_i)$ rappresenta l'intensità, o meglio il *digital number* (DN), associata ai pixel "noti". Sulla base di questi è necessario stimare i valori di intensità di tutti i pixel incogniti all'interno della matrice. A tale scopo è possibile individuare una serie di tecniche, basate tutte su un criterio di medie pesate dei dati a disposizione. Tutte queste hanno in comune la stessa formula di applicazione che ha forma, secondo la notazione già utilizzata:

$$\hat{I}(x_0) = \sum_{i=1}^n \lambda_i I(x_i)$$

In essa il termine $\hat{I}(x_0)$ rappresenta il valore di intensità stimato nella posizione x_0 di interesse, mentre λ_i è il peso i-esimo da applicare al valore di intensità noto del pixel i-esimo. Ciò che cambia tra una tecnica di interpolazione spaziale e un'altra è il modo in cui il peso viene calcolato.

2.6.1. *Nearest Neighbour (NN)*

Il metodo NN determina il valore di intensità da attribuire ad un pixel in cui questo è incognito in base al valore di intensità del pixel più vicino, secondo un criterio di vicinanza. Quest'ultimo è espresso tracciando gli assi delle congiungenti i centri dei vari pixel a intensità nota, generando in tal modo delle griglie di poligoni V_i (tipo Voronoi) come quella in figura 2.9.

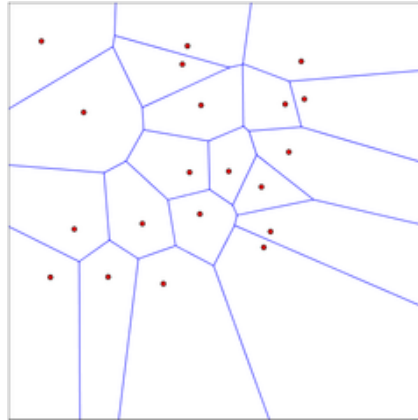


Fig. 2.9 – Griglia tipo Voronoi per interpolazione NN

In questo modo viene tracciato un poligono per ogni pixel noto della griglia. La stima del valore di intensità incognito da attribuire al pixel in esame è effettuata considerando la posizione in cui quest'ultimo si trova: il valore di intensità assegnato è infatti uguale a quello del pixel centro del poligono in cui cade il pixel incognito. Riassumendo, il peso da attribuire ad ogni pixel ad intensità nota è:

$$\lambda_i = \begin{cases} 1 & \text{se } x_i \in V_i \\ 0 & \text{se } x_i \notin V_i \end{cases}$$

Il limite dell'utilizzo di tale tecnica nel caso di una griglia di pixel è che in un intorno circolare di raggio unitario di un pixel incognito è possibile trovare otto diversi valori di intensità cui, tecnicamente, dovrebbe essere assegnato lo stesso peso. Per tale motivo, si preferisce ricorrere a tecniche più raffinate.

2.6.2. *Inverse Distance Weighting (IDW)*

Diversamente dalla tecnica NN questo metodo di interpolazione e il successivo forniscono una stima del valore di intensità del pixel incognito utilizzando una combinazione dei valori di intensità dei pixel in un proprio intorno. In particolare il metodo IDW attribuisce a tali pixel

pesi inversamente proporzionali alla distanza tra questi e il pixel incognito, centrato nell'intorno. L'ipotesi alla base è che il valore di intensità incognito sia più simile a quello dei pixel a lui vicini piuttosto che a quelli più distanti e dunque i pesi degli n pixel ad intensità nota nell'intorno possono essere espressi nella forma:

$$\lambda_i = \frac{1/d_i^P}{\sum_{i=1}^n 1/d_i^P}$$

dove d_i è la distanza tra la posizione del pixel incognito e quella del pixel i -esimo noto e P è un parametro di potenza.

Grazie a tale forma il valore dei pesi attribuiti diminuisce all'aumentare della distanza, specialmente all'aumentare del valore del parametro di potenza. Il valore di quest'ultimo serve infatti per enfatizzare o meno l'importanza dei pixel più vicini, in modo da rendere la stima più o meno locale: ad un valore superiore ad uno di P i pesi maggiori vengono attribuiti ai pixel immediatamente adiacenti a quello incognito, mentre al contrario i pesi sono più distribuiti sulle distanze. La scelta di tale parametro è arbitraria; se esso assume valore pari a 0 si effettua una cosiddetta "media mobile", se pari a 1 si effettua un'interpolazione lineare e se superiore ad 1 un'interpolazione non lineare .

2.6.3. *Least Square Plane fitting (LSP)*

Il principio del LSP si basa sull'assunzione che per tre punti nello spazio passi uno ed uno solo piano che li contiene tutti. Per poter porre correttamente il problema e utilizzare in modo appropriato questa tecnica il campione di dati a disposizione, nella forma $\{(x_i, y_i, I_i)\}_{i=1}^n$, deve contenere tre o più valori di intensità noti, unitamente alle coordinate all'interno della matrice. Dal set di osservazioni nell'intorno del pixel x_0 incognito è possibile determinare i valori A, B e C tali che il piano di equazione $I = Ax + By + C$ approssimi al meglio i valori di intensità dei campioni noti, in modo tale che la somma degli errori quadrati tra le intensità reali I_i e i valori stimati sul piano \hat{I}_i sia la minima possibile. Tale errore quadrato è il seguente:

$$E(A, B, C) = \sum_{i=1}^n [(Ax_i + By_i + C) - I_i]^2$$

Il minimo di tale funzione si ottiene in corrispondenza del valore del gradiente $\nabla E = (0,0,0)$. Si giunge in tal modo a un sistema di tre equazioni lineari nelle incognite A, B e C risolvibile. In particolare:

$$(0,0,0) = \nabla E = 2 \sum_{i=1}^n [(Ax_i + By_i + C) - I_i](x_i, y_i, 1)$$

Il risultato del sistema così calcolato è dato dai parametri A, B e C che meglio approssimano il campione a disposizione, generando un piano del tipo di quello indicato in figura 2.10. Introducendo all'interno dell'equazione del piano le coordinate del pixel incognito si ricava il corrispettivo valore di intensità stimata.

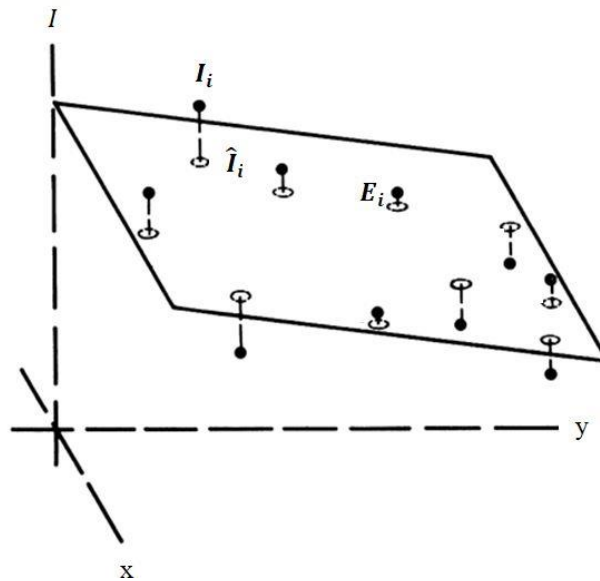


Fig. 2.10 – Interpolazione LSP di pixel a intensità nota

2.7.Osservazioni LR

Il numero di immagini a bassa risoluzione necessarie alla ricostruzione di un'immagine SR è fortemente dipendente dalle condizioni di immagine e dall'incremento di risoluzione che si vuole ottenere.

Se il rumore di immagine nelle osservazioni LR è in generale basso e se vi è un'ottima conoscenza degli *shift* sub-pixel tra le varie immagini, l'unico limite sull'aumento di risoluzione è rappresentato dal numero di osservazioni a disposizione. Tale osservazione giustifica la necessità di avere un algoritmo di registrazione stabile, accurato ed efficace. La motivazione è che in tal modo viene ridotta l'influenza di errori grossolani commessi durante il procedimento di registrazione che si andrebbero a sommare ad eventuali vizi già presenti nel set di partenza. Molteplici autori suggeriscono che il numero minimo di osservazioni LR necessarie, in condizioni ottimali di rumore e difetti, sia pari al quadrato dell'aumento di

risoluzione che si vuole ottenere. Ad esempio, volendo ottenere un aumento di risoluzione pari a 4 lungo ogni direzione ($1 \text{ pixel LR} = 4 \times 4 \text{ pixel HR}$), il numero minimo di osservazioni LR a disposizione è di 16.

In condizioni standard, ovvero considerando la presenza di un certo disturbo dovuto a svariate cause, è bene avere a disposizione immagini aggiuntive per minimizzarne l'influenza. In più la distribuzione degli *shift* sub-pixel nelle varie direzioni dovrebbe essere il più omogenea possibile.

La ricostruzione di un'immagine con un numero insufficiente di osservazioni LR non è impossibile, ma sicuramente non porta all'ottenimento di risultati ottimali, specie per alti valori del fattore di aumento della risoluzione. In più, quest'ultimo non può essere grande a piacere, dato che ad un suo incremento lineare corrisponde un incremento quadratico del numero di osservazioni LR necessarie. Queste devono essere rispondenti ad un criterio di differenza di una rispetto all'altra e la sensibilità nell'individuazione degli *shift* sub-pixel può non essere sufficiente nel caso questi siano eccessivamente piccoli.

3. TEST PRELIMINARI

All'interno di questo capitolo è riportata la fase della sperimentazione volta alla determinazione di una combinazione di algoritmi e funzioni ottimale per il processo di super risoluzione. Successivamente, l'attenzione viene posta sui parametri che concorrono ad una buona riuscita dell'intero procedimento, come ad esempio il valore del filtro di *pre-processing*, il numero minimo e sufficiente di osservazioni LR da utilizzare, etc. Tutte le valutazioni vengono effettuate sulla base di criteri statistici, per mantenere un certo livello di oggettività dell'intero procedimento.

Di tutte le tipologie di algoritmi SR esistenti, per affrontare il problema della scarsa qualità delle immagini termiche, che generalmente hanno una risoluzione di 480×640 pixel, l'approccio ritenuto più adatto è quello di un metodo diretto *reconstruction-based*. In più, volendo comprendere al meglio il problema della super risoluzione e le difficoltà ad esso legate, la realizzazione di un nuovo algoritmo è stata ritenuta preferibile all'utilizzo di un algoritmo già scritto da altri.

Il primo ostacolo da affrontare è rappresentato dalla scelta del linguaggio in cui scrivere il programma, fra tutti quelli a disposizione. Data la natura matriciale degli oggetti da trattare e la relativa semplicità di utilizzo e scrittura di uno *script*, la scelta del *software* è ricaduta su MATLAB, nativo MathWorks[®], ed in particolare sulla sua *release* R2017. La scelta si giustifica anche a causa di una *toolbox* specifica per l'elaborazione di immagini digitali (*Image Processing Toolbox*) ed una per il *processing* di segnali (*Signal Processing Toolbox*), entrambe rispondenti a necessità che si sarebbero incontrate durante la redazione dell'algoritmo. Grazie ad esse, è infatti possibile implementare su MATLAB una serie di nuove funzioni e *routine* specifiche per i campi di utilizzo, non presenti originariamente all'interno del *software*. Ad esempio, nonostante le operazioni svolte siano tutte a livello matriciale, è necessario verificare visivamente i risultati ottenuti tramite immagini. In assenza di un pacchetto di funzioni specifico, MATLAB non permette di trasformare una o più matrici rispettivamente in un'immagine in scala di grigi o a colori, mentre tramite la *toolbox* specifica ne consente la visualizzazione, permettendo controlli e verifiche più immediati.

Definiti gli strumenti, il passo successivo consiste nella pianificazione dei test da eseguire, per ottenere un algoritmo completo e funzionante. I test non sono stati effettuati immediatamente su immagini termiche a bassa risoluzione, proprio a causa della loro natura che le rende scarsamente adatte a eseguire test semplici (le matrici hanno al proprio interno valori di

temperatura decimali invece che valori interi tra 0 e 255, rendendo gli algoritmi più lenti e più difficoltosa la visualizzazione). L'idea è quella di eseguire i primi test su osservazioni LR "artificiali", ovvero generate degradando immagini ad alta risoluzione. In tal modo si ha a disposizione l'immagine ad alta risoluzione originale, con cui confrontare il risultato ottenuto dall'applicazione dell'algoritmo e quantificare quanto le due immagini siano tra loro simili. L'esecuzione di tale confronto è possibile solo in questa fase, dato che nei set di osservazioni LR reali non è presente un termine di confronto HR e quindi il risultato è da accettare in ogni caso, a meno di errori grossolani chiaramente identificabili.

Una volta conclusi i test sulle immagini generate, la combinazione di algoritmi e i parametri corretti da utilizzare determinati in questa fase sono stati applicati ad osservazioni LR ottenute da fotocamera, senza la possibilità di misurare con precisione gli spostamenti e le rotazioni della camera rispetto alla scena. Infine, successivamente ad un'ulteriore fase di raffinazione e redazione di algoritmi accessori, è stato possibile passare alla vera e propria ricostruzione di immagini termografiche a bassa risoluzione.

Dovendo agire in un campo relativamente nuovo e poco conosciuto, le parti costituenti l'algoritmo sono state redatte e testate singolarmente di volta in volta, ottenendo in tal modo una buona stabilità di calcolo e sicurezza del risultato. L'intero algoritmo per i test preliminari si compone delle seguenti *routine*:

- 1) Generazione di osservazioni a bassa risoluzione;
- 2) Registrazione delle osservazioni LR, tramite cross correlazione normalizzata o tramite routine MATLAB di registrazione *intensity-based*;
- 3) Proiezione su una griglia a risoluzione aumentata;
- 4) Interpolazione delle osservazioni, tramite media aritmetica, LSP o IDW;

Ognuna delle *routine* elencate necessita dei risultati delle precedenti su cui lavorare; per tale motivo un'esigenza importante delle prove effettuate è il salvataggio delle immagini durante tutto il percorso cui vengono sottoposte, anche per poterne tenere traccia e svolgere verifiche successive. Per non perdere dati o comprometterne la visualizzazione, risulta fondamentale determinare quale sia il protocollo migliore per la loro memorizzazione. Il formato di salvataggio inizialmente usato durante l'esecuzione degli algoritmi è stato il JPEG, per una questione di leggerezza. In tal modo si ricadeva però in errore, dato che tale formato porta alla perdita di dati e alla modifica degli stessi all'interno del *file*. Per tale motivo tutti gli algoritmi sono stati modificati nel formato di salvataggio delle informazioni utilizzando file di testo

ASCII in formato .txt contenenti le intere matrici calcolate, con l'eccezione dell'immagine HR ricostruita da salvare in formato TIFF per poterla visualizzare.

3.1. Generazione delle osservazioni a bassa risoluzione

La logica del modello d'immagine viene presa a riferimento per la creazione delle osservazioni a bassa risoluzione "artificiali" da fondere per ottenere il risultato finale. Partendo da un'immagine ad alta risoluzione è infatti possibile, dopo aver applicato *shift* e rotazioni casuali e aver mediato le intensità dei pixel HR, ottenere da essa numerose immagini a bassa risoluzione diverse l'una dall'altra. In particolare, per effettuare la prima serie di test è stata scelta un'immagine di partenza a risoluzione 4032×3024 pixel a colori, da ridurre con un fattore pari a 4. La risoluzione finale delle osservazioni LR è dunque pari a 1008×756 pixel, il cui valore di intensità è il risultato della ricombinazione dei valori dei pixel HR all'interno di una finestra 4×4 . Lo *script* genera inizialmente due valori interi casuali compresi tra -4 e 4, per introdurre traslazioni lungo l'asse x e l'asse y, e un valore casuale razionale di angolo compreso tra -4° e 4° , del quale l'osservazione LR sarà ruotata rispetto all'immagine di partenza. Le trasformazioni appena citate vengono applicate nel sistema di riferimento della griglia HR, che ha la propria origine nel pixel in alto a sinistra. Il processo di degradazione dell'immagine HR è riportato in figura 3.1.

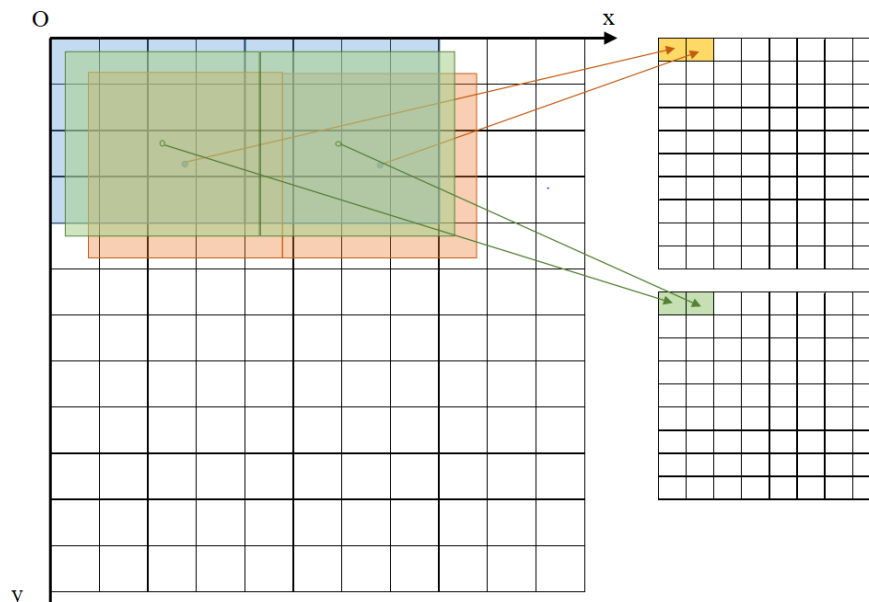


Fig. 3.1 – Immagine HR di partenza (sinistra) ed esempi di osservazioni LR generate (destra)

Dalla posizione di partenza la griglia è traslata in orizzontale e verticale e successivamente ruotata dei valori casuali generati, per portare disomogeneità tra le varie osservazioni LR. L'intensità del singolo pixel LR è ottenuta assegnando un peso ad ogni pixel della finestra 4×4 a seconda della distanza dal centro e sommando poi tutte le intensità dei pixel HR moltiplicate per i rispettivi pesi. In tal modo, a seconda dei valori di traslazione e rotazione, ogni osservazione LR risulta diversa da tutte le altre.

Per poter svolgere verifiche successive sul funzionamento degli algoritmi di registrazione e per effettuare dei confronti tra l'immagine HR di partenza e quella ricostruita, dato che le osservazioni LR sono registrate rispetto ad una di riferimento e che quindi l'immagine HR ricostruita sarà orientata come quest'ultima, i parametri di traslazione e rotazione per ogni immagine generata vengono salvati su un apposito file di testo, da cui recuperarli in caso di necessità.

Volendo ottenere come risultato finale un'immagine a risoluzione quadruplicata lungo le due direzioni, secondo quanto affermato al paragrafo 2.6, il numero minimo di osservazioni LR da generare è pari a

$$(\textit{Reduction_Factor})^2 = 16$$

ma volendo introdurre ridondanza nel problema ne vengono utilizzate il doppio, quindi 32.

Le immagini così generate vengono salvate in formato TIFF, in modo da non perdere alcuna informazione al loro interno.

3.2.Registrazione delle osservazioni LR

Per la registrazione delle immagini sono stati utilizzati due approcci differenti, che sfruttano uno la cross correlazione normalizzata e l'altro una registrazione automatica *intensity-based* tramite *routine* MATLAB.

3.2.1. Cross correlazione normalizzata

Fra le tecniche a disposizione, per eseguire la registrazione, la prima scelta è ricaduta sulla cross correlazione normalizzata, implementata tramite la routine MATLAB **normxcorr2**, che permette di ricavare una matrice di correlazione fra un ritaglio del riferimento e un

ritaglio del *target* (contenuto interamente all'interno del primo ritaglio) sfruttando le trasformate discrete di Fourier.

Una delle 32 osservazioni a bassa risoluzione viene presa come riferimento fisso, e rispetto ad essa verrà effettuata la registrazione delle restanti 31. Il primo passaggio consiste nell'applicazione di un filtro di *smoothing* gaussiano a varianza nota, imposta sia all'immagine di riferimento che all'immagine *target* per le motivazioni elencate al paragrafo 2.5.1.

Ritagliata poi una finestra sufficientemente grande all'interno del riferimento, per ogni rimanente osservazione LR viene ritagliata una finestra più piccola da confrontare con la prima. È in questo passaggio che viene imposto un grosso vincolo, poiché la scelta della posizione e delle dimensioni del riquadro, il cui ritaglio è effettuato con una apposita funzione MATLAB, risulta determinante per il raggiungimento di un buon risultato. Il riquadro deve infatti contenere al proprio interno soggetti particolari all'interno della scena, facilmente riconoscibili, minimizzando anche la quantità di zone omogenee, che potrebbero generare errori grossolani nel processo di registrazione. Ciò implica la scelta manuale dell'origine e delle dimensioni del riquadro di ritaglio, rendendo in tal modo l'algoritmo dipendente dalle scelte dell'operatore.

Determinata la posizione del picco all'interno della matrice di correlazione, sfruttando le relazioni geometriche che intercorrono tra le due finestre è possibile determinare gli *offset*, ovvero i parametri di trasformazione obiettivo della registrazione, da applicare al *target* per registrarlo all'interno del riferimento.

Dall'utilizzo dell'algoritmo è emerso che la registrazione così effettuata non tiene conto di eventuali angoli di rotazione del *target* rispetto al riferimento, dato che l'algoritmo ricerca la miglior correlazione tra le due immagini in input senza ruotarle, fornendo in output il picco del grado di correlazione associato ad un vettore di traslazione, calcolato a parte. La soluzione a tale problema è stata individuata nell'introduzione di una preliminare rotazione del *target*, posta all'interno di un ciclo in modo da coprire l'intero *range* di rotazioni possibili (in base agli angoli forniti nell'algoritmo di degradazione). Tale soluzione porta però alla generazione di porzioni di bordo completamente campite in nero, inserite per simulare la rotazione e non perdere informazioni, che potrebbero falsare le correlazione. Il *target* viene ruotato di un angolo iniziale di 8° in senso orario e successivamente ritagliato con una finestra delle dimensioni adeguate, tenendo ben presente che al suo interno sia contenuto il particolare di

interesse. Al variare dell'angolo sono registrati in un file di testo l'angolo di rotazione, i parametri di traslazione e il valore di picco della matrice di correlazione stimati dall'algoritmo. Iterativamente viene introdotto un piccolo incremento di angolo, che cala a seconda di quanto il picco di correlazione si discosta dal valore 1, cui segue la ripetizione del procedimento. Il concetto è che in tal modo, fra tutte le rotazioni applicate all'immagine *target*, ad una è associato un picco maggiore rispetto agli altri che identifica il corretto angolo di rotazione e dunque anche i corretti valori di traslazione dell'immagine *target* rispetto a quella di riferimento. Il risultato dell'applicazione dell'algoritmo è un file di testo contenente al proprio interno la lista dei parametri di trasformazione più corretti da applicare all'immagine da registrare.

Poiché la matrice di correlazione è calcolata per ogni immagine, per ogni incremento di angolo di 0.005° da -8° a 8° (a seconda del valore dell'ultimo picco, dato che se questo è inferiore a 0.98 l'incremento è maggiore), il tempo computazionale richiesto dalla macchina per l'esecuzione dell'intero *script* e quindi per la determinazione di tutti i parametri di registrazione, è di circa una decina di ore. Ovviamente i tempi necessari al processo dipendono fortemente dalla macchina a disposizione: nel corso della sperimentazione è stato utilizzato un calcolatore, messo a disposizione dal dipartimento, con processore a 8 *core*, *clock* pari a 2,93 GHz e 16 GByte di RAM.

3.2.1.1. Aumento delle dimensioni dell'immagine

I valori di *shift* individuati secondo il procedimento descritto risultano essere degli interi, fattore che rende impossibile quantificare gli *shift* sub-pixel che sono il reale oggetto di interesse. Per sopperire a tale problematica l'idea è quella di intervenire sulle osservazioni LR a monte del processo di registrazione. Prima di effettuare la cross correlazione normalizzata si procede infatti all'aumento delle dimensioni effettive delle immagini (non corrispondente a un reale aumento di risoluzione): come schematizzato in figura 3.2, a partire dall'osservazione LR viene generata una nuova immagine in una griglia delle dimensioni dell'immagine HR obiettivo del procedimento, in questo caso dunque di 4032×3024 pixel, in cui i valori di intensità all'interno di finestre con dimensioni pari al fattore di riduzione (4×4) sono fra loro identici. Le immagini così ottenute continuano ad avere la stessa risoluzione di partenza, con la differenza di poter individuare degli *shift* all'interno di aree della stessa intensità, quindi considerabili come sub-pixel.

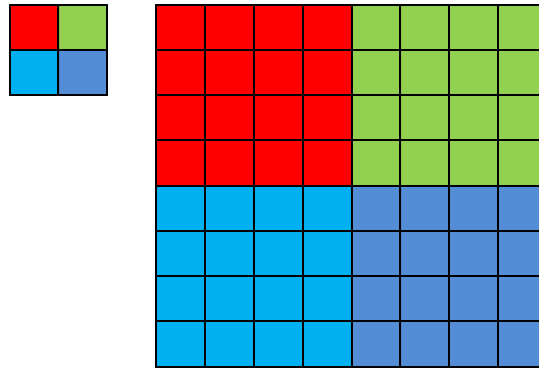


Fig. 3.2 – Osservazione LR normale (sinistra) e a dimensioni aumentate (destra)

Utilizzando lo *script* di cross correlazione normalizzata su questo nuovo set di immagini, dette “aumentate”, con l’accortezza di modificare le dimensioni dei riquadri di ritaglio, si ottiene come sopra un file di testo contenente i parametri di traslazione e rotazione più appropriati per ogni immagine. Ciò che cambia rispetto al caso precedente è che il significato dei parametri di trasformazione è ora modificato: nonostante questi siano ancora degli interi, in essi è possibile leggere degli *shift* con sensibilità pari a un quarto del pixel LR (utilizzando una finestra di aumento 4×4).

3.2.1.2. Applicazione dei parametri di trasformazione

Per concludere il processo di registrazione, vengono applicati alle immagini aumentate i parametri di trasformazione determinati come sopra, mantenendo griglie di dimensione HR. L’applicazione dei parametri è effettuata tramite le *routine* MATLAB **imrotate** e **imtranslate**, la prima per la rotazione e la seconda per le traslazioni, entrambe da impostare in modo da perdere la minor quantità di informazioni possibile mantenendo al tempo stesso una risoluzione fissa. Infatti, l’applicazione di una rotazione con la funzione **imrotate** genera, per angoli positivi, una rotazione in senso antiorario attorno al centro dell’immagine. Ciò provoca, nel caso si voglia assolutamente evitare la perdita di informazioni, l’aumento delle dimensioni della matrice, che nelle zone di bordo dove non sono disponibili valori di intensità viene riempita da valori pari a 0 (nero). Diversamente, volendo mantenere fissate le dimensioni della matrice ruotata, la perdita di informazioni dovuta al ritaglio della matrice stessa, è inevitabile. Quanto detto per la rotazione è valido anche nel caso delle traslazioni, orizzontali e verticali, delle quantità specificate che portano ad aumentare le dimensioni della matrice per mantenere i dati inalterati.

3.2.2. *Registrazione automatica intensity-based*

A causa del tempo macchina impiegato dall' algoritmo di cross correlazione normalizzata, si è cercata una soluzione alternativa per effettuare la registrazione delle immagini; quest'ultima è rappresentata dalle *routine* MATLAB specifiche per la registrazione **imregister** e **imregtform**, che permettono rispettivamente di registrare automaticamente e iterativamente un'immagine *target* rispetto a una di riferimento con un algoritmo *intensity-based* o di ottenere semplicemente la matrice di trasformazione necessaria, da applicare successivamente.

Utilizzando le funzioni appena descritte e specificando l'opzione di "trasformazione rigida" il processo di registrazione è effettuato applicando traslazioni e rotazioni. Con questi algoritmi non si ha quindi la necessità di introdurre di volta in volta la rotazione del *target*.

Per poter cogliere anche scostamenti inferiori al pixel, l'algoritmo viene lanciato sul set di immagini a risoluzione aumentata, mantenendo in tal modo il parallelismo con il caso della cross correlazione normalizzata.

Anche in questo caso, preliminarmente alla registrazione, sia l'immagine di riferimento che i vari *target* sono stati sfocati utilizzando un filtro di *smoothing* gaussiano a varianza nota e imposta.

Successivamente, si passa al processo di registrazione vero e proprio, che richiede innanzitutto l'introduzione di parametri per la convergenza del processo e per l'accettazione del risultato. La variazione dei parametri per la registrazione automatica porta a cambiamenti nella velocità e nella precisione raggiunta del processo di registrazione: ponendo valori più stringenti il procedimento diventa più lento, ma al tempo stesso più preciso, mentre utilizzando valori numericamente più alti il risultato è ottenuto con maggior velocità, a costo di una minore precisione. Per trovare un buon compromesso tra i due fattori, mantenendo comunque presente che dei due risulta essere più importante la precisione poiché una registrazione errata porterebbe ad una ricostruzione errata a sua volta, sono state eseguite alcune prove di registrazione su vari *dataset* di immagini. I valori dei parametri di registrazione sono stati poi fissati e mantenuti costanti in tutti i test successivi.

Una volta impostati tutti i parametri di registrazione questa viene effettuata automaticamente, senza la necessità di specificare *control points* o di selezionare riquadri all'interno delle immagini, mantenendo in tal modo l'intero procedimento indipendente da scelte soggettive

dell'operatore. In più, il processo effettuato in questo modo risulta essere molto più rapido rispetto alla cross correlazione normalizzata, dato che l'esecuzione della registrazione per tutto il set delle 32 immagini LR aumentate richiede all'incirca mezz'ora.

3.3. Proiezione delle immagini su una griglia a risoluzione aumentata

Come da modello già descritto, la fase successiva alla registrazione consiste nel processo di proiezione delle immagini registrate, ovvero il posizionamento di tutti i *digital number* delle matrici all'interno di un'unica griglia a risoluzione aumentata. Successivamente al posizionamento è poi possibile passare all'ultima fase, l'interpolazione dei valori incogniti di *digital number* all'interno della griglia, ottenendo infine l'immagine ricostruita. In questo paragrafo, con l'esclusione del primo sottoparagrafo, verrà trattata la sola fase di posizionamento delle immagini, rimandando l'aspetto dell'interpolazione al paragrafo successivo.

3.3.1. Proiezione e media dei *digital number*

In una prima redazione grossolana degli algoritmi, per una questione di semplicità le fasi di proiezione ed interpolazione delle immagini sono fatte coincidere: per ottenere l'immagine ad alta risoluzione finale si considerano tutte le immagini a dimensioni aumentate registrate, mediando le intensità dei pixel nella stessa posizione. A livello di algoritmo, viene creata una singola matrice delle dimensioni della griglia HR al cui interno vengono sommati, per ogni posizione, tutti i valori di *digital number* delle matrici a dimensioni aumentate, ottenendo quindi una matrice somma di tutte le immagini registrate. Infine, ogni valore all'interno di tale matrice è diviso per il numero di *digital number* diversi da 0 che hanno contribuito al totale (per evitare problemi sui bordi, in cui alcune immagini presentano zone campite in nero). In tal modo viene calcolata una media dei valori di tutte le immagini a dimensioni aumentate registrate, effettuando di fatto una prima fusione di esse in un'unica immagine a super risoluzione.

Questo passaggio chiude la prima serie di algoritmi redatti, che hanno portato alla creazione di una prima immagine "a super-risoluzione" a colori. Complessivamente i tempi richiesti per l'ottenimento di un'immagine SR sono risultati essere estremamente lunghi, in quanto l'algoritmo scritto risulta essere scarsamente ottimizzato.

Per evitare tempi macchina eccessivamente prolungati, la scelta più logica risulta essere quella di abbandonare le immagini a colori, effettuando tutti i test successivi su immagini in scala di grigio. Tale semplice scelta permette infatti di ridurre significativamente i tempi necessari all'effettuazione di un ciclo completo, dato che ogni processo è applicato ad una sola matrice di pixel invece che a tre (con eccezione della registrazione, che fin da subito era effettuata solo su una delle tre bande). In più, per ottenere un'ulteriore riduzione dei tempi computazionali e per effettuare test i più simili possibile all'applicazione finale sulle immagini termiche, la risoluzione delle immagini HR è variata fino a un valore di 2560×1920 , che, degradato con un fattore di riduzione pari a 4, porta ad osservazioni LR con risoluzione 640×480 , pari a quella delle immagini termiche.

3.3.2. *Proiezione di immagini esplose*

Dopo l'esecuzione di un primo algoritmo semplificato di ricostruzione, l'obiettivo è quello di affinare l'intero procedimento, soprattutto dal punto di vista del grado di somiglianza raggiunto tra l'immagine HR originale e quella ricostruita. Per quanto riguarda il processo di registrazione, sia la cross correlazione normalizzata che la registrazione automatica forniscono risultati soddisfacenti e per tale motivo l'attenzione ricade sulle fasi successive. Per quanto riguarda la prima, è necessario trovare una soluzione che permetta di collocare i singoli pixel delle osservazioni LR nella posizione corretta all'interno della griglia a risoluzione aumentata, mantenendo dove non vi siano valori di intensità disponibili degli spazi vuoti da riempire successivamente interpolando le intensità dei pixel ad essi adiacenti.

La realizzazione di quanto descritto segue un percorso diverso, ma parallelo, a quello seguito dagli algoritmi già descritti. Preliminarmente le osservazioni LR vengono rielaborate per ottenerne una loro versione "esplosa", praticamente vuota, come schematizzato in figura 3.3.

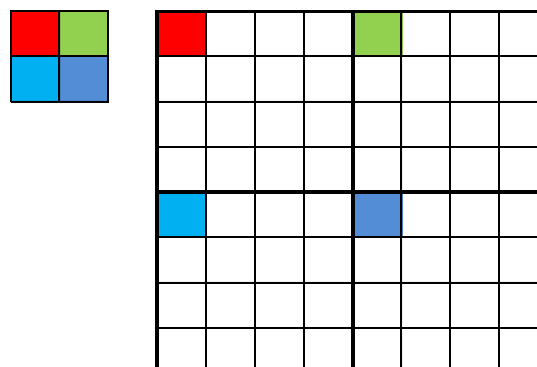


Fig. 3.3 – *Osservazione LR originale (sinistra) e “esplosa” (destra)*

Viene creata una nuova griglia delle dimensioni dell'immagine HR, suddivisa idealmente in finestre delle dimensioni del singolo pixel LR (nel caso indicato 4×4). Nell'angolo in alto a sinistra di ogni singola finestra è inserito il pixel corrispondente alla posizione della finestra attualmente considerata (nell'angolo in alto a sinistra della prima finestra vi è il pixel in posizione (1,1) dell'immagine LR), mentre nella restante parte sono inseriti degli 0.

Successivamente, la registrazione è effettuata come in precedenza utilizzando sempre il set di immagini aumentate, con la differenza, nel caso della registrazione automatica, che questa non è applicata direttamente ad esse, ma viene utilizzata per ottenere la matrice di trasformazione, da applicare successivamente alle immagini esplose tramite la funzione **imwarp**. Anche nel caso della registrazione tramite cross correlazione normalizzata i parametri sono ottenuti utilizzando il set di immagini aumentate e poi applicati al set di immagini esplose, utilizzando le stesse funzioni di rotazione e traslazione del caso precedente.

Le immagini esplose così registrate, secondo i due differenti algoritmi e quindi in due cicli separati, vengono successivamente proiettate all'interno di una griglia a risoluzione aumentata solidale all'osservazione LR esplosa di riferimento. Di volta in volta, all'interno della griglia vengono sommati i valori di intensità di tutte le osservazioni LR, riempiendola parzialmente. Agli spazi vuoti all'interno della griglia vengono sostituiti i valori di intensità di tutte le osservazioni. Per tenere conto della possibilità che in alcuni pixel della griglia HR possano cadere più pixel di osservazioni LR a intensità diversa e per dare lo stesso peso a tutti quanti, il procedimento di riempimento è effettuato allo stesso modo del calcolo della media: ogni elemento della griglia HR è la somma dei rispettivi elementi delle osservazioni LR, successivamente da dividere per il numero di elementi diversi da 0 che effettivamente hanno contribuito. In tal modo, dove è presente un solo elemento ad esso è assegnato peso unitario, mentre dove sono presenti più valori di intensità, ad ognuno è assegnato lo stesso peso, effettuando di fatto una media aritmetica. La differenza con la semplice media effettuata in precedenza è che nel presente caso, dove non sono presenti valori di intensità all'interno della griglia è lasciato uno spazio vuoto, come esemplificato in figura 3.4.

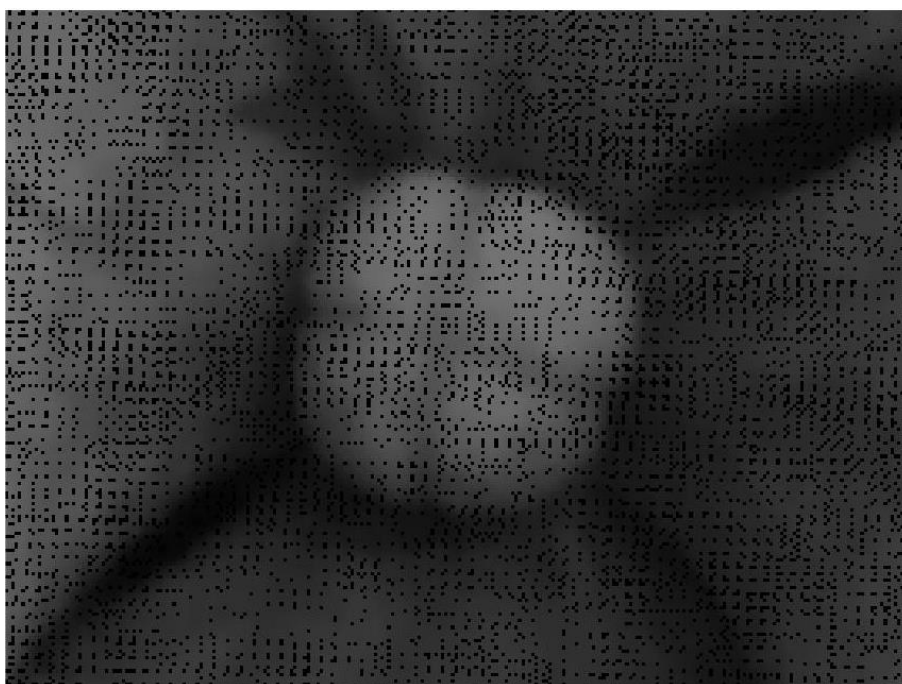


Fig. 3.4 – *Particolare della griglia di proiezione parzialmente riempita*

3.4. Interpolazione tramite Inverse Distance Weighting

Per effettuare un'interpolazione vera e propria e non una media aritmetica è necessario considerare, per ogni punto da interpolare, tutti i pixel nel suo intorno ed assegnarvi dei pesi, a seconda della tecnica utilizzata. Nella redazione del nuovo algoritmo di interpolazione, per ogni pixel della matrice di proiezione ottenuta in precedenza viene creata una finestra che rappresenti un intorno, al cui interno possono ricadere uno o più valori di intensità ottenuti dalle osservazioni LR.

Inizialmente, il metodo designato per l'interpolazione è il *Least Squares Plane fitting*, che richiede però almeno tre valori di *digital number* nell'intorno del pixel considerato per creare un sistema determinato e risolvibile. Poiché tale condizione non risulta sempre vera, il risultato del test effettuato con tale tecnica non è considerabile come corretto.

Vi è dunque la necessità di variare la tecnica di interpolazione utilizzata, mantenendo allo stesso tempo una sufficiente precisione. La soluzione è rappresentata dall'*Inverse Distance Weighting*, con fattore di potenza P pari a 1, in cui il peso di ogni pixel all'interno della finestra 5×5 è assegnato in base alla distanza dei valori a disposizione rispetto al centro della finestra, in cui si trova il pixel da interpolare. La realizzazione dell'algoritmo segue di pari passo la formula per il calcolo dei pesi esplicitata al paragrafo 2.6.2. Ripetendo il

procedimento per ogni spazio vuoto all'interno della matrice di proiezione delle immagini, il risultato ottenuto è nuovamente un esempio di immagine a super-risoluzione, ora più accurato rispetto all'utilizzo della semplice media.

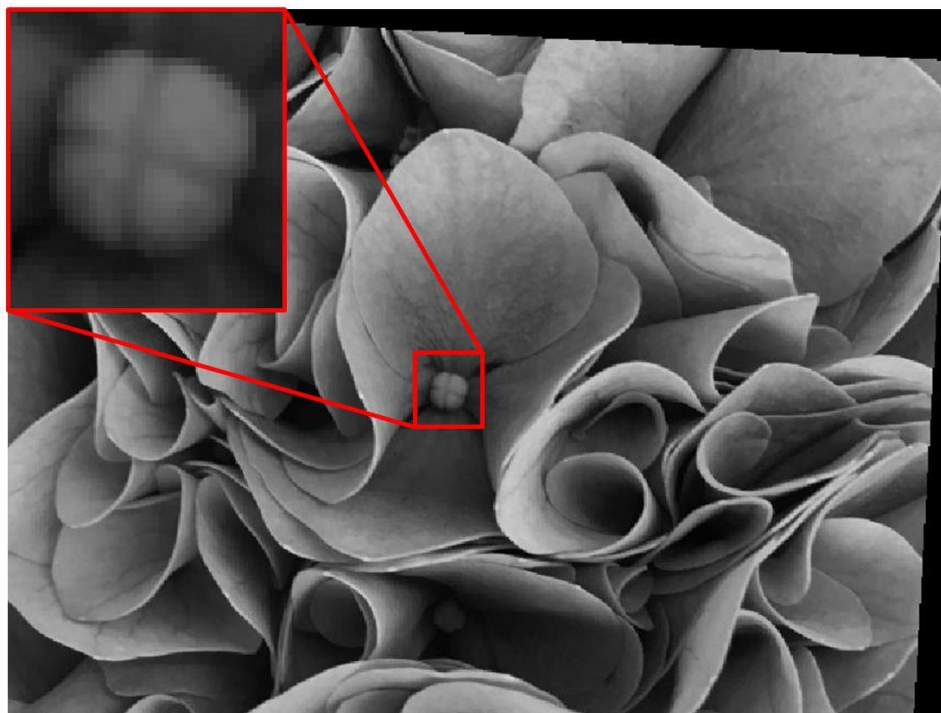


Fig. 3.5 – Particolare di immagine LR

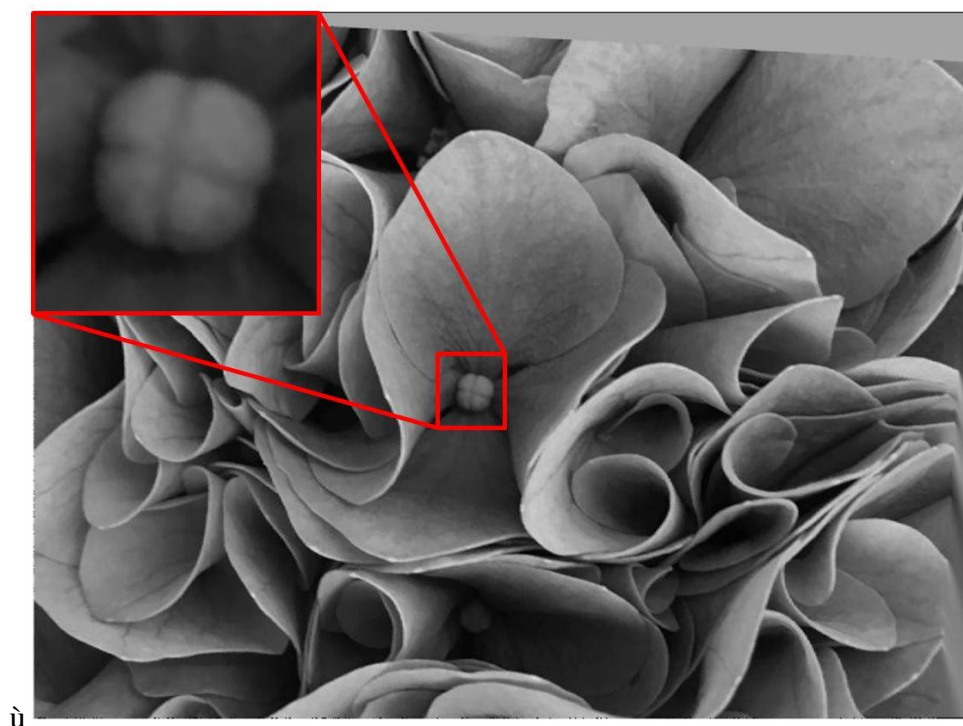


Fig. 3.6 – Particolare di immagine HR ricostruita

Nelle immagini 3.5 e 3.6 sono riportate rispettivamente una delle osservazioni LR (a risoluzione 640×480 pixel) e l'immagine ad alta risoluzione ricostruita (a risoluzione 2560×1920 pixel) utilizzando la combinazione di registrazione *intensity-based* e interpolazione tramite IDW. L'incremento di risoluzione operato nel passaggio dal set di osservazioni a bassa risoluzione all'immagine ricostruita è pari a 4 e nella ricostruzione del caso presente sono state utilizzate 32 immagini LR, in esubero rispetto al minimo riscontrato in bibliografia per poter eliminare, almeno in una prima fase, eventuali problemi legati alla dimensione del campione di partenza.

Confrontando fra loro le immagini, soprattutto nel particolare riportato, a seguito della ricostruzione vi è un effettivo miglioramento della qualità, riscontrabile nella maggior quantità di particolari visibili all'interno di ogni soggetto e in una maggior fluidità delle linee di bordo.

Un'ulteriore verifica dell'efficacia dell'algoritmo è riportata nelle figure 3.7 e 3.8, in cui il soggetto della ricostruzione è ora generato da un *dataset* diverso, un'automobile con targa visibile. Lo scopo del test è verificare l'efficacia nella ricostruzione dei caratteri della targa, in modo da capire se vi sia un reale aumento di risoluzione anche in caso di scritte.

Anche in questo caso le immagini a bassa risoluzione sono state ottenute a partire dall'immagine originale ad alta risoluzione introducendo *shift* casuali. Confrontando l'osservazione LR originale e l'immagine ricostruita è possibile osservare come, oltre a lettere e numeri di grandi dimensioni leggibili già in bassa risoluzione, anche quelli di minori dimensioni (particolari nel lato sinistro e al centro) siano stati ricostruiti e risultino successivamente più chiari.



Fig. 3.7 – Particolare di una scritta in immagine LR



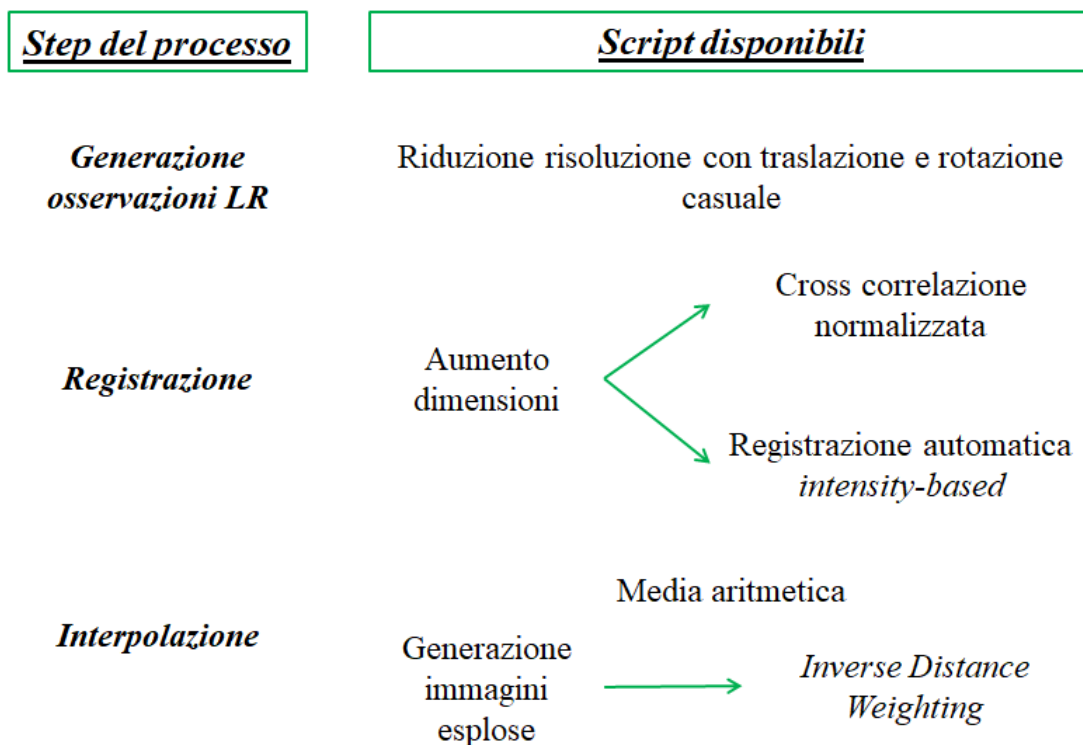
Fig. 3.8 – Particolare di una scritta in immagine ricostruita

3.5. Verifiche di qualità delle immagini ricostruite

Una volta creati gli algoritmi per la super risoluzione, il passo successivo più naturale è quello di verificare quanto questi siano realmente efficaci e se generino effettivamente delle immagini HR raffrontabili con l'immagine HR di partenza da cui le osservazioni LR sono state ottenute.

3.5.1. Confronto tra combinazioni di algoritmi

Parte integrante del lavoro di tesi, oltre alla scrittura degli algoritmi di super-risoluzione, è la verifica dei risultati ottenuti, effettuata nuovamente tramite uno *script* che permette di automatizzare il processo per le varie combinazioni di algoritmi utilizzate. I risultati sono ottenuti variando di volta in volta gli *script* di registrazione e/o interpolazione utilizzati, per un totale di quattro combinazioni possibili. In tutti i casi esaminati rimangono costanti le osservazioni LR generate che costituiscono la base per i test effettuati, e l'algoritmo per l'aumento delle dimensioni delle matrici. Nella tabella seguente vengono riassunti tutti gli algoritmi utilizzati per effettuare i test:



Tab. 3.1 – *Elenco degli algoritmi utilizzati*

Il punto di partenza per effettuare le verifiche di corrispondenza tra l'immagine HR originale e quella ricostruita è rappresentato dalla matrice degli scarti tra le due immagini, ovvero le differenze tra esse. Di volta in volta lo *script* considera una delle quattro immagini generate dalle altrettante combinazioni e preliminarmente ne effettua una rotazione pari in modulo, ma opposta, all'angolo di rotazione di cui è stata ruotata l'osservazione LR presa come riferimento rispetto all'immagine HR originale, noto dal *file* di testo contenente tutti i parametri di degradazione di quest'ultima. Infatti, fondamentale per l'esecuzione di una verifica effettivamente oggettiva e corretta, è tenere conto che il processo di registrazione fornisce i parametri per sovrapporre le osservazioni LR ad una di riferimento, che a sua volta è ruotata e traslata rispetto all'immagine HR originale.

Per tale motivo, per poter svolgere verifiche corrette e oggettive, è necessario registrare preliminarmente l'immagine ricostruita rispetto a quella originale. Una volta effettuata la rotazione, l'immagine ricostruita e quella originale sono teoricamente allineate allo stesso modo, a meno di una traslazione lungo l'asse x e una lungo l'asse y. Per quantificare tali *offset* si utilizza nuovamente la funzione di cross correlazione normalizzata, applicando anche in questo caso dei filtri di *smoothing*, con la differenza che ora non sussiste la necessità di ruotare di volta in volta l'immagine, dato che la rotazione corretta è già stata introdotta. Trovati gli *offset*, questi sono applicati all'immagine HR ricostruita, che a seguito di tale processo risulta effettivamente sovrapponibile all'immagine originale. La fase descritta è giustificata dall'impossibilità di utilizzare i parametri casuali noti, relativi all'immagine di riferimento, salvati nel processo di degradazione, a causa del diverso sistema di riferimento in cui le operazioni sono svolte rispetto a quello dell'immagine originale.

A questo punto entrambe le immagini vengono ritagliate con un riquadro che ne contorni una porzione interna, eliminando in tal modo il problema delle zone campite di nero sui bordi che falserebbero il risultato delle verifiche. Queste ultime sono dunque effettuate su immagini a risoluzione diversa da quella effettiva di ricostruzione, ma centrate sul particolare di interesse delle scena, con lo scopo di eliminare le zone di bordo campite in nero durante il processo.

Preparate le immagini, vengono svolte le verifiche vere e proprie, che in una prima fase consistono nel calcolo e nella visualizzazione tramite istogramma degli scarti e, come termine di confronto, nel calcolo anche degli scarti che una generica immagine LR aumentata, anch'essa registrata, presenta rispetto all'immagine HR originale.

Nell’algoritmo viene successivamente implementata anche una funzione per determinare alcune statistiche riguardanti le serie degli scarti pixel per pixel tra le due immagini; per ogni matrice degli scarti viene creato un vettore contenente gli elementi della matrice stessa, dal quale vengono successivamente calcolati, dopo averli ordinati in successione crescente, deviazione standard, valore massimo e valore minimo. A parte vengono anche calcolate delle medie su 5 valori di scarto successivi, a partire da quelli minori fino a quelli maggiori, per verificare la velocità di variazione; calcolata la media sui 5 valori più bassi, si calcola una nuova media scalando di una posizione la finestra e prendendo quindi in considerazione i valori ordinati dal secondo al sesto, etc. Il calcolo della deviazione standard σ , detti s_i i valori degli n scarti i -esimi, è effettuato secondo la seguente formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (s_i - \bar{s})^2}{n - 1}}$$

Le statistiche calcolate per ogni combinazione di script di registrazione e interpolazione, ad eccezione delle serie di scarti mediati, sono riportate nelle figure seguenti:

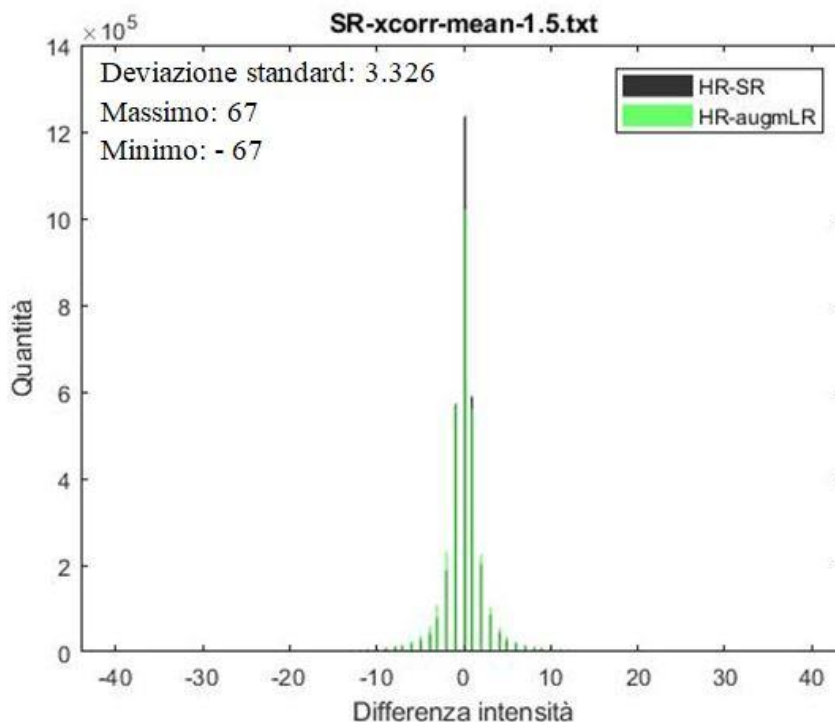


Fig. 3.9a – *Registrazione con NCC e interpolazione con media*

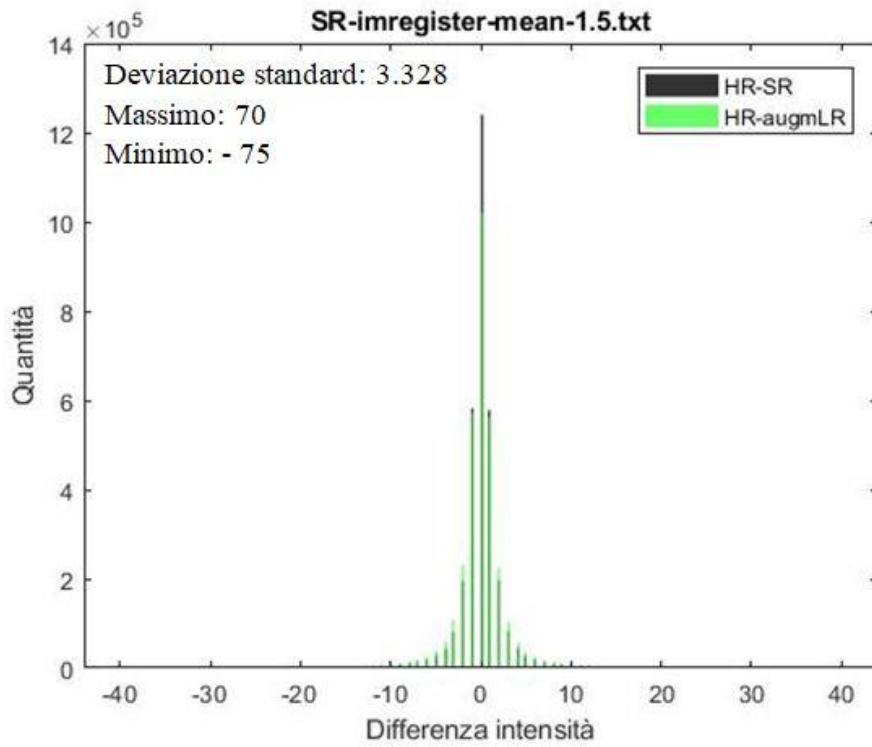


Fig. 3.9b – *Registrazione intensity-based e interpolazione con media*

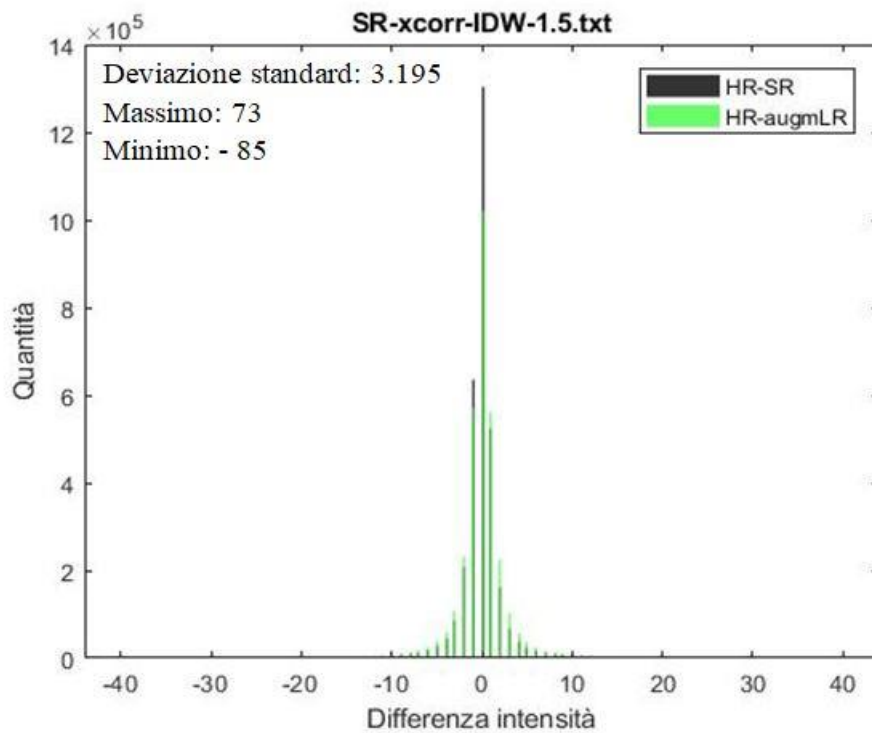


Fig. 3.9c – *Registrazione con NCC e interpolazione con IDW*

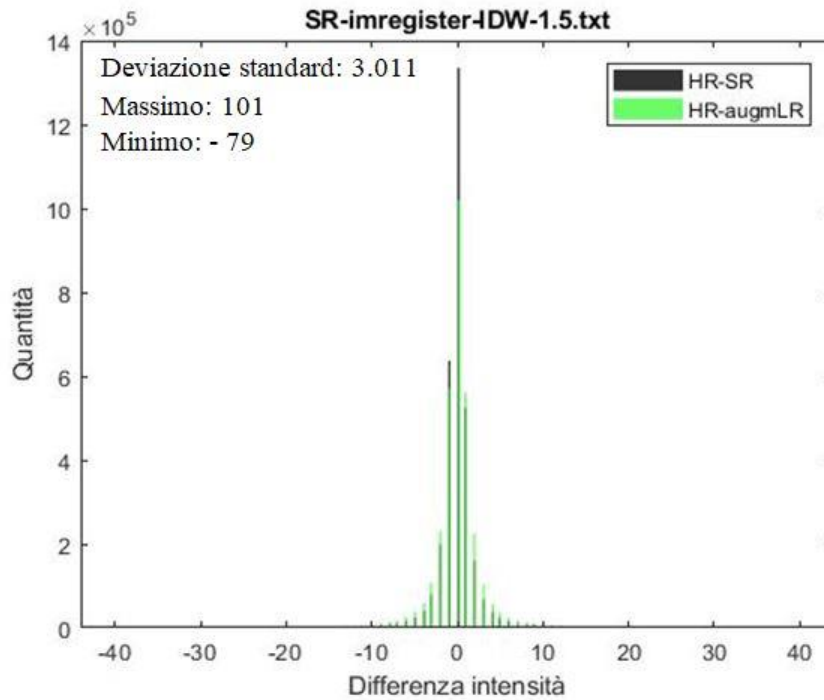


Fig. 3.9d – *Registrazione intensity-based e interpolazione con IDW*

In questa fase di verifiche è da tenere in conto che tutte le immagini ricostruite sono il risultato dell'interpolazione di osservazioni LR registrate utilizzando un valore di deviazione standard per il filtro di *smoothing* gaussiano, costante per tutte le combinazioni e pari a 1.5, e che quindi non vi è attualmente alcuna influenza della variazione di quest'ultimo

Da quanto riportato risulta che, qualunque sia la combinazione di *script* utilizzata, si ottiene un netto miglioramento in termini di scarti rispetto ad una immagine a bassa risoluzione semplicemente aumentata di dimensioni. Determinata la presenza di un miglioramento, la combinazione che genera la maggior quantità di scarti, rispetto all'immagine originale, pari a 0 è quella ottenuta da registrazione automatica iterativa *intensity-based* e interpolazione tramite IDW. Tale risultato può dipendere dalla maggior precisione (ed anche alla maggiore stabilità) dell'algoritmo di registrazione MATLAB rispetto a quello redatto autonomamente, dato che sicuramente quest'ultimo è meno ottimizzato e più instabile del primo. L'impatto probabilmente maggiore sul risultato lo ha però la tecnica di interpolazione utilizzata, poiché una semplice media non tiene conto del contributo che i pixel adiacenti possono fornire, diversamente da un'interpolazione più fine. Infatti, confrontando i grafici a due a due, è possibile notare che il miglioramento ottenuto con una tecnica più raffinata di interpolazione è nettamente superiore a quello ottenuto con l'algoritmo di registrazione automatica. Nonostante ciò, quest'ultimo è considerato di gran lunga più efficiente rispetto alla cross

correlazione normalizzata, non tanto per il risultato ottenuto, quanto per la rapidità con cui i parametri di trasformazione vengono calcolati.

In base a quanto affermato e visto finora, la combinazione scelta per tutti i successivi test e per le applicazioni future è la composizione di registrazione automatica iterativa *intensity-based* e interpolazione con IDW.

3.5.2. Filtro di smoothing

Una volta determinata la migliore combinazione di algoritmi tra tutte quelle a disposizione, il passo successivo è stato quello di verificare le ipotesi da bibliografia relative al valore della deviazione standard da applicare tramite il filtro di *smoothing* gaussiano in fase di preregistrazione. Secondo quanto ricercato tale valore ottimale si attesta pari a 0.5 pixel, un buon compromesso per eliminare eventuali disturbi presenti all'interno delle immagini e per smussare bordi eccessivamente netti, senza compromettere eccessivamente, al tempo stesso, l'immagine.

La verifica dell'ipotesi viene effettuata lanciando l'intero algoritmo di ricostruzione e variando di volta in volta il valore di deviazione standard da applicare sia all'immagine di riferimento che alle varie immagini *target*. Come già in precedenza, il confronto viene effettuato prendendo in considerazione i valori di massimo, minimo e deviazione standard degli scarti tra l'immagine originale e quella ricostruita. I test vengono effettuati utilizzando la combinazione di algoritmi definita nel paragrafo precedente e considerando valori di deviazione standard per lo *smoothing* pari a 0.0, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0, 1.5 e 2.0 pixel. I risultati ottenuti sono riportati in tabella 3.2.

<u>Dev. Std smoothing</u>	<u>Dev. Std scarti HR-SR</u>	<u>Scarto massimo</u>	<u>Scarto minimo</u>
0.0	3.013	101	-71
0.3	3.013	101	-71
0.4	3.010	101	-71
0.5	2.985	101	-77
0.6	2.985	101	-77
0.7	2.992	101	-77
1.0	3.014	101	-77
1.5	3.011	101	-79
2.0	3.004	88	-79
0.5	2.961	86	-77

Tab. 3.2 – Statistiche degli scarti dei test sullo smoothing



Fig. 3.10 – *Bordi sfumati nell'immagine ricostruita*

I valori molto elevati in valore assoluto relativi agli scarti massimi e minimi sono dovuti alla presenza di bordi netti all'interno delle immagini. Il processo di ricostruzione, infatti, non è in grado di ricreare bordi netti fra due oggetti a causa del concetto stesso di interpolazione, che porta alla generazione di una sfumatura di colore nel passaggio da un oggetto ad un altro, come mostrato in figura 3.10 (l'immagine è la rappresentazione dello scarto tra la matrice dell'immagine HR originale e quella ricostruita). In più, come verificabile negli istogrammi già mostrati, la quantità di scarti a valore assoluto molto elevato è irrisoria rispetto a quelli a basso valore assoluto.

La verifica vera e propria viene dunque effettuata sui valori di deviazione standard degli scarti tra i valori di intensità dell'immagine ricostruita e quella di partenza. Da quanto mostrato in tabella non si notano grandi variazioni tra l'applicazione di un filtro di *smoothing* a diversi valori di deviazione standard, poiché le variazioni negli scarti quadratici medi sono minime. Per quanto piccole siano le differenze, il risultato ottenuto conferma che l'utilizzo di un filtro di *smoothing* a deviazione standard 0.5 porta al risultato migliore, motivo per cui i test successivi ad ora vengono svolti utilizzando tale valore.

3.5.3. *Posizione dei pixel nelle immagini esplose*

Prima di passare ai test successivi viene effettuata una verifica sulla fase di esplosione delle osservazioni LR, in cui l'operatore effettua la scelta sul posizionamento del valore di intensità del pixel LR all'interno della finestra della griglia HR. Nei primi test tale pixel viene posto in posizione (1,1) della finestra 4×4 per poter procedere alla fase di ricostruzione. Il punto da verificare è se la scelta di tale posizione rispetto ad un'altra possa influenzare il risultato finale dell'algoritmo; viene quindi lanciato l'intero processo più volte, modificando per ogni situazione la posizione del pixel all'interno della finestra. Il risultato finale ottenuto da tale prova conferma la mancanza di influenza che la scelta ha sul risultato della ricostruzione. Per tale motivo la posizione del pixel nelle immagini esplose viene mantenuta sul pixel (1,1) della finestra nella griglia a dimensioni aumentate.

3.5.4. *Numero di immagini necessarie al processo di ricostruzione*

L'intero lavoro, volto inizialmente al "semplice" ottenimento di un'immagine ricostruita, segue di volta in volta un processo di ottimizzazione, con il tentativo di ricavare informazioni su quali siano i migliori parametri da inserire nell'intero processo, come ad esempio il valore di deviazione standard per il filtro di *smoothing* o la posizione del singolo pixel dell'osservazione LR all'interno della finestra di pixel HR nelle immagini esplose.

Si focalizza ora l'attenzione non tanto sul processo in sé, quanto sui dati a disposizione, ovvero sulla dimensione del campione di partenza. Tutti i test effettuati fino a questo punto sono stati svolti inserendo in input 32 immagini a bassa risoluzione, ottenute dal generatore *random* di cui sopra. Tale numero è teoricamente molto elevato e garantisce, se le immagini sono ottenute con le dovute accortezze, una ricostruzione sufficientemente precisa. La verifica da effettuare consiste nel tentare di ricostruire le immagini riducendo o aumentando di volta in volta le dimensioni del campione di partenza, mantenendo allo stesso tempo costante il fattore di aumento della risoluzione, con valore ancora pari a 4.

Il set completo di immagini a bassa risoluzione è quello utilizzato finora e di volta in volta da esso ne vengono selezionate alcune a seconda del numero necessario. Le dimensioni dei campioni presi in esame sono assunte in modo tale da poter quantificare il miglioramento rispetto alle immagini a bassa risoluzione e il numero di immagini LR per ciascuno è rispettivamente di 1, 4, 8, 16 e 32. La metodologia di verifica rimane identica a quella applicata nei casi precedenti, con la determinazione degli istogrammi e il calcolo delle

statistiche. Il confronto sul miglioramento viene quindi effettuato prendendo in esame la deviazione standard degli scarti tra l'immagine ad alta risoluzione originale e quelle ricostruite. Lo scopo ultimo è quello di determinare un numero minimo di osservazioni LR necessarie alla ricostruzione, tenendo ben presente che eventuali immagini a bassa risoluzione in esubero rispetto al minimo dichiarato sufficiente non possono che migliorare la qualità finale della ricostruzione, o almeno semplificarla. Parallelamente, è vero anche che un numero maggiore di immagini in input porta ad un aumento del tempo computazionale necessario alla registrazione, fattore di cui tenere conto.

Per mantenere oggettività ed evitare quindi di selezionare un set di immagini particolarmente "buone" o "errate", ogni test sui campioni di grandezza N prescelta è effettuato su finestre mobili all'interno delle 32 immagini totali. Di volta in volta viene ricostruita l'immagine considerando le N osservazioni LR e poi spostando la finestra del campione in avanti. Ad esempio ciò significa che il test con quattro immagini, considerando di partire dalla prima delle 32 del set, viene effettuato prima sulle immagini da 1 a 4, poi su quelle da 2 a 5, da 3 a 6, etc., fino ad arrivare alle ultime quattro, per un totale di 28 combinazioni di 4 immagini. Si calcolano poi le statistiche, in particolare la deviazione standard, relative ad ogni immagine ricostruita a partire da un campione di dimensione N. La deviazione standard relativa al generico campione di dimensione N è poi determinata calcolando una media di tutte quelle relative a campioni della dimensione in esame.

Il test è stato svolto utilizzando gli stessi algoritmi dei casi precedenti, a differenza di un'aggiunta di codice per la selezione delle dimensioni e l'automatizzazione del movimento delle finestre lungo le 32 immagini a bassa risoluzione. Non ci è così la necessità di lanciare di volta in volta l'algoritmo manualmente e alla fine è possibile effettuare la verifica sull'intero set delle immagini ricostruite ottenendo tutte le statistiche, in particolare lo scarto quadratico medio. I risultati numerici ottenuti durante tale operazione sono riportati nella tabella 3.3.

<u><i>Deviazione standard scarti HR-SR</i></u>					
<u><i>1</i></u>	<u><i>4</i></u>	<u><i>8</i></u>	<u><i>16</i></u>	<u><i>24</i></u>	<u><i>32</i></u>
5.395	4.729	4.000	3.283	3.108	2.977
5.497	4.722	3.614	3.089	2.909	
5.335	4.805	3.633	3.409	3.563	
5.491	4.617	3.936	3.281	3.143	
5.427	4.964	3.677	3.338	3.338	
5.734	4.623	3.752	3.112	2.918	
5.529	4.924	3.693	3.443	3.253	

5.575	4.669	3.794	3.127	2.973
5.330	4.670	3.864	3.279	3.093
5.313	4.761	3.681	3.276	
5.300	4.608	3.659	3.186	
5.511	4.831	3.749	3.302	
5.377	4.695	3.891	3.514	
5.357	4.697	3.615	3.190	
5.597	4.768	3.924	3.594	
5.341	4.645	3.709	3.221	
5.365	4.631	3.827	3.224	
5.530	4.669	3.997		
5.384	4.618	3.627		
5.453	4.694	4.086		
5.452	4.720	3.616		
5.575	4.752	3.861		
5.858	4.690	3.832		
5.451	4.593	3.860		
5.377	4.777	3.769		
5.527	4.879			
5.400	4.735			
5.554	4.674			
5.624	4.802			
5.404				
5.425				
5.369				

Tab. 3.3 – *SQM degli scarti tra immagine originale e immagini ricostruite*

Da questi valori di deviazione standard degli scarti è poi possibile calcolare un unico valore medio rappresentativo del numero di immagini N utilizzato, come indicato nel grafico di figura 3.11.

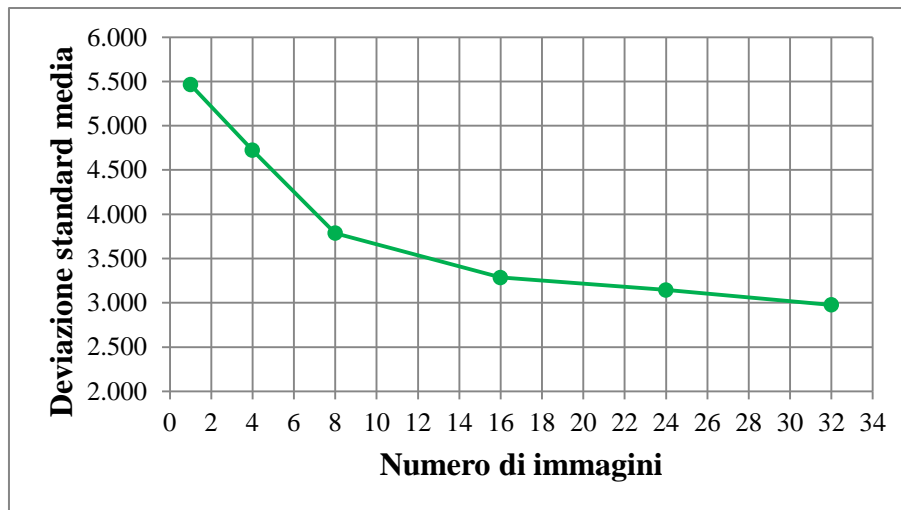


Fig. 3.11 – *Andamento dello SQM degli scarti*

Osservando l'andamento dello SQM, è possibile individuare come il miglioramento ottenuto con l'aumento del numero di immagini utilizzate diminuisce all'aumentare dello stesso. È possibile affermare che, per un fattore di aumento della risoluzione pari a 4, il numero minimo di immagini da utilizzare nel processo di ricostruzione sia pari a 16 ovvero, generalizzando il problema analizzato, che il numero minimo di osservazioni LR da coinvolgere sia effettivamente pari al quadrato dell'aumento di risoluzione ricercato, confermando quanto ritrovato in bibliografia. Aumentando il numero di immagini oltre a 16 il miglioramento è meno evidente, per cui bisogna considerare quanto sia oneroso, sia in termini di ripresa che in termini di tempi di elaborazione, aggiungere immagini oltre tale soglia. Per quanto riguarda il limite massimo di immagini da utilizzare, la scelta è quindi quella di non esprimere un numero dato che, benché l'aumento di qualità ottenibile utilizzando molte osservazioni in esubero sia limitato, l'utilizzo di più osservazioni del necessario consente di introdurre una maggior ridondanza nel problema.

Per comprendere al meglio il significato di quanto appena descritto, in figura 3.12 sono riportati dei particolari relativamente piccoli estratti rispettivamente dall'immagine originale, e dalle immagini ricostruite utilizzando 1,4,8,16,24 e 32 osservazioni LR. La differenza con l'immagine originale risulta ovviamente più marcata nel confronto con una semplice osservazione LR, ma già con l'utilizzo di 4 immagini a bassa risoluzione vi è la ricostruzione di particolari prima non visibili fino ad arrivare, nel caso della ricostruzione con 24 e 32 immagini, a differenze praticamente non percettibili.

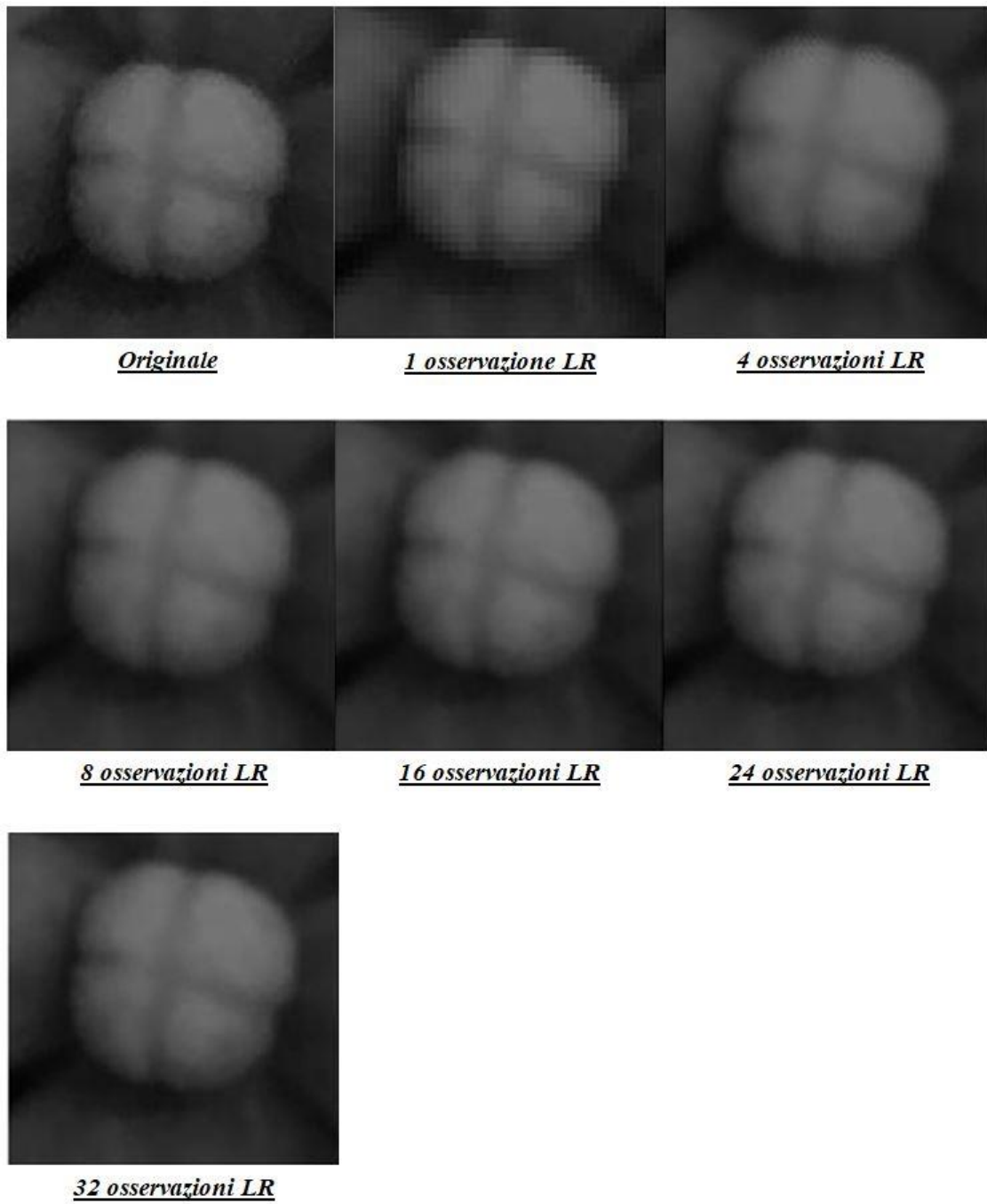


Fig. 3.12 – Particolari ricostruiti con diversi gruppi di immagini

4. APPLICAZIONI

Come anticipato in precedenza, la fase dei test preliminari ha lo scopo di pervenire ad una serie di applicazioni su immagini reali ottenute da una fotocamera e non tramite la degradazione di immagini HR di partenza. L'assenza di un'immagine ad alta risoluzione di partenza porta alla conseguenza che da questo punto in poi non sussiste più la possibilità di quantificare oggettivamente la qualità della ricostruzione ottenuta, termine di paragone utilizzato in precedenza tramite il calcolo degli scarti. L'unica possibilità di verifica che continua a sussistere è una semplice verifica visiva della ricostruzione, che permette la sola individuazione di eventuali errori grossolani.

La premessa per procedere con questa fase è dunque che l'algoritmo scelto sia affidabile e non produca errori difficilmente individuabili ad occhio nudo, come lievi errori nel processo di registrazione, che genererebbero una ricostruzione inesatta.

4.1. Immagini digitali in scala di grigi

Prima di procedere alla ricostruzione di immagini termiche, gli algoritmi vengono testati su una serie di immagini tradizionali a bassa risoluzione, ottenute da fotocamere o telefoni.

4.1.1. Canon EOS 6D

La prima idea per una serie di test è quella di ottenere le osservazioni a bassa risoluzione con una fotocamera Canon EOS 6D, modello che permette un'impostazione personalizzata della risoluzione delle immagini in uscita. Per poter ottenere immagini della qualità desiderata, quest'ultima è stata impostata a 720×480 pixel e il formato di salvataggio è duplice: *raw* e TIFF, in modo da scegliere successivamente quale dei due effettivamente utilizzare. Il soggetto in esame è una delle facciate della sede della Scuola di Ingegneria ed Architettura di Bologna e le fotografie sono state scattate ponendo la macchina su un treppiede; l'inserimento di trasformazioni rigide diverse per ogni immagine è ottenuto variando di volta in volta l'angolo tra la base della camera e il piano di appoggio e spostando il treppiede lateralmente o sollevandolo/abbassandolo. Come anticipato in precedenza, la mancanza di un'immagine HR di partenza e l'introduzione di spostamenti e rotazioni non quantificabili (le vibrazioni fornite alla macchina con la semplice pressione di un tasto di scatto sono sufficienti a fornire *shift* a livello di pixel o sub-pixel) non permettono un confronto oggettivo successivo alla

ricostruzione tramite il calcolo delle statistiche legate agli scarti. La macchina utilizzata avrebbe tecnicamente permesso di ottenere un'immagine a risoluzione confrontabile con quella di ricostruzione, ma per problemi di natura differente elencati in seguito tale fase di verifica non è risultata possibile. Come ulteriore test le immagini ottenute in tal modo presentano una serie di particolarità:

- Le rotazioni e gli spostamenti forniti al treppiede sono di grande entità, di molto superiori al singolo pixel (le traslazioni del treppiede sono all'incirca di 40 cm);
- In molte aree delle immagini, principalmente lungo il bordo inferiore, sono presenti soggetti in movimento come persone o biciclette;
- A causa degli spostamenti e del tempo impiegato per ottenere tutte le immagini, pari circa a venti minuti, in queste vi sono leggere variazioni dell'illuminazione, che portano a un cambio dell'intensità dei colori.

Dei due set di immagini ottenute viene utilizzato quello in formato *raw*, da convertire successivamente in TIFF tramite *software* dedicato. La conversione di formato viene effettuata con il *software* nativo Canon, "Digital Photo Professional", nella sua *release* 4.7.1.0. Per limitare il tempo computazionale, anche in questo caso tutti i test vengono svolti su immagini in scala di grigi e quindi le immagini, inizialmente in RGB, vengono semplificate in tale modello di colore, ottenendo un campione di osservazioni LR come la seguente:



Fig. 4.1 – Osservazione LR scattata con la Canon

L'algoritmo di ricostruzione viene poi lanciato su 32 immagini del set così ottenuto, nella combinazione di registrazione tramite algoritmo MATLAB (la registrazione è effettuata considerando l'intera immagine e cercando la miglior corrispondenza tra riferimento e *target*) e interpolazione tramite IDW. L'intenzione, in fase di registrazione, è quella di considerare solamente trasformazioni di tipo rigido, senza considerare cambi di prospettiva o fattori di scala, poiché per il tipo di spostamenti forniti il campione iniziale di osservazioni LR dovrebbe presentare solo trasformazioni di tale tipologia. In caso contrario, vi sarebbe una conferma che per una buona ricostruzione serva avere un *dataset* di immagini scattate/ottenute con alcune limitazioni e criteri.

L'immagine ricostruita così ottenuta non è oggettivamente accettabile a causa delle grandi distorsioni presenti, dato che in essa è possibile individuare facilmente alcune immagini registrate erroneamente rispetto a quella di riferimento. Oltretutto, le parti in cui sono presenti soggetti in movimento con forti cambi di posizione risultano estremamente caotiche e confuse, proprio a causa dell'assenza di un soggetto fisso.

Il tentativo di modifica dei parametri dell'algoritmo di registrazione *intensity-based* MATLAB non produce significative variazioni nella precisione dei risultati raggiunti, ma allo stesso tempo porta ad un eccessivo aumento dei tempi computazionali. Per tale motivo i parametri vengono mantenuti ai valori migliori determinati in sede di test preliminari. In figura 4.2 è riportato il risultato della ricostruzione errata:



Fig. 4.2 – Tentativo di ricostruzione errato

4.1.2. Fotogrammi da video in bassa risoluzione

Parallelamente a quanto appena descritto, vengono effettuati dei test anche su un campione di immagini a bassa risoluzione estremamente diverso, a causa delle diverse condizioni iniziali. Le immagini sono infatti ottenute utilizzando la fotocamera di un normale *smartphone*, impostata per registrare video con risoluzione di 640×480 pixel. Diversamente dalla situazione precedente, la scena ripresa è un muro interno del dipartimento con alcuni oggetti particolari da ricostruire (figura 4.3). Di seguito sono elencate le diverse accortezze utilizzate nell'ottenimento di tale set di immagini:

- Nella scena non sono presenti soggetti in movimento;
- Le traslazioni e le rotazioni sono di entità molto inferiore e sono introdotte fornendo piccoli movimenti circolari allo *smartphone*;
- Le condizioni di illuminazione rimangono pressoché invariate per tutta la durata del video.

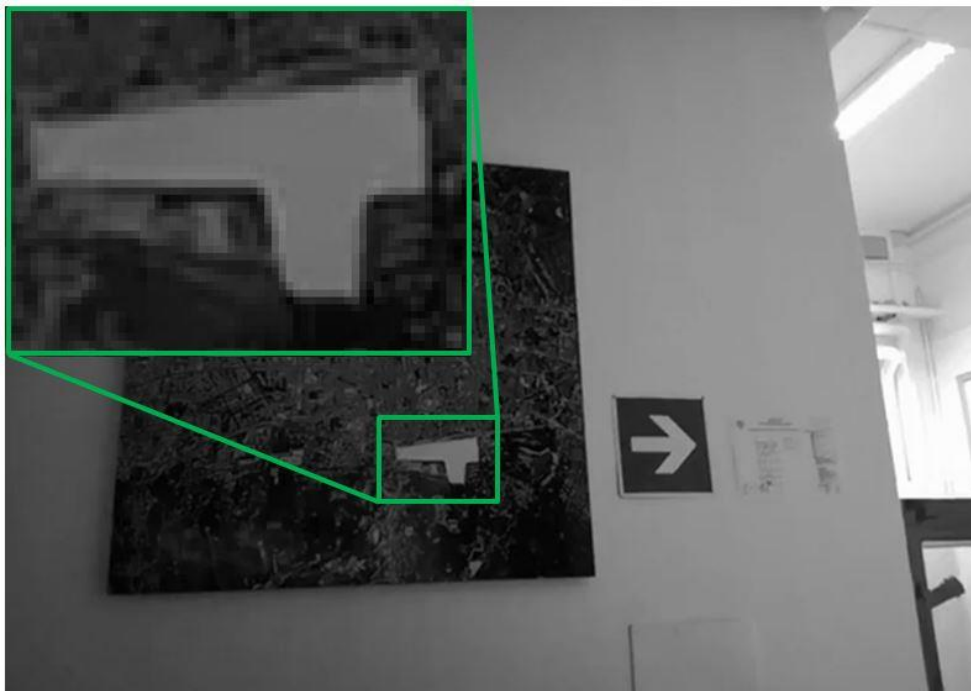


Fig. 4.3 – *Frame del video a bassa risoluzione*

Dal video, della durata di qualche secondo, vengono estratti con il *software* “VLC Media Player” una trentina di fotogrammi relativamente vicini, nell’arco dei quattro secondi di durata, l’uno all’altro. In tali fotogrammi, diversamente dal caso precedente, non compaiono forti influenze di fattori ambientali e la scena resta pressoché invariata, grazie anche alla

vicinanza degli istanti temporali di cattura. L'estrazione dei *frame* è effettuata salvando i dati in formato TIFF, per evitare distorsioni generate dal JPEG.

La ricostruzione dell'immagine ad alta risoluzione avviene come già specificato per il caso delle immagini dalla Canon. La metodologia utilizzata porta anche in questo caso ad un risultato, benché migliore del precedente, non accettabile. Uno dei problemi rilevati, sia in questo campione di immagini che nell'altro, è che la scena inquadra allo stesso tempo soggetti più vicini alla camera ed altri più lontani; per tale motivo piccoli movimenti dello *smartphone* si ripercuotono diversamente sui due gruppi di soggetti. Il risultato della registrazione effettuata sull'intera immagine è un generale effetto di *motion blur*, dato che nessuna immagine è registrata perfettamente sul riferimento (l'algoritmo MATLAB ricerca la miglior corrispondenza, che in questo caso è data da una sovrapposizione grossolana di tutta la scena, poiché una sovrapposizione perfetta di un solo particolare genererebbe un risultato peggiore sul resto).

Per tentare una ricostruzione più efficace per entrambi i *dataset*, i test vengono orientati sulla ricostruzione di una sola porzione dell'immagine, invece che su tutta quanta, per eliminare i problemi introdotti dalla presenza di soggetti in movimento immortalati in istanti eccessivamente lontani l'uno dall'altro o eccessivi cambi di prospettiva in zone non interessate dalla ricostruzione. Per poter effettuare una ricostruzione migliore viene inizialmente variato l'algoritmo di registrazione utilizzato (per poi variare successivamente la forma di quello *intensity-based*), passando alla cross-correlazione normalizzata nonostante l'eccessivo tempo che questa richiede. Il risultato ottenuto in tal modo risulta di gran lunga migliore del caso precedente, almeno per quanto riguarda l'area di interesse. Per poter ottenere un risultato analogo, ma in un tempo decisamente più contenuto, l'algoritmo di registrazione tramite *routine* MATLAB viene leggermente modificato, per poter effettuare dei ritagli sull'immagine riferimento o *target*. Invece di effettuare la registrazione sulle immagini intere, la scelta ricade su due differenti opzioni: registrare il *target* su un ritaglio del riferimento (posto automaticamente nell'angolo in alto a sinistra della griglia) oppure registrare il ritaglio del riferimento (mantenuto in questo caso in posizione originale) sul *target*, per poi invertire la matrice di trasformazione ottenuta ed applicarla al *target* stesso. Il secondo metodo permette di mantenere l'intera scena senza eliminarne parti, tenendo però conto che la registrazione è effettuata sul soggetto di interesse e quindi allontanandosi da esso aumentano le distorsioni. Il primo metodo porta invece all'eliminazione della porzione non

interessata dalla ricostruzione, dato che la sovrapposizione è effettuata solo sulla porzione selezionata, senza tener conto del resto.

La soluzione migliore appare inizialmente essere la seconda dato che, nonostante le traslazioni e le rotazioni di ogni immagine LR rispetto alle altre, il soggetto interessato dalla ricostruzione rimane all'incirca nella stessa posizione. In tal modo diminuisce il numero di iterazioni necessarie ad un primo posizionamento grossolano del *target* rispetto al riferimento, con il risultato di una registrazione più precisa. Tale tecnica presenta però un problema legato alla struttura delle matrici coinvolte nell'operazione, dato che la zona attorno al ritaglio della registrazione è completamente campita in nero (valore 0 nelle matrici) a causa delle rotazioni, fattore che porta a falsare la registrazione di alcune immagini.

La scelta definitiva sull'algorithm da utilizzare ricade quindi sulla registrazione diretta del *target* effettuata su un riquadro del riferimento traslato nell'angolo superiore sinistro, accettando la perdita di una porzione, comunque non significativa, dell'immagine.

Tale soluzione permette di ricostruire con sufficiente precisione l'immagine del *dataset* ottenuto dal video girato con lo *smartphone* riportata in figura 4.4, mentre nel caso delle immagini dalla Canon in fase di registrazione gli algoritmi non sono in grado di spostare correttamente tutto il set di immagini, con il risultato di alcune immagini ben sovrapposte ed altre completamente sbagliate.



Fig. 4.4 – Ricostruzione con registrazione *intensity-based* su ritaglio

4.2. Verifica di registrazione

Sulla base della disponibilità di un numero di osservazioni LR in esubero rispetto al minimo necessario e sufficiente, si ricerca un criterio statistico che permetta di accettare o rifiutare una particolare registrazione, in modo da eliminare l'influenza di eventuali immagini registrate erroneamente sul risultato finale. Escludendo fin da subito la possibilità di una semplice verifica visiva del risultato di registrazione, a causa della scarsa obiettività e precisione riscontrabili, nasce la necessità di elaborare uno *script* automatizzato, che operi secondo un criterio da definire.

Per mantenere oggettività nel processo di verifica ed evitare di dare maggiore importanza ad un'osservazione piuttosto che un'altra, la base del criterio sta nella determinazione dei valori degli scarti normalizzati dei *digital number* delle singole immagini LR rispetto al valore medio di quelli nella stessa posizione di tutte le immagini, secondo la seguente formula:

$$s_{norm,i,j} = \frac{I_{i,j} - \bar{I}_i}{\sigma_i}$$

in cui $s_{norm,i,j}$ è lo scarto normalizzato del *digital number* $I_{i,j}$ in posizione i -esima dell'immagine registrata j -esima e \bar{I}_i e σ_i sono rispettivamente media e deviazione standard dei valori di *digital number* in posizione i di tutte le immagini.

Tale metodologia, di natura puramente statistica, permette di individuare eventuali *outlier* in un campione di dati, poiché ad essi sono associati gli scarti normalizzati a valore assoluto più alto.

È possibile effettuare il calcolo degli scarti normalizzati secondo la formula indicata per ogni pixel della griglia delle immagini, quindi per un'innumerabile quantità di punti. Inizialmente l'idea è quella di prendere un certo numero di punti casuali sull'immagine, circa 5 o 6, e per ognuno di essi calcolare gli scarti normalizzati di tutte le immagini registrate. Il limite di tale modalità di applicazione sta nella scelta oggettiva di un criterio di accettazione/rifiuto, dato che risulta difficile determinare un numero minimo di punti per i quali uno scarto normalizzato alto relativo ad un'immagine porti al rifiuto della stessa. Applicando la verifica in questa forma ad un set di immagini, al cui interno ne è inserita una precedentemente modificata per risultare errata, al diminuire dell'errore fornito all'immagine diminuisce drasticamente l'efficacia dello *script*: se per traslazioni/rotazioni di grande entità l'immagine con lo scarto normalizzato più grande in tutti i punti di verifica è proprio quella sbagliata, al diminuire dell'errore introdotto non tutti i punti di verifica forniscono come immagine

registrata in modo peggiore quella sbagliata (per errori introdotti di traslazione verticale e orizzontale di 2 pixel solo due punti evidenziano come quell'immagine sia errata).

Per tale motivo, nonostante il concetto alla base sia efficace, va modificata la modalità della sua applicazione. In un secondo tentativo, il calcolo degli scarti normalizzati è effettuato su tutta la matrice e non più solo su alcuni punti di essa, ottenendo in tal modo numerosi valori di scarti normalizzati all'interno di matrici, una per ogni immagine registrata. Il confronto è effettuato sommando per ogni immagine i valori degli scarti normalizzati relativi ad essa. Successivamente tutte le somme, tante quante le osservazioni LR, vengono nuovamente normalizzate, permettendo in tal modo l'individuazione di *outlier* nel campione a disposizione. L'immagine cui è associato lo scarto normalizzato della somma degli scarti normalizzati più alto è infatti quella che più si discosta rispetto alle altre.

Definito un valore di soglia per l'accettazione/rifiuto, una eventuale immagine con valore superiore ad esso viene eliminata dal set e la verifica si ripete nuovamente ricalcolando i valori di scarto normalizzato e riconfrontandoli con tale valore di soglia. Il set di immagini ottenuto all'ultima iterazione, in cui tutti i valori di scarto normalizzato sono inferiori alla soglia, è quello effettivamente proiettato nella griglia ad alta risoluzione ed utilizzato per la ricostruzione.

Lo scopo dei primi test sull'utilizzo dell'algoritmo non è tanto quello di individuare un valore di soglia corretto, quanto quello di verificare che, fornito in input un set di immagini al cui interno ve ne siano una o più volutamente traslate di alcuni pixel o ruotate di porzioni di grado, queste ultime vengano individuate ed abbiano quindi lo scarto normalizzato della somma degli scarti normalizzati più alta di tutti. La conferma di quanto indagato si ha sin dalle prime prove, dato che la presenza di poche immagini errate all'interno di un set molto grande viene individuata ed associata ad un valore di scarto normalizzato molto alto, ben superiore a 2.5. Un limite è che, all'aumentare del numero di immagini registrate erroneamente all'interno del set, come ad esempio nel caso di quelle ottenute dalla Canon, in cui praticamente più della metà delle immagini risultano sbagliate, ciò che accade è l'opposto: le immagini tecnicamente da eliminare vengono considerate come corrette, mentre invece quelle ad essere classificate come sbagliate sono proprio quelle ben registrate. Tale situazione è la ovvia conseguenza della necessità di avere immagini ben registrate per eseguire la ricostruzione: il set di immagini di partenza deve rispondere ad una serie di requisiti, come già ripetuto, condizione che non sussiste nel set di immagini citato. La presenza di poche

immagini registrate erroneamente per i più disparati motivi è infatti ben identificabile in un grande set di immagini, mentre se ad essere errata è la quasi totalità del set, queste non sono identificabili.

4.2.1. Test chi quadro

Definite le condizioni al contorno dell'algorithmo, la difficoltà sta nell'individuazione di un valore corretto della soglia di accettazione/rifiuto delle immagini con cui confrontare gli scarti normalizzati calcolati. Tale valore è fortemente dipendente dalla tipologia di funzione di distribuzione di probabilità $f(s)$ che gli scarti assumono.

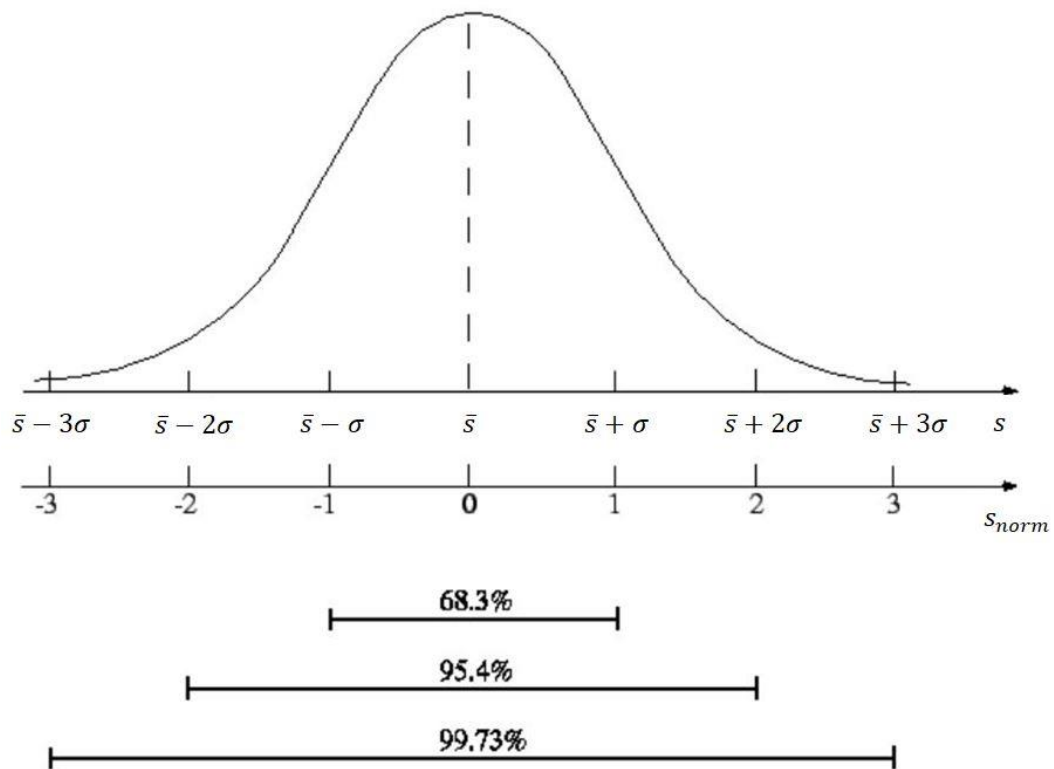


Fig. 4.5 – Ipotetica distribuzione normale di Gauss degli scarti

Se infatti questi avessero una distribuzione casuale e quindi la funzione fosse una distribuzione di probabilità di Gauss in forma

$$f(s) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}}$$

il valore di soglia limite al cui interno ricadrebbe il 95% degli scarti normalizzati delle somme degli scarti normalizzati corretti sarebbe pari a 2, come rappresentato anche in figura 4.1.

L'appartenenza degli scarti a una distribuzione normale di Gauss rappresenta dunque l'ipotesi di partenza H_0 da verificare tramite un test di significatività, il test χ^2 (chi quadro). Quest'ultimo serve a saggiare l'ipotesi che una certa discrepanza tra frequenze attese e frequenze osservate sia dovuta a:

- H_0 : all'appartenenza della popolazione alla distribuzione di ipotesi, quindi al caso;
- H_1 : al fatto che il campione provenga da una popolazione diversa rispetto a quella da cui deriva la frequenza attesa.

Il test consiste nel calcolo del rapporto qui descritto:

$$\chi^2 = \frac{(\text{frequenze osservate} - \text{frequenze attese})^2}{\text{frequenze attese}}$$

dove per frequenze si intendono sempre le frequenze assolute, dato che l'utilizzo di quelle relative non è mai possibile.

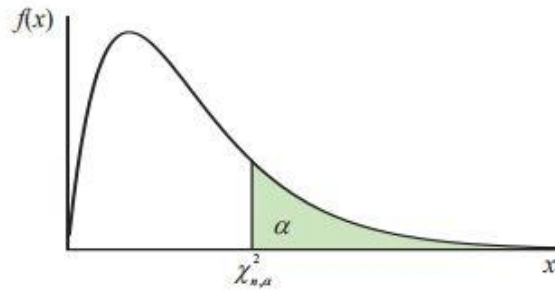
Per testare l'adattamento dei dati alla distribuzione ipotizzata basta confrontare le frequenze osservate sugli scarti normalizzati delle somme degli scarti normalizzati con le frequenze attese che una distribuzione di Gauss con quella media e deviazione standard avrebbe. L'ottenimento di valori osservati esattamente pari a quelli attesi è praticamente impossibile, mentre valori molto vicini a quelli teorici hanno buone probabilità di essere "variazioni casuali" degli stessi. Diversamente, valori molto lontani da quelli teorici indicano una bassa probabilità di fluttuazioni casuali. Il calcolo del valore χ^2 è riassumibile in questa sequenza:

- 1) Calcolo del valore medio teorico (\bar{x}) e della deviazione standard (σ) della popolazione osservata;
- 2) Determinazione delle frequenze della popolazione osservata all'interno delle classi di frequenza;
- 3) Calcolo di una distribuzione del tipo in ipotesi H_0 con valori di media e varianza \bar{x} e σ ;
- 4) Determinazione delle frequenze della popolazione attesa all'interno delle stesse classi di frequenza;
- 5) Calcolo del quadrato dello scarto della frequenza osservata rispetto a quella attesa (per eliminare i valori negativi e amplificare le differenze proporzionalmente alla loro grandezza);
- 6) Divisione per la frequenza teorica, per standardizzare le distanze;

7) Sommatoria di tutti i valori di χ^2 così calcolati.

Il valore così calcolato rappresenta la somma ponderata degli scarti delle frequenze di ciascuna classe rispetto alla sua attesa teorica. Per tale motivo, risulta immediato che maggiore è il valore trovato e maggiore è lo scostamento tra le frequenze osservate e quelle attese.

Il test vero e proprio viene effettuato confrontando il valore ottenuto con un valore teorico, funzione del numero di gradi di libertà, e quindi del numero delle classi, in esame. In un problema statistico il numero di gradi di libertà ν è definito come il numero di dati osservati n meno il numero dei parametri calcolati dai dati e usati nel calcolo (vincoli). Nel caso osservato il numero di gradi di libertà è dunque pari al numero delle classi meno 2, ovvero il numero di classi vincolate a contenere determinati elementi (la prima e l'ultima). In funzione del numero di gradi di libertà del problema, dalla tabella 4.1 della distribuzione dei quantili del χ^2 è possibile ricavare il valore da confrontare con il valore ottenuto. Se i due sono fra loro vicini l'ipotesi H_0 è confermata, in caso contrario è rifiutata.



v	α									
	0.995	0.990	0.975	0.950	0.900	0.100	0.050	0.025	0.010	0.005
1	0.0 ⁴ 393	0.0 ³ 157	0.0 ³ 982	0.0 ² 393	0.0158	2.71	3.84	5.02	6.63	7.88
2	0.0100	0.0201	0.0506	0.103	0.211	4.61	5.99	7.38	9.21	10.60
3	0.072	0.115	0.216	0.352	0.584	6.25	7.81	9.35	11.34	12.84
4	0.207	0.297	0.484	0.711	1.06	7.78	9.49	11.14	13.28	14.86
5	0.412	0.554	0.831	1.145	1.61	9.24	11.07	12.83	15.09	16.75
6	0.676	0.872	1.24	1.64	2.20	10.64	12.59	14.45	16.81	18.55
7	0.989	1.24	1.69	2.17	2.83	12.02	14.07	16.01	18.48	20.28
8	1.34	1.65	2.18	2.73	3.49	13.36	15.51	17.53	20.09	21.95
9	1.73	2.09	2.70	3.33	4.17	14.68	16.92	19.02	21.67	23.59
10	2.16	2.56	3.25	3.94	4.87	15.99	18.31	20.48	23.21	25.19
11	2.60	3.05	3.82	4.57	5.58	17.28	19.68	21.92	24.72	26.76
12	3.07	3.57	4.40	5.23	6.30	18.55	21.03	23.34	26.22	28.30
13	3.57	4.11	5.01	5.89	7.04	19.81	22.36	24.74	27.69	29.82
14	4.07	4.66	5.63	6.57	7.79	21.06	23.68	26.12	29.14	31.32
15	4.60	5.23	6.26	7.26	8.55	22.31	25.00	27.49	30.58	32.80
16	5.14	5.81	6.91	7.96	9.31	23.54	26.30	28.85	32.00	34.27
17	5.70	6.41	7.56	8.67	10.09	24.77	27.59	30.19	33.41	35.72
18	6.26	7.01	8.23	9.39	10.86	25.99	28.87	31.53	34.81	37.16
19	6.84	7.63	8.91	10.12	11.65	27.20	30.14	32.85	36.19	38.58
20	7.43	8.26	9.59	10.85	12.44	28.41	31.41	34.17	37.57	40.00
21	8.03	8.90	10.28	11.59	13.24	29.62	32.67	35.48	38.93	41.40
22	8.64	9.54	10.98	12.34	14.04	30.81	33.92	36.78	40.29	42.80
23	9.26	10.20	11.69	13.09	14.85	32.01	35.17	38.08	41.64	44.18
24	9.89	10.86	12.40	13.85	15.66	33.20	36.42	39.36	42.98	45.56
25	10.52	11.52	13.12	14.61	16.47	34.38	37.65	40.65	44.31	46.93
26	11.16	12.20	13.84	15.38	17.29	35.56	38.89	41.92	45.64	48.29
27	11.81	12.88	14.57	16.15	18.11	36.74	40.11	43.19	46.96	49.64
28	12.46	13.56	15.31	16.93	18.94	37.92	41.34	44.46	48.28	50.99
29	13.12	14.26	16.05	17.71	19.77	39.09	42.56	45.72	49.59	52.34
30	13.79	14.95	16.79	18.49	20.60	40.26	43.77	46.98	50.89	53.67
40	20.71	22.16	24.43	26.51	29.05	51.81	55.76	59.34	63.69	66.77
50	27.99	29.71	32.36	34.76	37.69	63.17	67.50	71.42	76.15	79.49
60	35.53	37.48	40.48	43.19	46.46	74.40	79.08	83.30	88.38	91.95
70	43.28	45.44	48.76	51.74	55.33	85.53	90.53	95.02	100.4	104.2
80	51.17	53.54	57.15	60.39	64.28	96.58	101.9	106.6	112.3	116.3
90	59.20	61.75	65.65	69.13	73.29	107.6	113.2	118.1	124.1	128.3
100	67.33	70.06	74.22	77.93	82.36	118.5	124.3	129.6	135.8	140.2

Tab. 4.1 – Valori dei quantili della distribuzione χ^2

Per l'esecuzione del test si utilizzano le statistiche relative alle immagini del set ottenuto dai fotogrammi del video girato con lo *smartphone*. La popolazione è dunque composta da 42 valori di scarti normalizzati, da distribuire in una serie di classi piccole a piacere. Nella seguente tabella sono riportati i risultati del test:

Classi	Frequenza osservata	Classi normalizzate	Frequenza relativa cumulata	Frequenze assolute attese	Chi quadro
2.80	1	2.80	0.997	0.088	9.394
2.60	0	2.60	0.995	0.149	0.149
2.40	0	2.40	0.992	0.240	0.240
2.20	0	2.20	0.986	0.372	0.372
2.00	0	2.00	0.977	0.554	0.554
1.80	0	1.80	0.964	0.792	0.792
1.60	1	1.60	0.945	1.090	0.007
1.40	5	1.40	0.919	1.441	8.788
1.20	3	1.20	0.885	1.831	0.747
1.00	2	1.00	0.841	2.234	0.025
0.80	1	0.80	0.788	2.621	1.002
0.60	0	0.60	0.726	2.954	2.954
0.40	5	0.40	0.655	3.199	1.014
0.20	2	0.20	0.579	3.329	0.531
0.00	2	0.00	0.500	3.329	0.531
-0.20	1	-0.20	0.421	3.199	1.511
-0.40	1	-0.40	0.345	2.954	1.292
-0.60	6	-0.60	0.274	2.621	4.357
-0.80	4	-0.80	0.212	2.234	1.395
-1.00	4	-1.00	0.159	1.831	2.571
-1.20	4	-1.20	0.115	1.441	4.543
-1.40	0	-1.40	0.081	1.090	1.090
-1.60	0	-1.60	0.055	0.792	0.792
-1.80	0	-1.80	0.036	0.554	0.554
-2.00	0	-2.00	0.023	0.372	0.372
-2.20	0	-2.20	0.014	0.240	0.240
-2.40	0	-2.40	0.008	0.149	0.149
-2.60	0	-2.60	0.005	0.088	0.088
-2.80	0	-2.80	0.003	0.107	0.107
	42				<u>46.161</u>

Tab. 4.2 – Risultati del test chi quadro

La somma dei valori di χ^2 su tutte le classi fornisce il valore 46.161, da confrontare con il valore di significatività relativo al quantile scelto. Considerando $\nu = 27$ gradi di libertà, la

soglia di significatività della funzione chi quadro relativo ad una percentuale dell'5% è pari a 40.11, inferiore al valore calcolato. Per indagare più a fondo l'ipotesi di partenza si considera anche il quantile 1%, cui è associato un valore di χ^2 pari a 46.96. Il valore calcolato è di poco inferiore a quello teorico e quindi vi è circa l'1% di possibilità che la popolazione considerata appartenga a una distribuzione normale. Per tale motivo l'ipotesi H_0 è respinta e si assume che la distribuzione degli scarti normalizzati così calcolati per il *dataset* considerato non appartenga a una distribuzione normale.

4.2.2. Disuguaglianza di Čebyšëv

Poiché la popolazione degli scarti normalizzati delle somme degli scarti normalizzati non appartiene a una distribuzione di Gauss, risulta necessario trovare un altro criterio per determinare un valore di soglia. Rimanendo nell'ambito della teoria della probabilità, si ricorre alla cosiddetta “disuguaglianza di Čebyšëv”, che afferma:

$$\Pr(|X - \bar{x}| \geq \lambda\sigma) \leq \frac{1}{\lambda^2}$$

ovvero, detto λ un numero reale positivo, X una variabile casuale a varianza σ^2 e media \bar{x} , la probabilità che X assuma un valore compreso tra $\bar{x} - \lambda\sigma$ e $\bar{x} + \lambda\sigma$ è maggiore di $1 - 1/\lambda^2$. Da questa disuguaglianza è possibile dedurre gli intervalli di confidenza, validi per qualunque distribuzione statistica, relativi a multipli della deviazione standard. In particolare si deduce che almeno il 94% dei valori di una popolazione sono compresi nell'intervallo con estremi $\bar{x} - 4\sigma$ e $\bar{x} + 4\sigma$.

Da tale affermazione è possibile assegnare un valore alla soglia di accettazione/rifiuto delle immagini registrate, tenendo comunque presente che esso è estremamente alto, con il rischio dunque di non scartare eventuali immagini errate.

Ci si riserva comunque di imporre una soglia pari a 3 (corrispondente all'89% dei valori) o, eventualmente, anche leggermente più bassa, dato che una possibile diminuzione è strettamente legata alla quantità di osservazioni LR disponibili all'interno del set, dato che con molte osservazioni a disposizione è possibile scartarne molte senza effettivamente intaccare il risultato della ricostruzione, diversamente da una situazione con un numero di osservazioni LR limitato o minimo.

4.2.3. Verifiche seriali su set di immagini

Definita la soglia di accettazione/rifiuto, l'algoritmo di verifica della registrazione è operativo ed in grado di eliminare eventuali immagini con scarti normalizzati eccessivamente alti. Rimane da capire quale possa essere l'influenza che tale algoritmo ha sul risultato finale della ricostruzione. Infatti, oltre ad eliminare immagini registrate erroneamente, è possibile che alcune immagini vengano eliminate a causa di valori di *digital number* eccessivamente diversi da quelli medi, ma non necessariamente sbagliati. La verifica viene ora eseguita in modo particolare: invece di eliminare le immagini confrontando gli scarti normalizzati con un valore di soglia, quest'ultimo è posto pari a 100, in modo da non essere superabile, e, ripetendo ciclicamente il calcolo degli scarti normalizzati, di volta in volta viene eliminata dal set l'immagine che presenta lo scarto normalizzato più alto, fino ad arrivare ad una ricostruzione con 3 immagini. Una volta ricostruita N volte l'immagine, ogni volta utilizzando un'osservazione LR registrata in meno, i risultati ricostruiti vengono confrontati a coppie tra loro calcolando le matrici di scarto e da esse i valori di somma degli scarti, media degli scarti e massimo degli scarti.

Il set di immagini utilizzato è quello ottenuto dallo *smartphone*, con la differenza dell'introduzione di tre immagini volutamente registrate in modo erroneo. Infatti le immagini 10, 15 e 20 hanno subito rispettivamente una traslazione di 5 pixel in orizzontale e verticale, una rotazione di 0.5° in senso orario e una traslazione di 2 pixel in orizzontale e verticale. La prova viene eseguita due volte, la prima inserendo le tre immagini modificate nel set di 42 immagini, la seconda considerando solamente le tre immagini errate e 20 immagini tecnicamente "ben registrate". I risultati sono riportati nei grafici delle seguenti figure, nelle cui ascisse è indicato il numero di immagini eliminate per ogni ricostruzione (il primo *marker* si riferisce allo scarto della statistica in esame tra l'immagine ricostruita utilizzando tutte le osservazioni LR a disposizione e quella ricostruita eliminando quella con lo scarto normalizzato più alto:

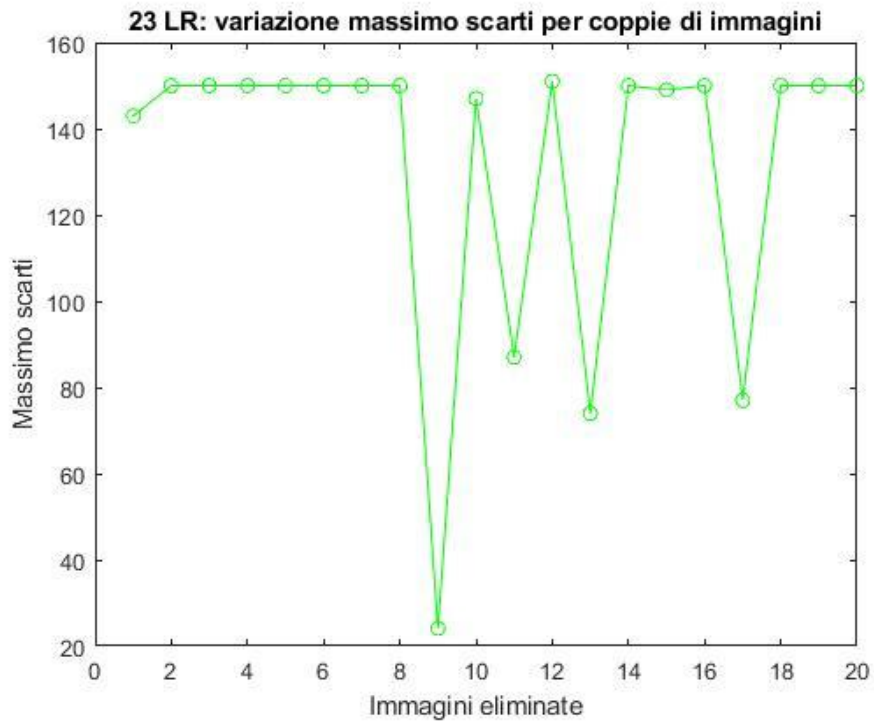


Fig. 4.6a – *Massimi degli scarti con 23 osservazioni LR*

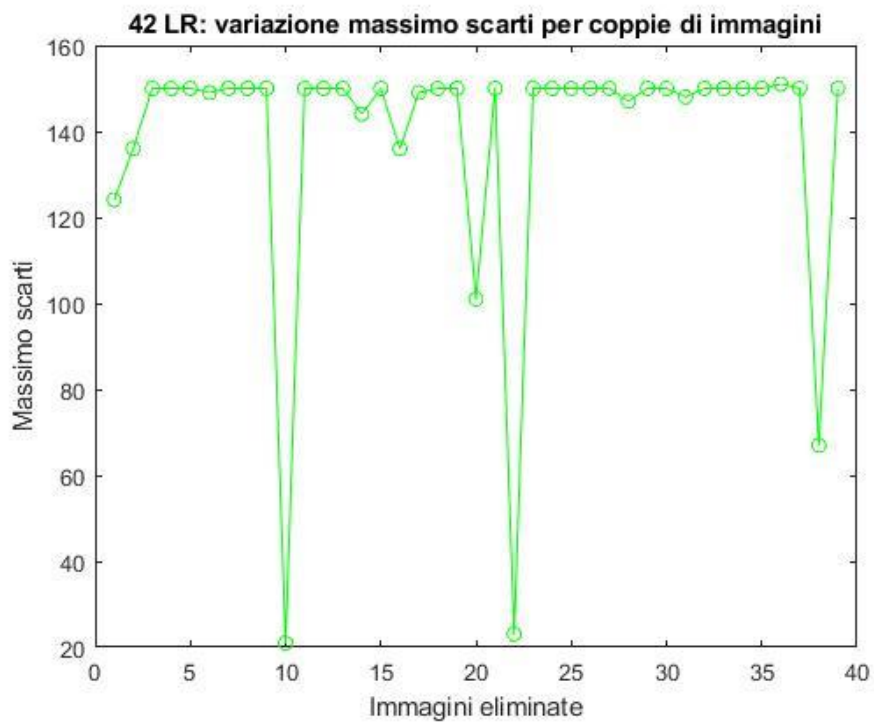


Fig. 4.6b - *Massimi degli scarti con 42 osservazioni LR*

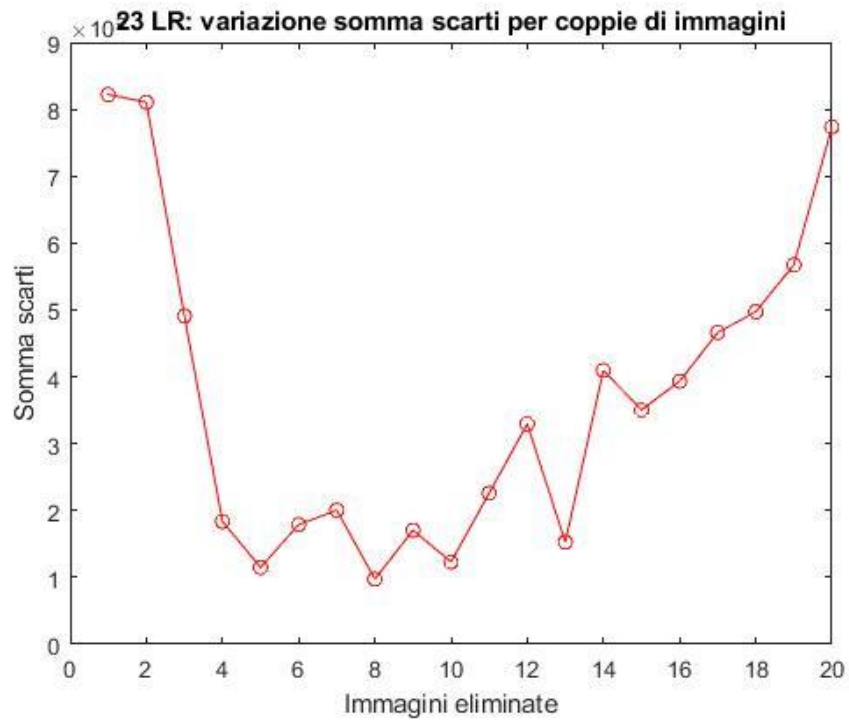


Fig. 4.6c – *Somma degli scarti con 23 osservazioni LR*

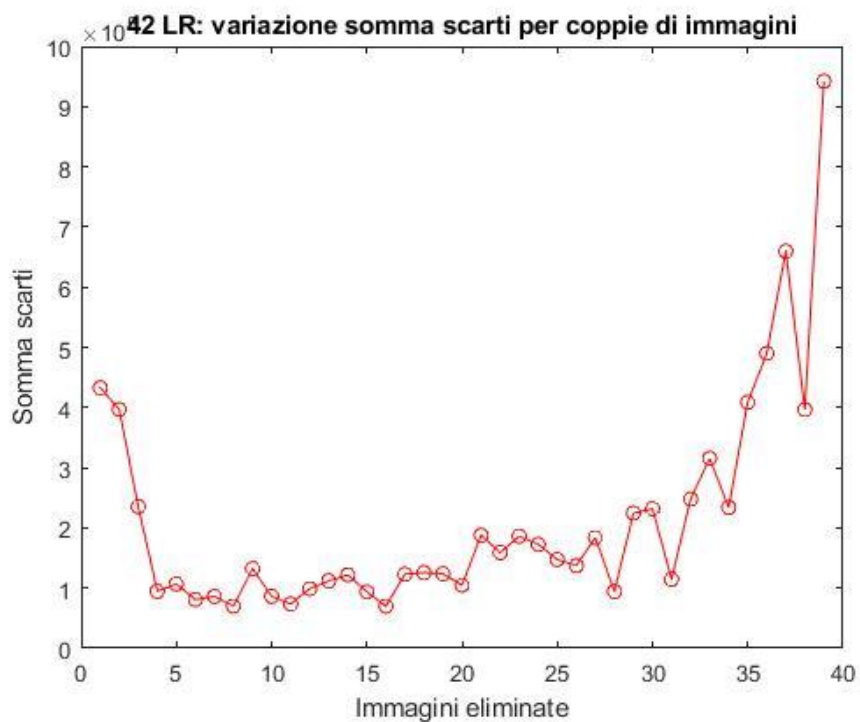


Fig. 4.6d - *Somma degli scarti con 42 osservazioni LR*

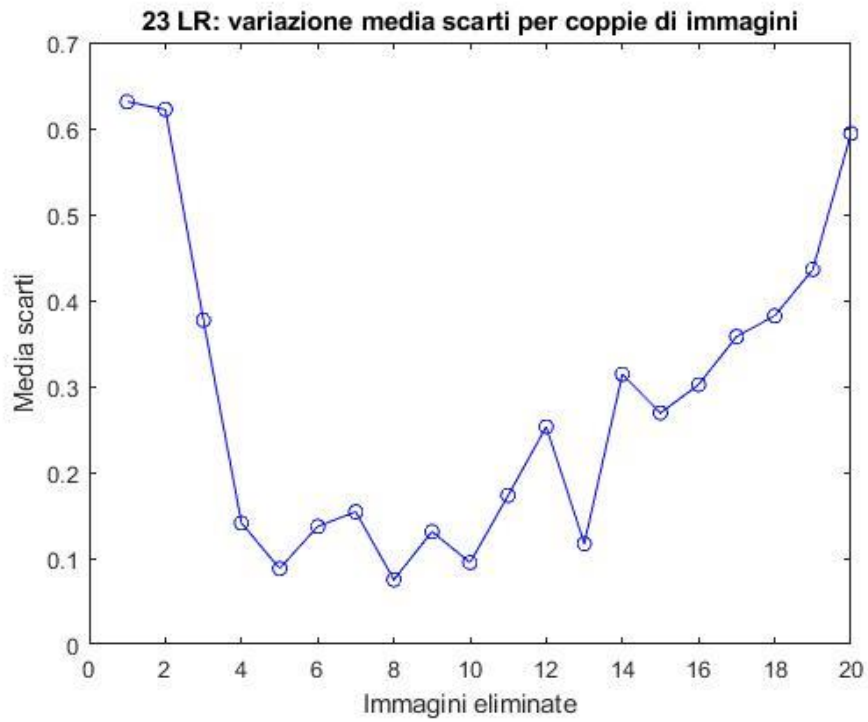


Fig. 4.6e – *Medie degli scarti con 23 osservazioni LR*

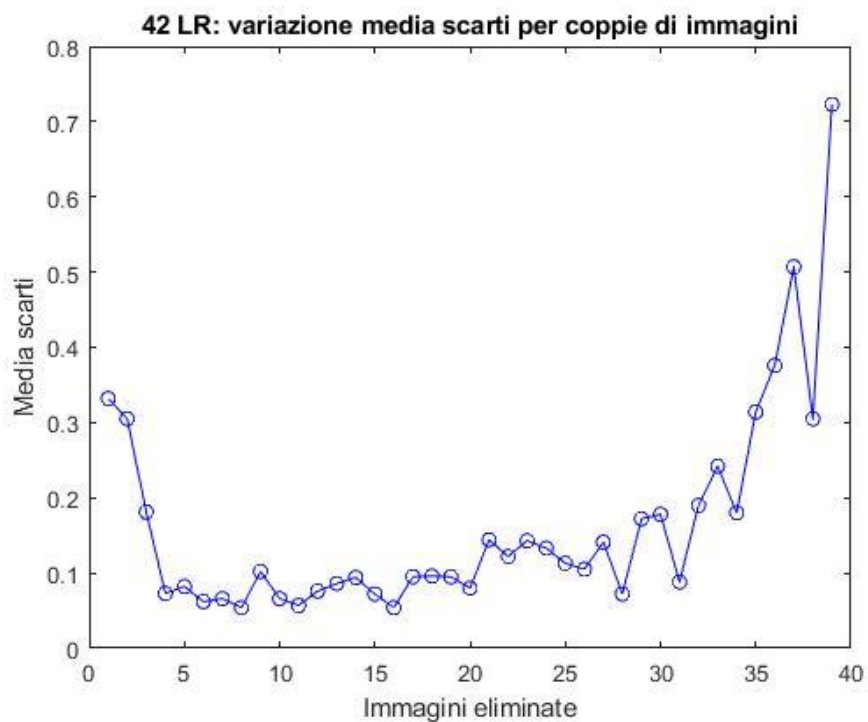


Fig. 4.6f - *Medie degli scarti con 42 osservazioni LR*

Da quanto appena mostrato si evince come l'eliminazione di immagini registrate erroneamente, rispettivamente i primi tre *marker* di ogni grafico, porti a un netto miglioramento nella somma e nella media degli scarti, mentre per quanto riguarda il loro massimo non vi sia una sostanziale influenza poiché, come detto in precedenza, tali picchi si localizzano tutti sui bordi netti che, non venendo ricostruiti, generano picchi molto elevati (la presenza di scarti massimi molto bassi è giustificata dalla scarsa importanza che l'immagine eliminata aveva sull'intera ricostruzione). È anche possibile notare come il miglioramento sulla somma e sulla media degli scarti sia molto più netto nel caso di 23 osservazioni LR nel set di partenza rispetto al caso con 42. Una volta eliminate le immagini errate in entrambi i casi la somma e la media degli scarti si attestano più o meno sullo stesso valore in entrambi i casi, per poi ricrescere a seguito dell'eliminazione di un eccessivo numero di immagini, confermando quanto descritto al paragrafo 3.5.4.

L'utilizzo di un algoritmo di verifica di registrazione risulta dunque estremamente efficace in presenza di immagini errate (con errori non necessariamente rilevabili a occhio nudo), mentre in presenza di set di immagini molto omogenee, per quanto ve ne possano essere alcune a scarto normalizzato superiore alla soglia, non porta a miglioramenti apprezzabili. La preferenza per le applicazioni future è dunque quella di mantenere l'utilizzo dell'algoritmo di verifica della registrazione, a patto però che i *dataset* di osservazioni LR utilizzati per la ricostruzione contengano un numero di immagini ben superiore al minimo necessario e sufficiente.

4.3. Osservazioni termiche in bassa risoluzione

L'ultima serie di applicazioni degli algoritmi creati coinvolge proprio l'obiettivo della sperimentazione, le immagini termiche.

La camera utilizzata è una FLIR P620, una macchina che permette una risoluzione massima infrarossa (IR) di 640×480 pixel con una sensibilità termica di 40 mK a $30 \text{ }^\circ\text{C}$, un'accuratezza di $\pm 2 \text{ }^\circ\text{C}$ e un intervallo di temperatura da $-40 \text{ }^\circ\text{C}$ a $+500 \text{ }^\circ\text{C}$.

Per mantenere le condizioni operative specificate nei paragrafi precedenti le immagini vengono ottenute ponendo la termocamera montata su un supporto con slitta, montato a sua volta su un treppiede con la possibilità di alzare/abbassare la base. In tal modo è possibile introdurre nella scena piccole traslazioni orizzontali e verticali, senza inserire però grosse rotazioni ed evitando dunque i problemi già sorti in precedenza. La prima scena allestita per la

sperimentazione consiste in una serie di luci a incandescenza poste su un vetro comunicante con l'esterno, per generare contrasto di temperatura tra i soggetti presenti. Le immagini vengono scattate da una griglia di posizioni composta da 4 traslazioni verticali ed una dozzina di traslazioni orizzontali. Gli spostamenti orizzontali forniti alla macchina sono dell'ordine del centimetro, mentre quelli verticali dell'ordine della decina di centimetri. L'intero set, rappresentante la scena indicata in figura 4.7, consta di 49 osservazioni a bassa risoluzione.

Le immagini dalla termocamera, salvate automaticamente in formato JPG sulla memoria SD, vengono successivamente convertite in formato TIFF, con al proprio interno tutte le informazioni sulle radiometrie. La conversione di formato è effettuata tramite il *software* "ResearchIR", nativo FLIR, con licenza limitata a 30 giorni (il *software* non è limitato alla sola conversione, ma mette a disposizione numerose altre funzioni per elaborazione di immagini termiche).

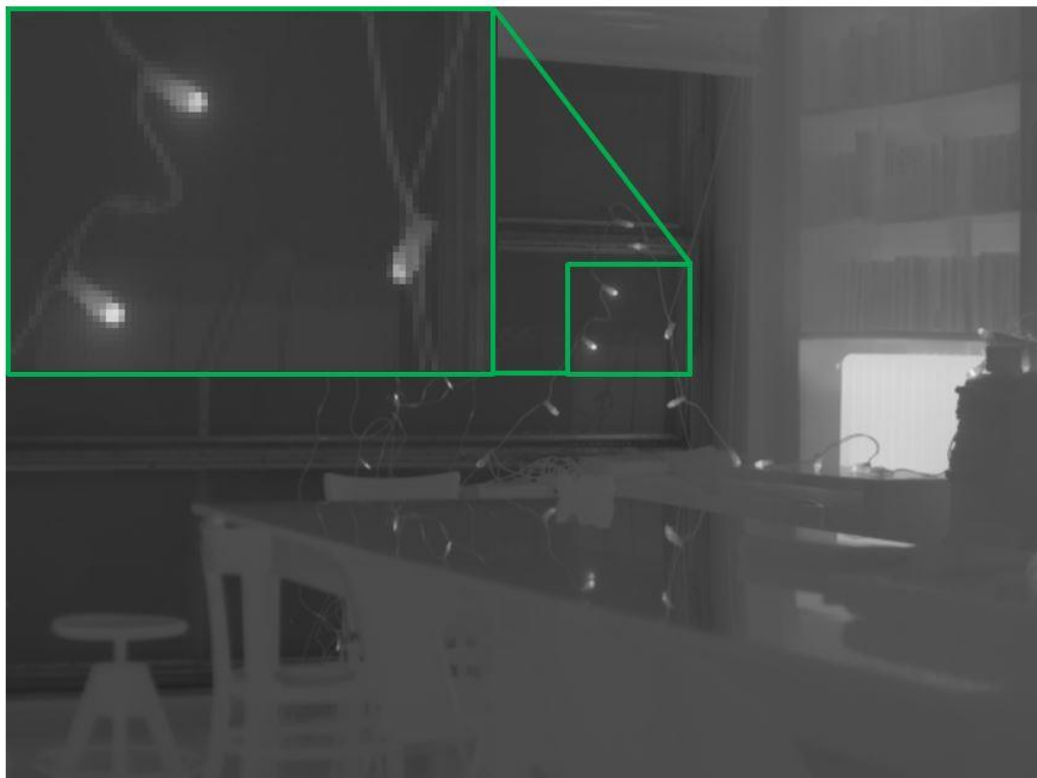


Fig. 4.7 – Immagine termica a bassa risoluzione

È da specificare come l'immagine visualizzata sia una semplice rappresentazione del reale contenuto della matrice ottenuta in fase di scatto. Infatti, il reale oggetto di interesse non è l'immagine in sé, ma la matrice da cui questa è generata. In essa è infatti riportata la radiometria dell'intera scena, quindi le temperature rilevate per ogni pixel. Poiché l'interesse

è aumentare la risoluzione di ottenimento di tale temperature (ciò che conta non è la visualizzazione, ma il dato) l'intera ricostruzione non è effettuata su file JPG come quello riportato, ma su file di testo contenenti tutte le temperature. Il numero di cifre decimali fornite dalla termocamera è tecnicamente molto elevato per l'applicazione ricercata e quindi, per limitare il tempo computazionale, il numero di cifre significative decimali nelle matrici calcolate viene limitato a 2 (il tempo impiegato per il salvataggio delle matrici aumenta linearmente all'aumentare delle cifre significative).

La ricostruzione viene effettuata con la combinazione di algoritmi specificata in precedenza, su un ritaglio della scena, e con una soglia di rigetto in fase di verifica delle immagini registrate pari a 2.5. Il soggetto della ricostruzione è in particolare la zona di finestra in cui sono presenti le luci ad incandescenza ed una porzione del termosifone, riportata nella sua forma originale nell'immagine come la porzione più incandescente di tutta la scena. Con i parametri specificati, di tutto il set vengono scartate 5 osservazioni registrate, considerate dunque non coerenti con il resto del set. Particolare di notevole importanza, a cui fare anche attenzione in fase di registrazione, è che la presenza di oggetti riflettenti la radiazione infrarossa può portare a ripetizioni dei soggetti nella scena, come si può osservare nel tavolo al centro.

Il risultato del procedimento di ricostruzione è riportato nella figura 4.8, in cui viene evidenziato anche il particolare delle luci da confrontare con l'immagine precedente.

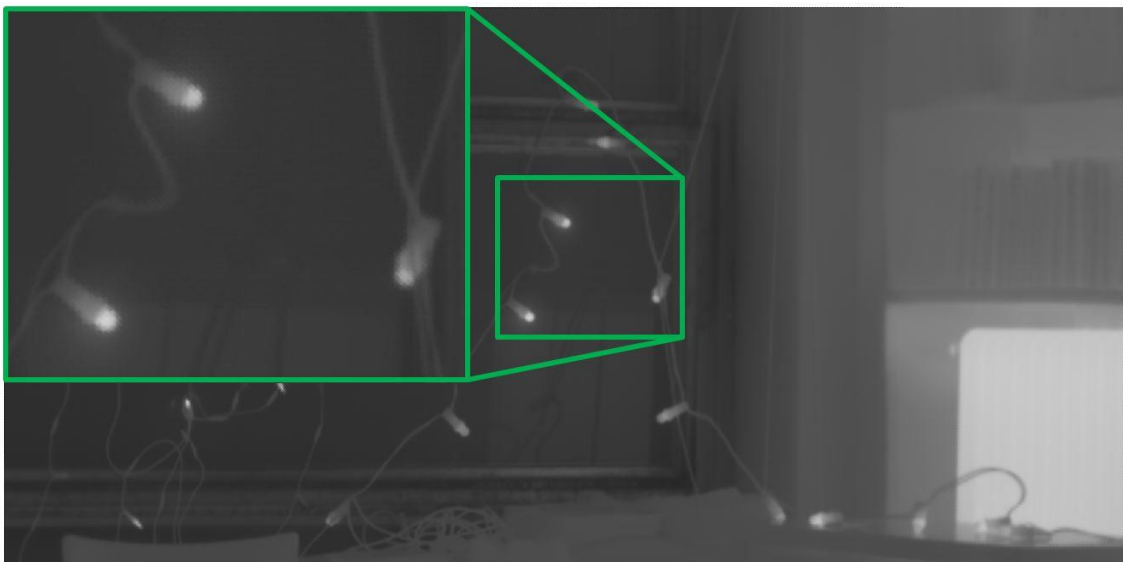


Fig. 4.8 – *Immagine termica ricostruita, con particolare*

La visualizzazione sia nel caso dell'osservazione LR che in quello dell'immagine ricostruita è in scala di grigi a causa delle condizioni operative specificate in precedenza: il risultato ottenuto è infatti una matrice di temperature, convertita successivamente in immagine scalando i valori con un fattore moltiplicativo pari al rapporto $\frac{255}{intensità\ massima}$ (255 è il massimo valore di *digital number* nel formato a 8 bit).

Dal confronto tra le due immagini emerge fin da subito un netto miglioramento della qualità di visualizzazione, sia prendendo in esame il particolare riportato, che osservando anche come altri elementi, come i cavi elettrici, siano meglio visualizzati.

Con le stesse accortezze e specifiche utilizzate per la precedente immagine, sono state ricostruite anche altre immagini, come riportato nelle figure 4.9 e 4.10, rispettivamente l'osservazione a bassa risoluzione e quella a risoluzione aumentata. In questo caso il soggetto è costituito da una serie di cuscinetti a sfera e delle *attache* metalliche poste su un supporto di legno, riscaldato da una lampada posta al di sopra (in fase di pianificazione l'idea era che il metallo riflettesse la radiazione infrarossa meglio del legno).

In questo caso entrambe le immagini vengono modificate per consentirne una buona visualizzazione ed enfatizzare il contrasto tra piccoli oggetti e sfondo omogeneo, (il fattore è calcolato su un riquadro di poco più grande rispetto a quello evidenziato).

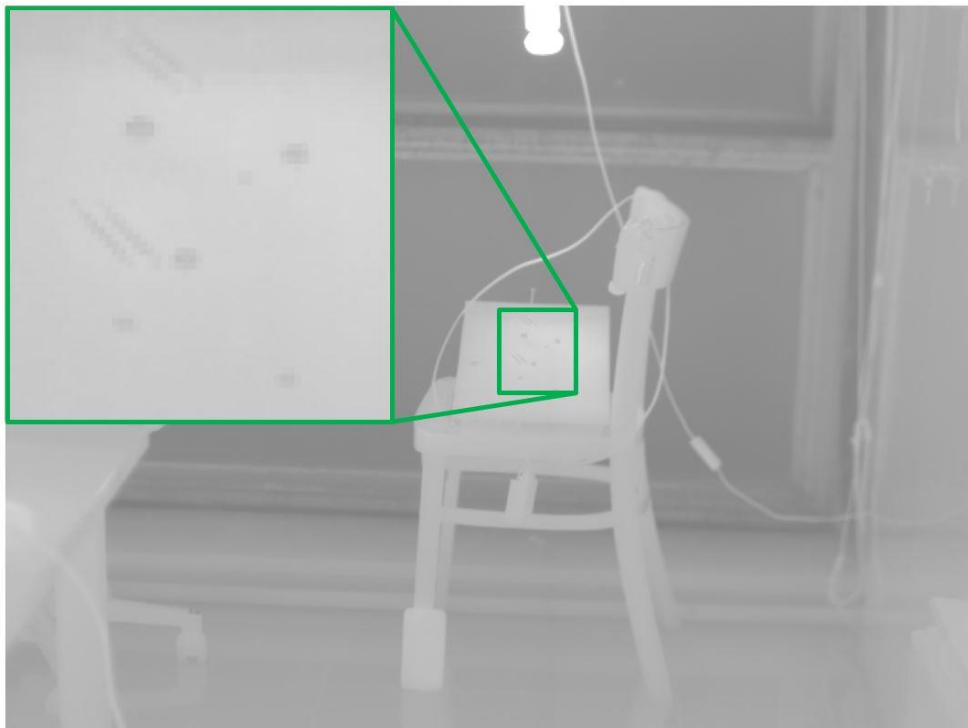


Fig. 4.9 – *Piccoli oggetti in osservazione LR*

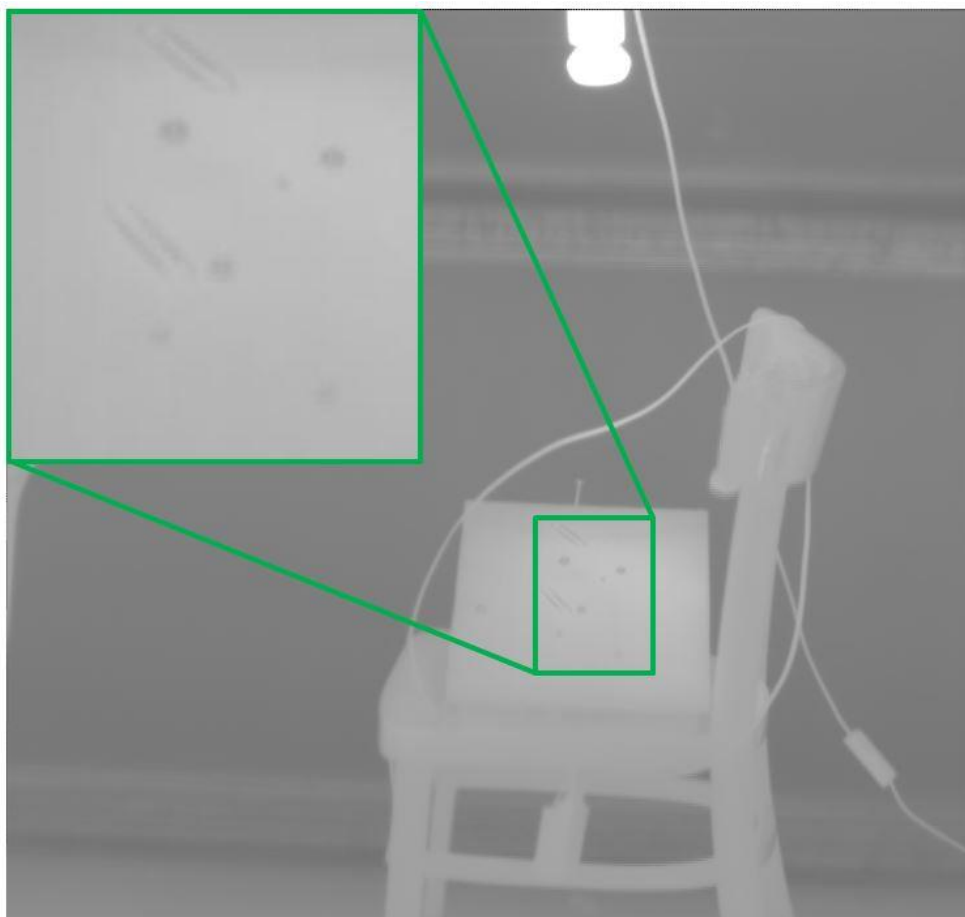


Fig. 4.10 – *Piccoli oggetti ricostruiti*

5. CONCLUSIONI

Il problema dell'aumento di risoluzione di immagini di scarsa qualità tramite la super risoluzione è stato affrontato con successo nel corso dell'esperienza di tesi. Infatti, per quanto la soluzione che minimizza il problema della risoluzione spaziale delle immagini termiche sia rappresentata dall'avanzamento tecnologico dei sensori e delle macchine termiche, queste mantengono un costo estremamente elevato, proibitivo per i più, fattore che ne rende impossibile la frequente sostituzione. Per tale motivo l'implementazione di un algoritmo di super risoluzione rappresenta probabilmente il miglior compromesso tra qualità dell'immagine e costo della strumentazione.

L'esperienza di tesi ha portato alla creazione di un algoritmo di super risoluzione *multiple-image reconstruction-based* in ambiente MATLAB, scritto in parte *ad hoc* ed in parte sfruttando *routine* già esistenti. La sperimentazione è stata svolta preliminarmente con test su immagini sintetiche, arrivando poi ad applicazioni sia su immagini digitali da fotocamera, che su immagini termiche acquisite specificamente per lo scopo.

L'applicazione dello *script* redatto ha permesso di ottenere un netto miglioramento nella risoluzione delle scene raffigurate nei *dataset* di immagini di prova in input, portando alla luce nell'immagine ricostruita particolari inizialmente molto sfocati o irriconoscibili e migliorando, anche grazie all'introduzione di bordi più smussati, la qualità dell'intera scena.

L'algoritmo, che permette la generazione di un'immagine ad alta risoluzione a partire da una serie di immagini di bassa qualità leggermente spostate l'una rispetto all'altra, è redatto secondo la logica registrazione-proiezione-interpolazione ed è composto da una serie di *sub-routine*: aumento delle dimensioni delle matrici, esplosione delle matrici, registrazione automatica *intensity-based*, proiezione delle matrici e interpolazione tramite *inverse distance weighting* sono i passi che portano a generare una singola immagine contenente tutte le informazioni dell'intero *dataset* delle immagini a bassa risoluzione.

Prima dell'applicazione su immagini reali, in un'attenta fase di simulazione, sono stati confrontati scenari diversi e algoritmi alternativi, al fine di determinare la procedura e le condizioni ottimali. In particolare, la miglior combinazione di *script* fra tutti quelli redatti, il numero di immagini necessario e sufficiente a una buona ricostruzione, i settaggi per i filtri di pre-registrazione e il posizionamento dei pixel nelle immagini esplose sono gli aspetti indagati nel corso della sperimentazione.

Dalla fase appena descritta sono emerse le particolarità del processo in esame. Parlando di osservazioni a bassa risoluzione non bisogna erroneamente immaginare che queste possano essere ottenute completamente a caso, senza un criterio; infatti il processo di acquisizione del campione di partenza è estremamente delicato e deve essere svolto tenendo ben presente alcune accortezze, come la necessità di introdurre nella scena solo spostamenti rigidi, limitare, per quanto possibile, le rotazioni ed effettuare traslazioni orizzontali/verticali di piccola entità (nell'esperienza di tesi il miglior set di immagini è stato ottenuto montando la camera su una slitta montata a sua volta su un cavalletto).

Un limite dell'algoritmo redatto è rappresentato dai tempi computazionali che l'intero processo richiede. La ricostruzione di un'immagine in scala di grigi di circa 2560×1920 pixel con valori a 8 bit richiede infatti all'incirca un'ora di tempo (nel caso di un'immagine a colori il tempo richiesto aumenta a causa della sua struttura a tre bande), mentre se l'obiettivo è ricostruire un'immagine termica, in cui il *digital number* è un valore reale con almeno due cifre decimali significative, i tempi si aggirano sulle tre ore.

Un punto cruciale, a cui è stata dedicata grande attenzione, è la co-registrazione delle osservazioni a bassa risoluzione (LR), probabilmente la fase più instabile e "rischiosa" dell'intero procedimento, per la quale sono state vagliate molteplici possibilità. Per garantire la stabilità dell'algoritmo e controllarne l'efficacia, è stata appositamente sviluppata una *routine* di verifica del processo di registrazione. Essa permette, tramite il calcolo degli scarti normalizzati rispetto alla media dei *digital number* e la verifica di appartenenza di questi a una particolare distribuzione statistica, di eliminare eventuali osservazioni registrate erroneamente, anche per piccoli valori di traslazioni o rotazioni, o particolarmente lontane dalla media delle altre immagini. È stato osservato come, in presenza di un campione di osservazioni LR in input di dimensioni maggiori rispetto a quelle minime sufficienti, l'eliminazione di osservazioni particolarmente differenti o addirittura errate rispetto alle altre porti beneficio all'intero procedimento di ricostruzione, con sensibile miglioramento della qualità.

Quanto sviluppato risulta efficace e sufficiente per una ricostruzione negli ambiti descritti, per quanto vi possano essere margini di miglioramento sotto diversi punti di vista, in particolare il tempo computazionale e l'efficacia del processo di registrazione, la cui stabilità influisce su tutte le fasi successive. Dalla sperimentazione effettuata emerge infatti che la fase con la

maggior influenza sull'intero processo è proprio quella di registrazione, nonostante anche l'interpolazione, a parità degli altri fattori, risulti determinante per la qualità del risultato.

Dopo aver verificato stabilità ed efficacia dell'algoritmo, questo è stato utilizzato con successo anche su set di immagini termiche ottenute *ad hoc*, vero obiettivo dell'esperienza di tesi, ricostruendo in tal modo matrici di temperature a risoluzione maggiore.

In conclusione, grazie alle procedure sviluppate ed ai *dataset* di immagini termiche acquisiti *ad hoc*, è stato possibile incrementare di quattro volte la risoluzione di alcune scene ottenute inizialmente con una risoluzione inferiore. Il procedimento si è rivelato efficace e stabile, o comunque controllabile grazie a specifiche procedure, e può costituire una valida alternativa all'acquisto di costosi sensori termici a maggiore risoluzione.

6. FONTI BIBLIOGRAFICHE E SITOGRAFIA

- **Baker, S., Kanade, T.:** *Limits on Super-Resolution and How to Break Them.* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24. No. 9, pp 1167-1183 (2002).
- **Diaz, G.:** *Appunti di Statistica* (2001).
- **Eberly, D.:** *Least Squares Fitting of Data* (2016).
- **Gandolfi, S.:** *Analisi Statistica delle incertezze casuali.* In: *Appunti del corso di Topografia* (2012).
- **Glasner, D., Bagon, S., Irani, M.:** *Super-Resolution from a Single Image*, Rehovot.
- **Hardie, R.C., Barnard, K, Bogner, J.G., Armstrong, E.E., Watson, E.A.:** *High-Resolution Image Reconstruction from a Sequence of Rotated and Translated Frames and its Application to an Infrared Imaging System.* In: *Electrical and Computer Engineering Faculty Publications*, Paper 8, Dayton (1998).
- **Hardie, R.:** *A fast Image Super-Resolution Algorithm Using an Adaptive Wiener Filter.* In: *IEEE Transactions on Image Processing*, Vol. 16, No. 12, pp 2953-2964 (2007).
- **Li, J.,Heap, A.D.:** *A Review of Spatial Interpolation Methods for Environmental Scientists*, Australia (2008).
- **Nasrollahi, K., Moeslund, T.B.:** *Super-resolution: a comprehensive survey.* In: *Machine Vision and Application*, pp 1423-1468, Aalborg, (2014).
- **Romero, L., Calderòn, F.:** *Scene Reconstruction, Pose Estimation and Tracking.* In: *A tutorial on Parametric Image Registration*, Vienna (2007).
- **Scarmata, G.:** *An Application of the Least Squares Plane Fitting Interpolation Process to Image Reconstruction and Enhancement*, Australia (2016).
- **Sicairos, G., Thurman, S.T., Fienup, J.R.:** *Efficient subpixel image registration algorithms.* In: *Optics Letters*, Vol. 33, No. 2, pp 156-158 (2008).
- **Verdoliva, L.:** *Appunti di Elaborazione di Segnali Multimediali* (2013).

- **<https://it.mathworks.com/help/matlab>:** Documentazione ufficiale sull'uso di Matlab e forum users.
- **<http://www.getreuer.info/tutorials/matlabimaging>.** **Getreuer, P.:** *Image Processing with MATLAB.*

APPENDICE A

Di seguito sono riportati gli algoritmi redatti durante la tesi per effettuare il procedimento di super risoluzione sulle immagini in esame.

- *Aumento dimensioni delle matrici:*

```
function aumento_dimensioni(num_img, aug_fac)
mkdir('augm_LR_images');
savdir = 'augm_LR_images';
for k=1:num_img
nomeA=['augm_LR_' num2str(k) '.txt'];
LR=['gLR_' num2str(k) 'TIFF'];
base=imread(LR);
dim_LR=size(base);
rig_LR=dim_LR(1);
col_LR=dim_LR(2);
augm=uint8(zeros(aug_fac*rig_LR, aug_fac*col_LR));
dim_augm=size(augm);
rig_augm=dim_augm(1);
col_augm=dim_augm(2);
for i=1:rig_augm
    for j=1:col_augm
        augm(i,j)=base(ceil(i/aug_fac), ceil(j/aug_fac));
    end
end
dlmwrite(fullfile(savdir, nomeA), augm(:, :), 'delimiter', '\t');
clear base augm
end
```

- *Esplosione delle matrici:*

```
function dotted_images(num_img, aug_fac)
mkdir 'dotted_LR_images';
savdir=('dotted_LR_images');
for i=1:num_img
nome=['gLR_' num2str(i) 'TIFF'];
LR_img=imread(nome);
dim_LR=size(LR_img);
dotted=zeros(dim_LR(1)*aug_fac, dim_LR(2)*aug_fac);
cont=0;
for k=1:size(dotted,1)
    j=1;
    while j<=size(dotted,2)
        if floor(k/aug_fac)==cont
            dotted(k,j)=LR_img(ceil(k/aug_fac), ceil(j/aug_fac));
            j=j+aug_fac;
        else
            j=j+aug_fac;
        end
    end
    cont=cont+(1/aug_fac);
end
buchi=['dot_gLR_' num2str(i) '.txt'];
dlmwrite(fullfile(savdir, buchi), dotted(:, :), 'delimiter', '\t');
end
```

- *Registrazione con algoritmo di cross-correlazione normalizzata*

```

function sub_pixel_shifts_xcorr(num_img,std_dev,ang_max)
oldFolder=cd('augm_LR_images');
riferimento1='augm_LR_1.txt';
delimiterIn='\t';
A1=(importdata(riferimento1,delimiterIn));
dim_rif=size(A1);
Aorig=(zeros(dim_rif(1),dim_rif(2)));
Aorig(:,:)=A1;
Aorig=uint8(Aorig);
orig=imgaussfilt(Aorig,std_dev);
rect_orig=[500 300 1399 1399];
sub_orig=imcrop(orig,rect_orig);
fileIDA=fopen('Parametri di trasformazione.txt','w');
fprintf(fileIDA,'%14s %8s %8s %8s \n','Immagine','Xtran','Ytran','Theta');
fprintf(fileIDA,'%14s %8s %8s %8s \n','augm_LR_1.txt','0.000','0.000','0.000');

for i=2:num_img
target1=['augm_LR_' num2str(i) '.txt'];
B1=(importdata(target1,delimiterIn));
dim_tar=size(B1);
Atran=(zeros(dim_tar(1),dim_tar(2)));
Atran(:,:)=B1;
Atran=uint8(Atran);
tran=imgaussfilt(Atran,std_dev);
fileID=fopen('Parametri singola immagine.txt','w');
fprintf(fileID,'%8s %8s %8s %12s \n','Xtran','Ytran','Theta','Picco_corr');
ang=-ang_max;
while abs(ang)<=ang_max
    Asub_tran=imrotate(tran,360+ang);
    rect_subtran=[1100 700 699 699];
    sub_tran=imcrop(Asub_tran,rect_subtran);
    c=normxcorr2(sub_tran(:,:),sub_orig(:,:));
    [max_c,imax]=max(abs(c(:)));
    [ypeak,xpeak]=ind2sub(size(c),imax(1));
    corr_offset=[(xpeak-size(sub_tran,2)) (ypeak-size(sub_tran,1))];
    rect_offset=[(rect_orig(1)-rect_subtran(1)) (rect_orig(2)-
rect_subtran(2))];
    offset=corr_offset+rect_offset;
    xoffset=offset(1);
    yoffset=offset(2);
    if max_c>0.98
        ang=ang+0.005;
    elseif max_c>0.93
        ang=ang+0.01;
    else
        ang=ang+0.1;
    end
    fprintf(fileID,'%8.3f %8.3f %8.3f %16.14f \n', xoffset, yoffset,
ang,max_c);

    clear sub_tran c max_c imax ypeak xpeak corr_offset rect_offset
    clear offset xoffset yoffset
end
fclose(fileID);
filename='Parametri singola immagine.txt';
delimiterIn2=' ';
A=importdata(filename,delimiterIn2);
dim_dati=size(A.data);
B=zeros(dim_dati(1),4);
B(:,1)=A.data(:,1);
B(:,2)=A.data(:,2);
B(:,3)=A.data(:,3);
B(:,4)=A.data(:,4);
C=sortrows(B,4);
dimC=size(C);

```

```

fprintf(fileIDA,'%14s %8.3f %8.3f %8.3f
\n',target1,C(dimC(1),1),C(dimC(1),2),C(dimC(1),3));
clear ang A B C dimC B1 B2 B3 dim_tar target1 target2 target3
end
fclose(fileIDA);
cd (oldFolder);

```

- *Registrazione automatica con algoritmo MATLAB:*

```

function sub_pixel_shifts_imregister_dot(base,cont,std_dev)
dir=[num2str(base) '-' num2str(cont) '_imreg_dot_LR_images_' num2str(std_dev)];
mkdir(dir);
savdir=dir;
oldFolder=cd('augm_LR_images');
delimiterIn='\t';
name_rif=['augm_LR_' num2str(base) '.txt'];
rif=importdata(name_rif);
rect_rif=[800 800 1200 800];
Asub_rif=imcrop(rif,rect_rif);
gauss_rif=imgaussfilt(Asub_rif,std_dev);
cd(oldFolder);

for i=base:cont
oldFolder=cd('augm_LR_images');
nome=['augm_LR_' num2str(i) '.txt'];
target=(importdata(nome,delimiterIn));
gauss_tar=imgaussfilt(target,std_dev);
[optimizer,metric]=imregconfig('multimodal');
optimizer.InitialRadius=0.0009;
optimizer.Epsilon=1.5e-6;
optimizer.GrowthFactor=1.01;
optimizer.MaximumIterations=300;
tform=imregtform(gauss_tar,gauss_rif,'rigid',optimizer,metric);

Rfixed=imref2d(size(Asub_rif));
nome_dot=['dot_gLR_' num2str(i) '.txt'];
cd(oldFolder);
oldFolder=cd('dotted_LR_images');
dot_target=(importdata(nome_dot));
reg_tar=imwarp(dot_target,tform,'nearest','OutputView',Rfixed);
reg_tar2=imwarp(target,tform,'nearest','OutputView',Rfixed);
figure(i); imshowpair(Asub_rif,reg_tar2,'falsecolor');
cd(oldFolder);
registered=['imregister_dot_LR_' num2str(i) '.txt'];
registered2=['imregister_augm_LR_' num2str(i) '.txt'];
dlmwrite(fullfile(savdir,registered),reg_tar(:,:),'delimiter','\t');
dlmwrite(fullfile(savdir,registered2),reg_tar2(:,:),'delimiter','\t');
end

```

- *Proiezione delle immagini registrate*

```

function proiezione_dotted(base,cont,std_dev)
dir=[num2str(base) '-' num2str(cont) '_imreg_dot_LR_images_' num2str(std_dev)];;
oldFolder=cd(dir);
test_name=['imregister_dot_LR_' num2str(base) '.txt'];
test_dim=importdata(test_name);
dim_HR=size(test_dim);
HR_image=(zeros(dim_HR(1),dim_HR(2)));
somma=(zeros(dim_HR(1),dim_HR(2)));
A=zeros(dim_HR(1),dim_HR(2));
img_reg_ok=importdata('Immagini registrate correttamente.txt');
dim_reg_ok=size(img_reg_ok);

```

```

for i=1:dim_reg_ok(1)
    reg_target=['imregister_dot_LR_' num2str(img_reg_ok(i)) '.txt'];
    delimiterIn='\t';
    A1=(importdata(reg_target,delimiterIn));
    ref_target(:,:)=A1;
    somma(:,:)=somma(:,:)+ref_target(:,:);
    for j=1:dim_HR(1)
        for k=1:dim_HR(2)
            non=nnz(ref_target(j,k));
            A(j,k)=A(j,k)+non;
            clear non
        end
    end
    clear A1 ref_target
end
HR_image(:,:)=rdivide(somma(:,:),A);
for i=1:dim_HR(1)
    for j=1:dim_HR(2)
        if isnan(HR_image(i,j))==1
            HR_image(i,j)=0;
        else
            end
        end
    end
end
cd(oldFolder);
nome=[num2str(base) '-' num2str(cont) '_projection_SR_' num2str(std_dev) '.txt'];
nome1=[num2str(base) '-' num2str(cont) '_projection_SR_' num2str(std_dev) '.TIFF'];
imwrite(uint8(HR_image),nome1);
dlmwrite(nome,HR_image,'delimiter','\t');

```

- *Verifica di registrazione:*

```

function verifica_registrazione(base,cont,std_dev)
dir=[num2str(base) '-' num2str(cont) '_imreg_dot_LR_images_' num2str(std_dev)];
oldFolder=cd(dir);
rif_name=['imregister_augm_LR_1.txt'];
rif=importdata(rif_name);
dim=size(rif);
somma=zeros(dim(1),dim(2));
delimiterIn='\t';
fileID=fopen('Somma scarti normalizzati.txt','w');
lista_errate=importdata('Registrata erroneamente.txt');
for i=base:cont
    if i==lista_errate(:,:)
    elseif i~=lista_errate(:,:)
        tar_name=['imregister_augm_LR_' num2str(i) '.txt'];
        targ=importdata(tar_name);
        somma=somma+targ;
    end
end
medie=somma/(cont-length(lista_errate)+1);
for i=base:cont
    if i==lista_errate(:,:)
    elseif i~=lista_errate(:,:)
        tar_name=['imregister_augm_LR_' num2str(i) '.txt'];
        targ=importdata(tar_name);
        scarto=abs(targ-medie);
        dev=sqrt((scarto.^2)/(cont-length(lista_errate)));
        norm_scarto=rdivide(scarto,dev);
        for j=1:dim(1)
            for k=1:dim(2)
                if isnan(norm_scarto(j,k))==1
                    norm_scarto(j,k)=0;
                else
                    end
            end
        end
    end
end

```



```

        end
    end
    somma_scarti_norm=sum(norm_scarto(:));
    fprintf(fileID,'%14f %14.3f \n',i,somma_scarti_norm);
end
end
fclose(fileID);
to_sort=importdata('Somma scarti normalizzati.txt');
media_norm=mean(to_sort(:,2));
dev_norm=std(to_sort(:,2));
scarto_scarto_norm=abs(to_sort(:,2)-media_norm);
scarto_norm_scarto_norm=scarto_scarto_norm/dev_norm;
to_sort(:,2)=scarto_norm_scarto_norm;
fileID=fopen('Scarti norm scarti norm.txt','w');
fprintf(fileID,'%14s %14s \n','Immagine','Scarti normalizzati delle somme');
sorted=sortrows(to_sort,2,'descend');
for i=1:(cont-length(lista_errate)+1)
    fprintf(fileID,'%14.0f %14.3f \n',sorted(i,1),sorted(i,2));
end
fclose(fileID);
fileID=fopen('Immagini registrate correttamente.txt','w');
scarti=importdata('Scarti norm_scarti norm.txt');
dim_scarti=size(scarti.data);
err=0;
for i=1:dim_scarti(1)
    if scarti.data(i,2)<=2.3
        fprintf(fileID,'%5f \n',scarti.data(i,1));
    elseif scarti.data(i,2)>2.3
        err=scarti.data(i,1);
        break
    end
end
end
fileIDA=fopen('Registrata erroneamente.txt','a');
fprintf(fileIDA,'%1f \n',err);
fclose(fileIDA);
fclose(fileID);
cd(oldFolder);

```

- *Interpolazione tramite Inverse Distance Weighting*

```

function interpolazione_IDW(base,cont,std_dev)
projection=[num2str(base) '-' num2str(cont) '_projection_SR_' num2str(std_dev)
'.txt'];
delimiterIn='\t';
HR_image=(importdata(projection,delimiterIn));
dim_HR=size(HR_image);
for i=3:dim_HR(1)-2
    for j=3:dim_HR(2)-2
        if HR_image(i,j)==0
            x_inf=j-2;
            x_sup=j+2;
            y_inf=i-2;
            y_sup=i+2;
            a=[x_inf,x_sup];
            c=[y_inf,y_sup];
            non=nnz(HR_image(y_inf:y_sup,x_inf:x_sup));
            dist=zeros(1,non);
            inv_dist=zeros(1,non);
            pixel=zeros(1,non);
            pesi=zeros(1,non);
            contA=1;
            for ii=y_inf:y_sup
                for jj=x_inf:x_sup
                    if HR_image(ii,jj)>0
                        dist(1,contA)=sqrt((mean(a)-jj)^2+(mean(c)-ii)^2);
                    end
                end
            end
            inv_dist(1,contA)=1/dist(1,contA);
            pixel(1,contA)=HR_image(i,j);
            pesi(1,contA)=inv_dist(1,contA)^2;
            contA=contA+1;
        end
    end
end

```

```

        inv_dist(1,contA)=1/dist(1,contA);
        pixel(1,contA)=HR_image(ii,jj);
        contA=contA+1;
    else
    end
end
end
somma_pesi=sum(inv_dist);
pesi(1,:)=inv_dist(1,:)/somma_pesi;
HR_image(i,j)=sum(times(pixel,pesi));
else
end
end
end
nome=[num2str(base) '-' num2str(cont) '-SR-imregister-IDW-' num2str(std_dev)
'.txt'];
nome1=[num2str(base) '-' num2str(cont) '_SR_imregister_IDW_' num2str(std_dev)
'TIFF'];
dlmwrite(nome,HR_image(:,:),'delimiter','\t');
imwrite(uint8(HR_image),nome1);

```