



**SAPIENZA**  
UNIVERSITÀ DI ROMA

MASTER II LIVELLO IN “CALCOLO SCIENTIFICO”

A.A. 2011-2012

# **Applicazione di Tecniche di Super-risoluzione a Immagini Astronomiche**

RAPPORTO FINALE DI STAGE  
DI

**Marco Castellano**

**Tutor**

Prof. Maurizio Falcone

**Tutor esterno (INAF)**

Dr. Adriano Fontana



# Indice

<b>Indice</b>	<b>ii</b>
<b>Introduzione</b>	<b>1</b>
<b>1 Introduzione alle Tecniche di Super-risoluzione</b>	<b>3</b>
1.1 Modello di formazione dell'immagine . . . . .	3
1.1.1 Super-risoluzione: metodi iterativi . . . . .	6
1.1.2 Approccio Bayesiano . . . . .	7
1.1.3 Super-risoluzione in frequenza . . . . .	8
1.1.4 Tecniche example-based e super-risoluzione da singole immagini . . . . .	8
<b>2 Immagini astronomiche</b>	<b>11</b>
2.1 Caratteristiche generali . . . . .	11
2.1.1 Acquisizione e calibrazione delle immagini . . . . .	13
2.2 Analisi di immagini astronomiche . . . . .	14
2.2.1 Osservazioni multi-banda . . . . .	15
2.3 Visualizzazione e formati delle immagini . . . . .	16
2.4 Potenzialità delle tecniche di super-risoluzione applicate a immagini astronomiche . . . . .	16
<b>3 Algoritmi variazionali di Super-Risoluzione</b>	<b>19</b>
3.1 Algoritmi di minimizzazione con regolarizzazione L2 . . . . .	19
3.2 Algoritmo con termine di regolarizzazione L1 . . . . .	21
3.3 Schemi numerici . . . . .	22
3.4 Implementazione Software . . . . .	24
3.4.1 Struttura e funzionamento di SuperResolve . . . . .	25
3.4.2 Analisi statistica . . . . .	31
3.5 Note sull'implementazione . . . . .	31
<b>4 Esperimenti numerici</b>	<b>35</b>
4.1 Immagini fotografiche . . . . .	36
4.1.1 Variazione del numero dei frame e casi estremi . . . . .	42
4.2 Immagini astronomiche . . . . .	45

---

4.2.1	Simulazioni da immagini <i>Hubble Ultra-Deep Field</i> . . . . .	46
4.2.2	Super-risoluzione di osservazioni astronomiche della missione EUCLID . . . . .	49
4.3	Prestazioni software . . . . .	55
<b>5</b>	<b>Conclusioni e Sviluppi Futuri</b>	<b>59</b>
5.1	Sviluppi futuri . . . . .	61
	<b>Bibliografia</b>	<b>63</b>

# Introduzione

L'ultimo decennio ha visto una evoluzione senza precedenti nelle tecnologie di acquisizione digitale di fotografie e video che ha avuto ricadute nei settori più disparati, dall'industria televisiva e cinematografica alla meteorologia e all'imaging medico. Tale progresso può essere misurato tramite la capacità di ottenere immagini di sempre maggiore "risoluzione", laddove questo termine non si intende esclusivamente riferito alla risoluzione spaziale vera e propria (ad esempio misurata dal numero di pixel disponibili) ma anche alla capacità di ridurre o compensare sorgenti di rumore e distorsioni ottiche negli strumenti di acquisizione delle immagini. In breve, la capacità di ottenere rappresentazioni digitali sempre più fedeli della realtà.

Di pari passo, tanto dal settore commerciale che da quello scientifico è venuta una sempre maggiore spinta verso la creazione di algoritmi e software in grado di migliorare, o quanto meno sfruttare nel miglior modo possibile, le immagini disponibili. In questo contesto nascono le tecniche di ricostruzione delle immagini definite di "super-risoluzione" (SR) che si pongono lo scopo di ottenere una immagine di maggiore definizione a partire dalle informazioni contenute in più frame di risoluzione inferiore.

La presente Tesi illustra il lavoro di stage svolto nel periodo Dicembre 2012 - Marzo 2013 presso l'Osservatorio Astronomico di Roma (INAF). Scopo del lavoro è stato valutare le potenzialità dell'introduzione di tecniche SR nel campo dell'astrofisica osservativa. Probabilmente tanto per casualità storiche che per ragioni pratiche, le tecniche di ricostruzione e aumento della definizione di immagini digitali hanno ricevuto limitata attenzione in astrofisica: per quanto ci è dato sapere, quello illustrato nella presente Tesi è il primo tentativo di applicazione delle tecniche SR a immagini astronomiche.

Nel primo capitolo verrà presentata una descrizione delle diverse tecniche di super-risoluzione proposte in letteratura e il modello di formazione dell'immagine che ne è alla base. Il capitolo 2 è dedicato a presentare le particolarità delle immagini astrofisiche prestando attenzione ai possibili vantaggi dell'uso di tecniche di super-risoluzione e alle problematiche legate all'esigenza di una accurata calibrazione delle immagini. Nel capitolo 3 vengono descritti gli approcci variazionali sperimentati durante il lavoro di stage discutendo gli schemi numerici e l'implementazione del relativo codice software in linguaggio FORTRAN 90 con parallelizzazione OpenMP. I risultati della sperimentazione su immagini sono dis-

cussi nel capitolo 4. Infine nel capitolo 5 verrà presentato un sommario del lavoro svolto e dei risultati ottenuti, assieme a una valutazione sul possibile uso della SR in astrofisica e i possibili sviluppi futuri di questa linea di ricerca.

# Capitolo 1

## Introduzione alle Tecniche di Super-risoluzione

Lo scopo delle tecniche di Super-Risoluzione (SR) è ricostruire un'immagine ad alta risoluzione a partire da un insieme di immagini a risoluzione inferiore. Definendo  $r$  il fattore di guadagno in risoluzione, cioè il rapporto tra le dimensioni lineari del pixel a bassa e alta risoluzione, il problema della SR può essere formulato nel seguente modo:

*Dato una stessa scena osservata tramite un insieme di immagini  $\{x_L^{(n)}\}_{n=1,p}$  ("immagini LR"), ciascuna di  $M \times N$  pixel, e dato  $r > 1$ , ricostruire un'immagine  $x_H$  ("immagine HR") di dimensioni  $r \cdot M \times r \cdot N$  pixel.*

Negli ultimi trent'anni sono state proposte in letteratura una varietà di metodi di super-risoluzione: scopo del presente capitolo è fornire una breve introduzione alle principali tecniche SR attualmente disponibili e alle loro basi matematiche. Si vedano ad es. Chaudhuri [7] e Milanfar [22] per una trattazione approfondita di gran parte delle tecniche qui menzionate e le review di Park et al. [25] e Yang and Huang [38] per una discussione dei recenti progressi e linee di ricerca.

### 1.1 Modello di formazione dell'immagine

La soluzione al problema della SR può essere intuitivamente compresa ragionando nello spazio dei pixel: come schematizzato in Figura 1.1, più immagini a bassa risoluzione che siano tra loro shiftate di multipli seminteri del pixel stesso campionano la scena in modo differente e dunque contengono informazione aggiuntiva utilizzabile per ricostruire un'immagine a risoluzione maggiore. Per giungere a delle tecniche matematiche di SR occorre però innanzitutto introdurre un modello di acquisizione delle immagini che descriva il passaggio dalla scena reale a una immagine elettronica più o meno alterata da fattori tecnici e discretizzata su una griglia di pixel.

Consideriamo innanzitutto un modello continuo in cui l'intensità dell'immagine bidimensionale in funzione della posizione sia data da  $Y(x,y)$ . Gli effetti del-

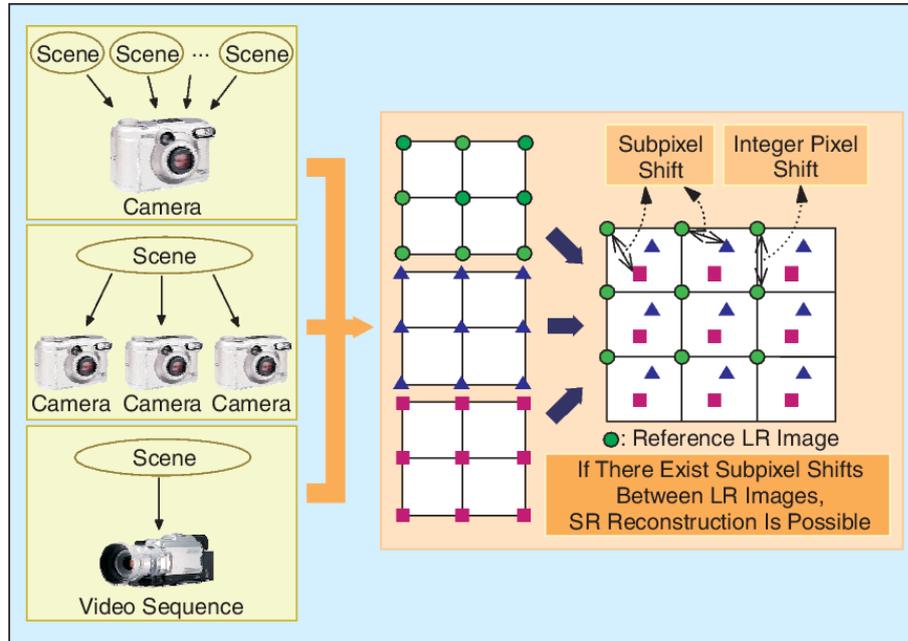


Figura 1.1: Una rappresentazione schematica della relazione tra più frame a bassa risoluzione shiftati tra loro e l'immagine ad alta risoluzione che si vuole ricostruire [tratto da 25].

l'atmosfera e delle ottiche fanno sì che l'immagine giunga alterata al sensore: tale processo può essere globalmente espresso tramite una convoluzione di  $Y(x,y)$  con un kernel  $H(x,y)$  denominato Point Spread Function (PSF):

$$\tilde{Y}(x,y) = Y(x,y) * H(x,y) \quad (1.1)$$

L'immagine registrata dalla fotocamera nei pixel con indici  $(i, j)$  è il risultato del campionamento di  $\tilde{Y}(x,y)$  a intervalli  $\delta x = T_1$  e  $\delta y = T_2$  lungo i due assi, a cui si aggiunge l'effetto del rumore, usualmente modellizzato come una componente additiva Gaussiana, indipendente dal segnale, a media nulla e varianza costante:

$$x(i, j) = \tilde{Y}(iT_1, jT_2) + \epsilon_n(i, j) \quad (1.2)$$

Il modello continuo sopra tratteggiato fornisce un'utile schematizzazione dei processi fisici in atto durante l'acquisizione di immagini digitali. Per poter sviluppare algoritmi che risolvano il problema della Super-risoluzione abbiamo però bisogno di un modello discreto che descriva il passaggio dall'immagine ad alta risoluzione  $x_H$  a una serie di immagini  $x_L$  a risoluzione inferiore. In analogia con il caso continuo, le  $n$  immagini  $x_L$  possono essere considerate come il risultato finale dell'applicazione all'immagine  $x_H$  di operatori di roto-traslazione (o "warping"), convoluzione (effetto della PSF) e campionamento (o "decimazione"). Dal momento

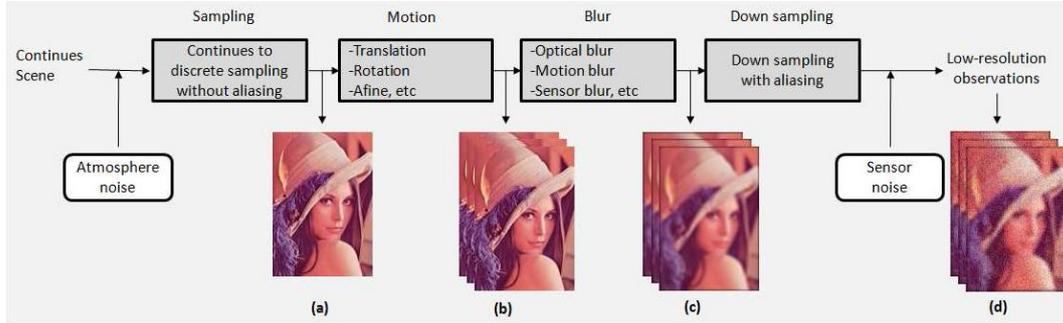


Figura 1.2: Modello di formazione dell'immagine [tratto da 25].

che tali operazioni sono lineari, il modello discreto può essere espresso sinteticamente in notazione matriciale considerando le immagini come vettori colonna di ordine arbitrario [e.g. 9]:

$$\underline{x}_L^{(n)} = DH_n W_n \underline{x}_H + \underline{\epsilon}_n \quad (1.3)$$

Dove:

- $\underline{x}_L^{(n)}$  è la n-esima immagine di dimensioni  $M \times N$  pixel riordinata come vettore.
- $\underline{x}_H$  è l'immagine ad alta risoluzione, di dimensioni  $rM \times rN$  pixel riordinata come vettore.
- $\underline{\epsilon}_n$  è la componente additiva di rumore Gaussiano della n-esima immagine riordinata come vettore.
- $W_n$  è la matrice reale con dimensioni  $r^2MN \times r^2MN$  di warping della n-esima immagine.
- $H_n$  è la matrice reale con dimensioni  $r^2MN \times r^2MN$  di convoluzione della n-esima immagine.
- $D$  è la matrice reale con dimensioni  $MN \times MN$  di decimazione.

Le equazioni relative alle  $p$  immagini possono infine essere raccolte in un'unica espressione:

$$\begin{bmatrix} \underline{x}_L^{(1)} \\ \vdots \\ \underline{x}_L^{(p)} \end{bmatrix} = \begin{bmatrix} DH_1 W_1 \\ \vdots \\ DH_p W_p \end{bmatrix} \underline{x}_H + \begin{bmatrix} \underline{\epsilon}_1 \\ \vdots \\ \underline{\epsilon}_p \end{bmatrix} \iff \underline{x}_L = K \underline{x}_H + \underline{\epsilon} \quad (1.4)$$

In cui  $\underline{x}_L$  e  $\underline{\epsilon}$  sono i vettori di dimensioni  $pMN \times 1$  con, rispettivamente, i valori dei pixel delle  $p$  immagini a bassa risoluzione e delle relative componenti

di rumore, mentre la matrice reale  $K$  racchiude le informazioni sul modello di formazione delle  $p$  immagini LR. La matrice  $K$  è costituita infatti da  $p$  blocchi ottenuti dai prodotti matriciali  $DH_nW_n$  ( $n=1,p$ ) e ha dunque dimensioni  $pMN \times r^2MN$ . Le proprietà del sistema dipendono dal numero  $p$  di immagini (non ridondanti) disponibili, e ovviamente se  $p < r^2$  il sistema sarà sotto-determinato.

### 1.1.1 Super-risoluzione: metodi iterativi

Il problema così formulato si presenta del tutto analogo ai classici problemi di ricostruzione di singole immagini [e.g. denoising, deblurring, cfr. 1]: un gran numero di algoritmi di super-risoluzione proposto in letteratura utilizza le tecniche già considerate per questa classe di problemi inversi. In particolare, assumendo l'ipotesi che  $\epsilon$  sia rumore bianco Gaussiano, possiamo trovare una soluzione approssimata di  $\underline{x}_H$  minimizzando il seguente funzionale energia:

$$E(\underline{x}_H) = \frac{1}{2} \|\underline{x}_L - K\underline{x}_H\|_2^2 \quad (1.5)$$

La soluzione di  $\min_{\underline{x}_H} (E(\underline{x}_H))$  soddisfa:

$$K^T K \underline{x}_H - K^T \underline{x}_L = 0 \quad (1.6)$$

in cui  $K^T$  è l'aggiunto di  $K$ .

Tuttavia, questo è un problema di minimizzazione mal posto e mal condizionato a causa della presenza del rumore e dal momento che tipicamente:  $K^T$  e  $K$  non sono in relazione uno-a-uno; gli autovalori di  $K$  sono piccoli portando dunque a instabilità numerica; nei casi reali la nostra conoscenza di  $K$ , cioè degli operatori  $D$ ,  $H_n$  e  $W_n$ , deriva dalle stesse immagini  $\underline{x}_L$ . Un metodo classico per risolvere problemi di minimizzazione mal posti consiste nell'aggiungere un termine di regolarizzazione:

$$E(\underline{x}_H) = \frac{1}{2} (\|\underline{x}_L - \underline{x}_H\|_2^2 + \lambda f(\underline{x}_H)) \quad (1.7)$$

dove  $f(\cdot)$  è una funzione convessa e  $\lambda \in \mathbb{R}^+$ . Nell'ottica in cui  $f(\underline{x}_H)$  costituisce un'ipotesi aggiuntiva sulla regolarità della soluzione cercata esso è dato dalla norma  $L_2$  [34] o  $L_1$  ["total variation", 32] di  $\nabla \underline{x}_H$ :

$$f(\underline{x}_H) = \|\nabla \underline{x}_H\|_2$$

$$f(\underline{x}_H) = \|\nabla \underline{x}_H\|_1$$

Come vedremo in maggiore dettaglio nel Capitolo 3, il minimo di 1.7 può essere trovato in modo iterativo. Tecniche SR variazionali di questo tipo sono state proposte, solo per citare alcuni esempi, da: Irani and Peleg [18], Hardie et al. [15], Zomet and Peleg [40], Bannore [2], Mitzel et al. [23], Unger et al. [36].

Vantaggi notevoli di questo genere di algoritmi sono dati dalla possibilità di utilizzare metodi di minimizzazione computazionalmente efficienti e dalla versatilità nell'uso di approcci di regolarizzazione diversi a seconda delle proprietà delle immagini.

Come alternativa agli approcci variazionali, altri autori [e.g. 33, 26] hanno introdotto tecniche iterative basate sul metodo di "proiezione su insiemi convessi" (POCS), anch'esso precedentemente utilizzato per la "image restoration" di singoli frame [39]. In base a tali tecniche, le ipotesi a priori sulle proprietà della soluzione corrispondono al restringere la soluzione stessa a far parte di insiemi convessi chiusi  $C_i$ . Ad esempio la consistenza con i dati ( $p$  immagini a bassa risoluzione  $x_L^i$ ) implica l'appartenenza della soluzione  $x_H$  a  $p$  insiemi convessi:

$$C_i = \{x_H \mid \|x_L^i - DH_i W_i x_H\|^2 \leq \sigma^2, 1 \leq i \leq p\} \quad (1.8)$$

Analogamente possono essere definiti insiemi che rappresentino vincoli sulla regolarità o sull'ampiezza dell'immagine. La soluzione apparterrà dunque all'intersezione degli  $N$  insiemi  $C_s = \cap_i^N C_i$  se tale insieme non è vuoto. Definito per ciascun  $C_i$  un operatore di proiezione  $P_i$ , la soluzione  $x_H$  viene cercata iterativamente a partire da un arbitrario punto di partenza  $x_H^0$ :

$$x_H^{m+1} = P_m P_{m-1} \dots P_2 P_1 x_H^m \quad (1.9)$$

Le tecniche POCS risultano da un punto di vista computazionale meno efficienti degli approcci variazionali, ma forniscono la possibilità di utilizzare svariati tipi di ipotesi sul genere di soluzione cercato.

### 1.1.2 Approccio Bayesiano

Le tecniche di super-risoluzione di tipo Bayesiano [e.g. 27] trattano il problema di ricostruire l'immagine HR da un punto di vista statistico a partire dal teorema di Bayes:

$$X_H = \arg \max_{x_H} P(x_H | x_L) = \arg \max_{x_H} \frac{P(x_L | x_H) P(x_H)}{P(x_L)} \quad (1.10)$$

In Eq. 1.10,  $P(x_L | x_H)$  costituisce il termine di *likelihood* o fedeltà della soluzione rispetto ai dati ( $x_L$ ), mentre  $P(x_H)$  è il termine di *prior*, che costituisce l'ipotesi a priori sulla regolarità della soluzione ad esempio tramite funzionali di tipo Gibbs:

$$P(x_H) = \frac{1}{Z} \exp[-U(x)] \quad (1.11)$$

dove  $Z$  è una costante di normalizzazione e la funzione energia  $U(x)$  definita in base alle derivate dell'immagine misura il "costo" causato da soluzioni non regolari. Il termine di *likelihood*  $P(x_L | x_H)$  è basato sul modello di formazione delle immagini e comprende inoltre ipotesi sulle proprietà del rumore nelle diverse immagini: ciò costituisce un interessante vantaggio dell'approccio Bayesiano rispetto alle tecniche "deterministiche" esposte nel precedente paragrafo.

### 1.1.3 Super-risoluzione in frequenza

Sebbene oggi non rientrino tra le tecniche SR più utilizzate, le basi del concetto di super-risoluzione sono state gettate ragionando nello spazio delle frequenze dal pionieristico lavoro Tsai and Huang [35] in cui gli autori sfruttano le proprietà di traslazione e aliasing delle Trasformate di Fourier per ricostruire l'immagine HR dai frame LR. Chiamiamo  $X_H(x, y)$  la funzione continua che descrive la scena in esame e  $X_n(x, y) = X_H(x + \delta_x^{(n)}, y + \delta_y^{(n)})$ , con  $n = 1, \dots, p$  l'insieme di immagini traslate di quantità arbitrarie ma note  $(\delta_x^{(n)}, \delta_y^{(n)})$ . La trasformata di Fourier continua (CFT)  $\tilde{X}$  dei  $p$  frame traslati si può scrivere:

$$\tilde{X}_n(u_1, u_2) = \exp[j2\pi(\delta_x^{(n)}u_1 + \delta_y^{(n)}u_2)]\tilde{X}_H(u_1, u_2) \quad (1.12)$$

Le immagini shiftate sono campionate a intervalli  $T_1$  e  $T_2$ :  $y_n(i, j) = X_n(iT_1 + \delta_x^{(n)}, jT_2 + \delta_y^{(n)})$ , con  $i=0, \dots, M-1$  e  $j=0, \dots, N-1$ . Chiamiamo  $\tilde{y}_n(v_1, v_2)$  le trasformate di Fourier discrete (DFT) delle  $p$  immagini  $y_n(i, j)$ . Queste sono legate alle CFT  $\tilde{X}_n$  dalla relazione di *aliasing*:

$$\tilde{y}_n(v_1, v_2) = \frac{1}{T_1 T_2} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \tilde{X}_n\left(\frac{2\pi}{T_1}\left(\frac{v_1}{M} - k_1\right), \frac{2\pi}{T_2}\left(\frac{v_2}{N} - k_2\right)\right) \quad (1.13)$$

Assumendo che  $\tilde{X}_H(u_1, u_2)$  sia *band-limited*, cioè  $|\tilde{X}_H(u_1, u_2)| = 0$  per  $|u_1| \geq M\pi/T_1$  e  $|u_2| \geq N\pi/T_2$ , combinando le equazioni 1.12 e 1.13 otteniamo una relazione tra i coefficienti DFT di  $\tilde{y}_n$  e un campionamento dei coefficienti CFT ignoti di  $\tilde{X}_H$ :

$$\underline{\tilde{y}} = \Phi \underline{\tilde{X}}_H \quad (1.14)$$

Dove,  $\underline{\tilde{y}}$  è un vettore colonna  $p \times 1$  il cui  $n$ -esimo elemento è il coefficiente DFT di  $\tilde{y}_n(v_1, v_2)$ ,  $\underline{\tilde{X}}_H$  è un vettore colonna di  $MN$  elementi che contiene i coefficienti CFT di  $X_H(x, y)$ , e  $\Phi$  è una matrice  $p \times MN$  che lega  $\underline{\tilde{y}}$  a  $\underline{\tilde{X}}_H$ . L'equazione 1.14 definisce un insieme di equazioni lineari da cui ricavare  $\underline{\tilde{X}}_H$ , la cui DFT inversa è l'immagine super-risolta cercata.

Gli approcci al problema della SR in frequenza sono caratterizzati da una buona efficienza computazionale ma sono fortemente limitati dalle assunzioni di pure traslazioni, assenza di rumore e di distorsioni.

### 1.1.4 Tecniche example-based e super-risoluzione da singole immagini

Nelle tecniche SR sopra menzionate l'insieme di immagini a bassa risoluzione impone di fatto dei vincoli lineari sui valori di intensità dei pixel dell'immagine ad alta risoluzione che si vuole ricostruire sfruttando il fatto che l'informazione ad alta frequenza cercata è presente in forma di aliasing nell'insieme di immagini LR.

Un approccio diverso al problema è stato dato dalle tecniche di “example-based Super-Resolution”, note anche come “image hallucination”. La SR example-based si basa sulla ricerca di corrispondenze tra zone (“patches”) delle immagini LR e immagini ad alta risoluzione appartenenti a un *training-set*: dato il modello di formazione dell'immagine, il frame super-risolto viene ottenuto individuando i modelli di “patches” più appropriati, ad esempio tramite tecniche a campi di Markov [e.g. 10] o maximum a-posteriori likelihood [e.g. 8]. Recentemente Glasner et al. [11] hanno mostrato che, sfruttando particolari ridondanti all'interno della stessa immagine, si può ottenere un'immagine super-risolta da un solo frame a bassa risoluzione. Le tecniche example-based permettono di raggiungere fattori di guadagno in risoluzione maggiori rispetto alla SR “classica” ma hanno prestazioni ottimali in particolare su immagini di cui è possibile costruire training-set ampi e accurati (testi scritti, visi umani), non garantendo in altri casi che l'immagine SR “allucinata” riproduca i dettagli incogniti della vera immagine ad alta risoluzione.



## Capitolo 2

# Immagini astronomiche

Lo scopo dell'*imaging* astrofisico è permettere l'indagine delle proprietà fisiche degli oggetti celesti da misure della loro posizione e, principalmente, del loro flusso luminoso. Le immagini vengono acquisite da sensori CCD posti al fuoco di sistemi ottici complessi di cui il telescopio (generalmente costruito con specchi a geometria conica) costituisce il principale elemento. Sebbene il sistema ottico di un telescopio e dei suoi strumenti sia molto più complesso di una comune macchina fotografica o videocamera, il processo di acquisizione di immagini astronomiche digitali può essere descritto tramite il modello di formazione delle immagini digitali illustrato nel Paragrafo 1.1, considerando tali immagini come risultato dell'applicazione di operatori di campionamento, warping, convoluzione a una scena reale ad alta risoluzione ignota. Tuttavia, come evidenzieremo nel presente capitolo, vi sono alcune differenze tra le caratteristiche delle immagini astronomiche e le comuni immagini fotografiche. Introdurremo inoltre alcuni aspetti sulla visualizzazione e analisi scientifica delle osservazioni astronomiche evidenziando infine la potenzialità delle tecniche di super-risoluzione nell'ambito astrofisico.

### 2.1 Caratteristiche generali

Osservazioni tramite imaging sono comunemente sfruttate per studiare una grande varietà di oggetti celesti: dal sole, ai corpi del sistema solare, dalle popolazioni stellari della nostra galassia a galassie distanti miliardi di anni luce. Date le enormi differenze tra queste sorgenti, su tutte la loro distanza e la luminosità apparente, i metodi in cui vengono analizzate le immagini e le proprietà delle immagini stesse possono variare, pur mantenendo alcune caratteristiche comuni: in particolare, le immagini astronomiche devono permettere un'analisi *fisica* in termini di *flusso* e *posizione* degli oggetti celesti.

***Segnale e sorgenti di rumore nelle immagini.*** Una prima caratteristica fondamentale, oltre che una sostanziale differenza rispetto alle comuni fotografie, è data dal fatto che ai valori (conteggi) registrati nei pixel di immagini astronomiche deve poter essere assegnato un valore in termini di fotoni ricevuti dalla corrispondente

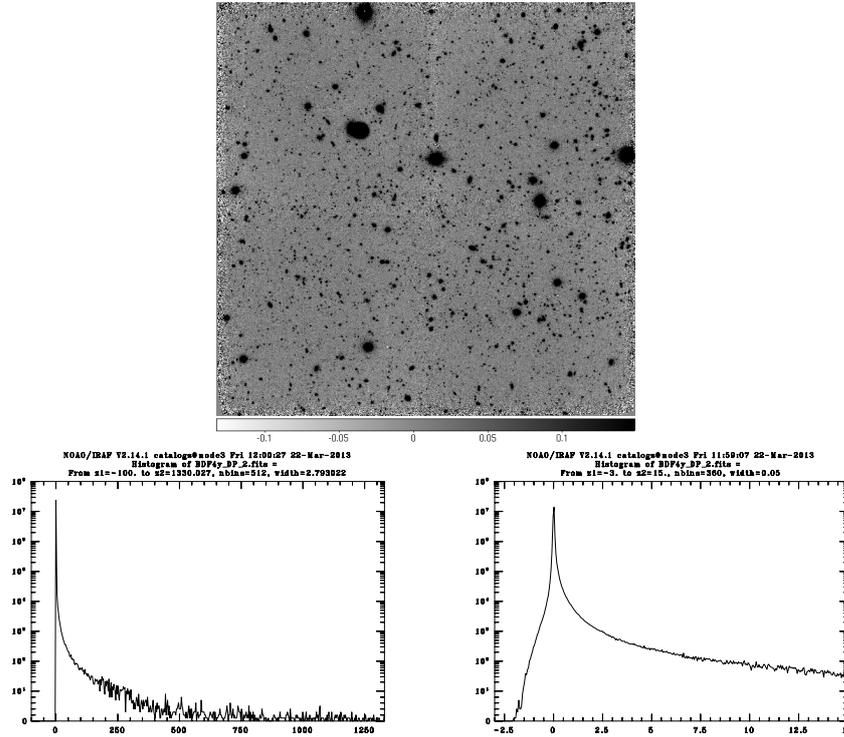


Figura 2.1: In alto: immagine a  $\lambda \sim 1 \mu\text{m}$  di una regione di cielo di  $\sim 7.5 \times 7.5$  arcmin, ottenuta con la camera infrarossa Hawk-I (ESO-VLT) per un tempo totale di esposizione di  $\sim 16$  ore [6]. In basso: istogramma dei valori dei pixel (a sinistra) e ingrandimento dell'istogramma attorno a  $\sim 0$  (destra).

regione di cielo. Per consentire tale genere di analisi fisica dei valori registrati dai pixel le immagini astronomiche digitali devono innanzitutto avere un elevato range dinamico: generalmente di 16bit (equivalente a 65536 livelli di grigio), o comunque mai inferiore a 12bit (4096 livelli).

Un sensore CCD è di fatto un contatore proporzionale: il valore dei singoli pixel è dato dal numero di elettroni di conduzione generati al suo interno assorbendo i fotoni incidenti. Assumendo che ogni fotone produca un foto-elettrone all'interno del CCD, ciascun pixel che "osserva" una sorgente che emette flusso luminoso  $F$  acquisirà un numero di conteggi  $N_*$  dato da:

$$N_* = \frac{F \cdot \Omega_i \cdot T \cdot A \cdot t}{E} \quad (2.1)$$

Dove:  $\Omega_i$  è l'angolo solido sotteso dal pixel,  $A$  e  $T$  sono rispettivamente l'area di raccolta luce e l'efficienza di trasmissione del sistema ottico,  $E$  è l'energia del singolo fotone e  $t$  il tempo totale di esposizione.

In Fig. 2.1 è riportata un'immagine astronomica destinata allo studio delle proprietà di galassie distanti. L'aspetto dell'immagine stessa, e gli istogrammi dei

valori dei pixel rende subito evidente come, accanto alle sorgenti (stelle e in gran parte galassie), una componente importante, e in certi punti dominante dell'immagine sia il rumore. I pixel in cui sono presenti le sorgenti presentano un elevato range di valori in rapporto al livello di rumore presente: non solo parte delle sorgenti sono appena distinguibili dal rumore, ma anche sorgenti molto evidenti presentano una struttura diffusa e i loro "contorni" vanno a confondersi con il rumore circostante. Come visto in precedenza il "segnale" è dato dal numero di fotoni provenienti dalla sorgente che incidono sul singolo pixel. Di contro, il rumore ha diverse componenti: 1) il rumore Poissoniano del segnale stesso ( $\sqrt{N_*}$ ), 2) rumore dovuto a fotoni non provenienti dalla sorgente ma a luce spuria raccolta dal telescopio o a emissione atmosferica ("sky background"). 3) rumore elettronico di tipo shot-noise dovuto alla lettura del sensore ("readout-noise") e 4) rumore termico, cioè elettroni di conduzione generati da agitazione termica all'interno del reticolo cristallino ("dark-current"). In gran parte dei casi il termine di rumore dominante nei pixel non occupati da sorgenti è dovuto allo sky-background che, con buona approssimazione, può essere considerato Gaussiano.

**Coordinate e dimensione dei pixel.** Oltre al valore dei conteggi, dovuto al segnale e al contributo di rumore, a ogni pixel di una immagine astronomica viene associata la relativa posizione *assoluta* in coordinate angolari sulla sfera celeste, solitamente espressa nelle unità convenzionali di *ascensione retta* e *declinazione*. Analogamente è pratica comune indicare in frazioni di grado (solitamente secondi d'arco,  $1 \text{ arcsec} = \frac{1}{3600} \text{ grado}$ ) la dimensione lineare dei pixel di una camera per osservazioni astronomiche.

### 2.1.1 Acquisizione e calibrazione delle immagini

Per consentire una accurata analisi del flusso e della posizione delle sorgenti osservate le immagini astronomiche devono essere sottoposte ad alcuni accorgimenti di manipolazione e calibrazione.

Bisogna innanzitutto considerare che un funzionamento ottimale dei sensori CCD richiede che il flusso luminoso ricevuto dai pixel non sia così elevato da portare il CCD in un regime di funzionamento non lineare o da giungere alla saturazione dei pixel, cioè al riempimento completo della buca di potenziale del semiconduttore da parte degli elettroni di conduzione. A tale scopo è pratica comune costruire l'immagine finale come somma di più immagini ciascuna acquisita con un tempo di esposizione "ottimale" del CCD considerando l'intensità delle sorgenti più luminose presenti e l'intensità del background alla lunghezza d'onda di osservazione. La somma delle immagini (cioè dei relativi tempi di esposizione) permette di raggiungere il rapporto tra segnale e rumore desiderato ( $S/N \propto \sqrt{t}$ ).

I diversi frame vengono in genere osservati leggermente shiftati tra loro allo scopo di ottenere un'immagine finale maggiormente uniforme e priva di difetti. Le immagini "raw" ottenute dalla camera CCD non sono direttamente utilizzabili per l'analisi scientifica ma devono prima essere sottoposte a varie operazioni di "riduzione" o "detrending". Tali operazioni sono classificabili nei seguenti passi:

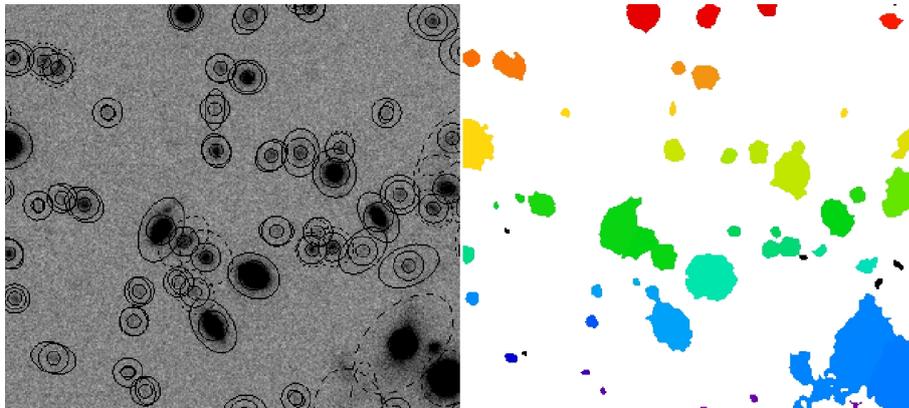


Figura 2.2: Esempio di segmentazione di un'immagine astronomica (a destra) e di aperture, ellittiche e circolari, utilizzate per la misura del flusso delle sorgenti (sinistra).

equalizzazione in modo da compensare le naturali differenze tra i pixel nella conversione tra fotoni e foto-elettroni; mascheramento dei pixel difettosi; sottrazione di componenti a bassa frequenza (corrente di bias nella lettura del CCD, componenti costanti dello sky-background). Sommare frame leggermente shiftati tra loro permette di eliminare difetti e disuniformità nell'immagine finale ("immagine coadded").

Accanto al processo di riduzione e di somma dei frame, è estremamente importante per lo sfruttamento scientifico delle immagini il processo di *calibrazione*. Tale processo ha lo scopo di definire con la maggior precisione possibile i valori di flusso (calibrazione fotometrica) e posizione (calibrazione astrometrica) degli oggetti presenti sull'immagine. La base comune delle procedure di calibrazione è l'uso di sorgenti "standard" il cui flusso e posizione assoluti siano noti con elevata precisione da precedenti osservazioni. Alla fine dei processi di riduzione/somma e calibrazione a ciascun pixel dell'immagine astronomica è attribuibile un valore assoluto (e relativa incertezza) tanto del flusso ricevuto che della posizione osservata in cielo.

## 2.2 Analisi di immagini astronomiche

L'analisi delle immagini astronomiche si basa su due passi fondamentali: innanzitutto un processo di *segmentazione*, per individuare le sorgenti presenti e misurare le loro dimensioni sull'immagine, e dunque la misura del loro flusso, o "fotometria". Tra i software concepiti allo scopo prendiamo come riferimento il codice SExtractor<sup>1</sup>[4], sicuramente il più utilizzato attualmente, quanto meno nel campo dell'astrofisica extragalattica. Tale software individua gli oggetti presenti sull'im-

<sup>1</sup><http://www.astromatic.net/software/sextractor>

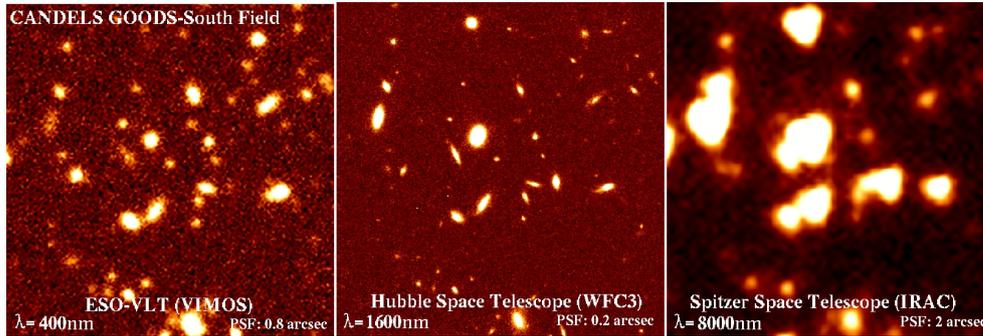


Figura 2.3: Osservazioni multi-banda della stessa regione di cielo: le immagini sono state acquisite con diversi strumenti, da terra e da spazio, e presentano risoluzioni differenti sia in termini di PSF che di dimensione dei pixel.

magine tramite una procedura di *thresholding*. Definita a scelta dell'utente una soglia di significatività in S/N, la segmentazione viene eseguita misurando il valore mediano del background e la sua RMS su una versione filtrata dell'immagine e dunque individuando le sorgenti (aree di pixel connesse) con S/N al di sopra della soglia richiesta (Fig. 2.1.1 a destra). La misura del flusso viene poi eseguita calcolando la somma dei valori dei pixel (dopo la sottrazione della componente di background) entro aperture circolari o ellittiche [20] che includono le sorgenti (pannello a sinistra in Fig. 2.1.1). Gli oggetti presenti sull'immagine sono infine raccolti in un catalogo che ne elenca le proprietà: posizione, dimensione, e flusso delle sorgenti espresso in conteggi o, solitamente, nella scala delle magnitudini:

$$m = -2.5 \log(N_*) + m_0 \quad (2.2)$$

Dove  $m_0$ , magnitudine di "zero-point" dell'immagine, viene determinata in precedenza tramite il processo di calibrazione fotometrica (Par. 2.1.1) e serve per riportare la scala in unità standard.

### 2.2.1 Osservazioni multi-banda

In molti casi per studiare le proprietà fisiche degli oggetti celesti è necessario quantificare la loro emissione luminosa in funzione della lunghezza d'onda. Allo scopo la stessa zona di cielo viene osservata con più immagini acquisite selezionando intervalli di lunghezza d'onda ben definiti con appositi filtri posti all'interno del sistema ottico. Questo genere di analisi (osservazioni "multi-banda") viene spesso eseguito anche utilizzando sensori o telescopi diversi, ciascuno appositamente costruito in modo da essere maggiormente sensibile in un dato range spettrale. È questo il caso ad esempio di zone di cielo osservate nel range visibile o vicino infrarosso ( $0.4-2 \mu m$ ) con telescopi da terra e da telescopi spaziali a quelle lunghezze d'onda che sono assorbite dall'atmosfera, quali l'ultravioletto o il medio e lontano infrarosso ( $>3 \mu m$ ). Tali immagini si presentano notevolmente diverse in quan-

to a risoluzione (e.g. Fig. 2.2), non solo a causa delle differenti caratteristiche degli strumenti ma anche per la variazione della point spread function a diverse lunghezze d'onda e l'effetto più o meno marcato della turbolenza atmosferica.

### 2.3 Visualizzazione e formati delle immagini

Per le immagini astronomiche è stato definito un apposito formato standard denominato Flexible Image Transport System<sup>2</sup> (FITS) che permette di avere a disposizione tutte le informazioni necessarie per assegnare a ciascuno dei pixel valori fisici di flusso e posizione. L'header dei formati FITS contiene infatti una serie di keyword che forniscono valori di riferimento per la fotometria (magnitudine di zero-point, lunghezza d'onda) e l'astrometria (dimensione angolare dei pixel, coordinate angolari di un pixel di riferimento, angolo di rotazione dell'immagine ecc). I programmi di manipolazione e analisi di immagini FITS utilizzano delle librerie standard (C o FORTRAN) per la lettura e scrittura delle immagini. Per visualizzare le immagini FITS esistono appositi programmi<sup>3</sup> che rendono disponibili le informazioni contenute nell'header e forniscono opzioni di scelta della look-up-table (LUT) di visualizzazione. Infatti, a causa dell'elevato range di valori dei pixel, un normale mapping lineare sull'intervallo [0,255] usato dai display non è sempre ottimale, ed è spesso necessario utilizzare LUT logaritmiche o LUT lineari su una finestra di valori (e.g. convertendo in bianco/nero il 10% di pixel con valori più alti/bassi). Per lo stesso motivo è spesso di aiuto alla visualizzazione l'utilizzo di immagini in falsi colori (e.g. Fig. 2.2).

### 2.4 Potenzialità delle tecniche di super-risoluzione applicate a immagini astronomiche

Date le caratteristiche delle osservazioni di imaging astronomico possiamo individuare sia motivazioni e opportunità che aspetti critici dell'introduzione di tecniche di super-risoluzione in questo campo.

- **Potenzialità.** La disponibilità di immagini ad alta risoluzione è estremamente importante in astrofisica. Innanzitutto una maggiore risoluzione permette una migliore identificazione delle stesse sorgenti sulle immagini, e non è raro il caso che problemi di "blending" tra sorgenti, o la scarsa risoluzione di alcune osservazioni astronomiche impediscano una chiara comprensione dei fenomeni fisici legati agli oggetti celesti. Le immagini ad alta risoluzione permettono poi uno studio più accurato della morfologia delle singole sorgenti, un aspetto di grande interesse per lo studio di come queste si sono formate ed evolute. Inoltre, solo in immagini con sufficiente risoluzione è

---

<sup>2</sup><http://fits.gsfc.nasa.gov/>

<sup>3</sup>e.g. SAOImage ds9, <http://hea-www.harvard.edu/RD/ds9/site/Home.html>

possibile lo studio accurato di alcuni fenomeni fisici, su tutti quelli legati al “lensing gravitazionale” e in particolare del cosiddetto “lensing debole” che costituisce una delle frontiere della moderna cosmologia osservativa. Infine, le tecniche SR possono risultare molto utili per l’analisi “multi-banda” (Par. 2.2.1) di dataset composti da più immagini con diversa risoluzione, uno dei problemi più delicati nell’analisi di osservazioni astronomiche. Tecniche attualmente utilizzate per l’analisi di questo genere di dati (ad es. il cosiddetto “PSF-matching”) consistono nel degradare le immagini ad alta risoluzione in modo che possano essere analizzate in modo ‘coerente’ assieme alle immagini a risoluzione inferiore (in particolare definendo uguali aree per la misura del flusso). Disporre di tecniche che migliorano la risoluzione delle immagini costituirebbe un enorme passo avanti per l’analisi di osservazioni astronomiche.

- **Opportunità.** Diverse caratteristiche delle osservazioni astronomiche si prestano all’utilizzo di tecniche di super-risoluzione. Innanzitutto, come descritto nel Par. 2.1 è frequente che le immagini vengano costruite dalla somma di più frame leggermente shiftati tra loro. La creazione dell’immagine finale si basa attualmente sull’uso di algoritmi di interpolazione (lineare o polinomiale) per riportare i diversi frame su una griglia comune [e.g. 5]: l’uso di tecniche SR potrebbe costituire in questo caso una più che valida alternativa. In secondo luogo la calibrazione astrometrica, un procedimento per cui a oggi esistono tecniche molto affidabili, permette di determinare gli shift tra le diverse immagini con elevata precisione: ciò evita il bisogno di utilizzare tecniche di ‘registrazione’ delle immagini che sono un aspetto abbastanza delicato dell’uso di tecniche SR sulle fotografie e video [e.g. 15]. Infine, la stessa Point Spread Function può essere determinata con grande accuratezza (anche in funzione delle posizione) misurando la distribuzione di luce di stelle presenti sulle immagini stesse. Nelle osservazioni di imaging astronomico le stelle possono infatti essere a tutti gli effetti considerate puntiformi dal momento che, per via delle enormi distanze da noi, la loro dimensione angolare è inferiore al potere risolutivo di qualunque telescopio.
- **Aspetti critici.** Nell’applicare tecniche di super-risoluzione a immagini astrofisiche bisogna comunque tenere presenti alcune problematiche. Soprattutto vi è la necessità di mantenere inalterata la calibrazione fotometrica in modo da poter eseguire delle misure precise del flusso delle sorgenti anche sulle immagini super-risolte. Inoltre bisogna comprendere come l’elevato livello di rumore di questo genere di immagini influisca sull’uso di tecniche SR. Da un punto di vista tecnico la costruzione di un software di super-risoluzione di immagini astronomiche deve tenere presente le specificità del formato FITS, in particolare l’utilizzo di keyword di registrazione astrometrica, e possibilmente basarsi su linguaggi di alto livello e paradigmi di parallelizzazione e ottimizzazione in modo da poter gestire nel modo più

efficiente immagini a 16bit che possono essere anche di grandi dimensioni (di norma  $10^6$ - $10^7$  pixel).

## Capitolo 3

# Algoritmi variazionali di Super-Risoluzione

Allo scopo di sperimentare le tecniche di super-risoluzione su immagini astronomiche abbiamo scelto per il presente lavoro di studiare in dettaglio due schemi appartenenti alla famiglia delle tecniche variazionali (Par. 1.1.1): questo genere di tecniche sono di fatto le più studiate in letteratura e sono notevolmente versatili sia da un punto di vista teorico (regolarizzazione delle soluzioni) che computazionale (algoritmi di minimizzazione).

Riprendendo quanto esposto nel Cap. 1 e in particolare l'equazione 1.7. Essendo noto il modello di formazione dell'immagine (Par. 1.1), e dunque l'operatore  $K$  che include le operazioni di convoluzione ( $H$ ), warping ( $W$ ) e downsampling ( $D$ ), a partire da  $n$  immagini  $\underline{x}_L$  di dimensioni  $M \times N$  possiamo ricostruire un'immagine a risoluzione maggiore  $\underline{x}_H$  di dimensioni  $rM \times rN$  cercando il minimo di una funzione "energia" che include un termine di regolarizzazione  $f(\underline{x}_H)$  convesso in modo da assicurare che il problema sia ben posto:

$$E(\underline{x}_H) = \frac{1}{2}(\|\underline{x}_L - K\underline{x}_H\|_2^2 + \lambda f(\underline{x}_H)) \quad (3.1)$$

In quanto segue considereremo sempre casi in cui  $f(\underline{x}_H)$  è anche differenziabile o resa tale, e il funzionale  $E(\underline{x}_H)$  è strettamente convesso. Nei paragrafi 3.1 e 3.2 illustreremo gli algoritmi studiati in cui il termine di regolarizzazione  $f(\underline{x}_H)$  è dato rispettivamente dalla norma L2 e dalla norma L1 di Huber del gradiente  $\nabla \underline{x}_H$ . In 3.3 vengono descritti gli schemi numerici utilizzati. Infine, nel paragrafo 3.4 viene presentato il codice FORTRAN-90 scritto per implementare gli algoritmi descritti e testarli su immagini astronomiche.

### 3.1 Algoritmi di minimizzazione con regolarizzazione L2

Ci basiamo in quanto segue sui lavori di Zomet and Peleg [41] e Mitzel et al. [23]. Riscriviamo l'equazione 3.1 includendo come termine di regolarizzazione la norma

L2 del gradiente dell'immagine HR: qualitativamente una tale scelta corrisponde all'ipotesi che l'immagine che vogliamo ricostruire sia regolare e priva di forti discontinuità.

$$E(\underline{x}_H) = \frac{1}{2}(\|\underline{x}_L - K\underline{x}_H\|_2^2 + \lambda\|\nabla\underline{x}_H\|_2^2) \quad (3.2)$$

La condizione di minimo si ottiene eguagliando a zero la derivata rispetto a  $\underline{x}_H$  di  $E(\underline{x}_H)$ :

$$\nabla E(\underline{x}_H) = K^T(K\underline{x}_H - \underline{x}_L) - \lambda\Delta\underline{x}_H = \sum_{n=1}^N W_n^T H_n^T D^T (DH_n W_n \underline{x}_H - \underline{x}_L) - \lambda\Delta\underline{x}_H = 0 \quad (3.3)$$

La corrispondente soluzione può essere cercata in modo iterativo tramite uno schema di massima discesa del gradiente a passo costante:

$$\underline{x}_H^{m+1} = \underline{x}_H^m - \beta \cdot \nabla E(\underline{x}_H) = \underline{x}_H^m - \beta \cdot \left[ \sum_{n=1}^N W_n^T H_n^T D^T (DH_n W_n \underline{x}_H^m - \underline{x}_L) - \lambda\Delta\underline{x}_H^m \right] \quad (3.4)$$

Essendo  $E(\underline{x}_H)$  strettamente convesso il metodo converge a partire da qualunque punto iniziale scegliendo un  $\beta$  che rispetti la condizione  $\frac{1}{\beta} < L$  con  $L$  costante di contrazione [e.g. 24] per cui vale:

$$L \leq |\lambda|\|\Delta\| + \|K^T K\| = 8|\lambda| + 1 \quad (3.5)$$

$$\text{Essendo } \|K^T K\| = 1 \text{ e } \|\Delta\| \leq \|\text{div}\| \cdot \|\nabla\| = 2\sqrt{2} \cdot 2\sqrt{2}.$$

### Metodo del gradiente coniugato

In aggiunta alla minimizzazione per massima discesa a passo costante consideriamo un metodo di discesa per gradiente coniugato a passo ottimale [e.g. 15]. In questo caso riscriviamo le iterazioni nel seguente modo:

$$\underline{x}_H^{m+1} = \underline{x}_H^m + \beta_m d_m \quad (3.6)$$

dove:

$$d_0 = -\nabla E(\underline{x}_H^0)$$

$$d_m = -\nabla E(\underline{x}_H^m) + \alpha_m d_{m-1}$$

La costante  $\alpha_m$  è definita dal criterio di Fletcher-Reeves:

$$\alpha_m = \frac{\|\nabla E(\underline{x}_H^m)\|^2}{\|\nabla E(\underline{x}_H^{m-1})\|^2}$$

mentre il passo  $\beta_m$  si ottiene ponendo:

$$\frac{\partial E(\underline{x}_H^m + \beta_m d_m)}{\partial \beta_m} = 0$$

da cui, essendo  $Q = K^T K - \lambda \Delta$ , risulta:

$$\beta_m = -\frac{(d_m, \nabla E(\underline{x}_H^m))}{(d_m, Q d_m)} \quad (3.7)$$

### 3.2 Algoritmo con termine di regolarizzazione L1

Come alternativa agli algoritmi sopra descritti basati sulla norma L2 scegliamo il metodo proposto da Unger et al. [36]. Tale metodo è basato sulla norma L1 di Huber [17] così definita in riferimento a una immagine  $\underline{x}$  di dimensioni  $M \times N$ :

$$\|\underline{x}\|_\epsilon^h = \sum_{0 \leq i, j \leq MN} |x_{i,j}|_\epsilon h^2 \quad (3.8)$$

dove:

$$|x_{i,j}|_\epsilon = \frac{|x_{i,j}|^2}{2\epsilon} \quad \text{se } |x_{i,j}| \leq \epsilon$$

$$|x_{i,j}|_\epsilon = |x_{i,j}| - \frac{\epsilon}{2} \quad \text{se } |x_{i,j}| > \epsilon$$

Con  $\epsilon$  definito in base alla “mean absolute deviation”  $MAD = \text{median}[\underline{x} - \text{median}(\underline{x})]$  come suggerito da Holland and Welsch [16]:

$$\epsilon = \frac{1.345 \cdot MAD}{0.6746}$$

La funzione energia da minimizzare proposta dagli autori è dunque la seguente:

$$E(\underline{x}_H) = \|\underline{x}_L - K \underline{x}_H\|_{\epsilon_d}^r + \lambda \|\nabla \underline{x}_H\|_{\epsilon_u} \quad (3.9)$$

In cui  $\epsilon_u$  ed  $\epsilon_d$  sono riferiti rispettivamente a  $\underline{x}_H$  e a  $\underline{x}_L$ .

Una tale scelta è motivata dal diverso comportamento della norma di Huber, e in generale delle norme tipo L1 (e.g. algoritmi “total variation”) che preservano bordi “netti” all’interno dell’immagine, a differenza della norma L2 che tende a produrre immagini uniformi. L’algoritmo presentato dagli autori sfrutta lo schema primario-duale proposto da Pock et al. [29] per la minimizzazione di funzionali nella forma:

$$\min_{x \in X} \max_{y \in Y} [\langle Kx, y \rangle + G(x) - F^*(y)] \quad (3.10)$$

Sfruttando principi di dualità [in particolare la trasformata di Legendre-Fenchel, 31], gli autori riscrivono il problema nella forma:

$$\min_{\hat{x}_H} \max_{\underline{p}, \underline{q}} \left[ \langle \nabla \hat{x}_H, \underline{p} \rangle_Y - \frac{\epsilon_u}{2\lambda} \|\underline{p}\|^2 - \delta_{(\|\underline{p}\| \leq \lambda^2)} + \sum_{i=1}^n \left( \langle \underline{q}_i, K \hat{x}_H - \underline{x}_L \rangle_X - \frac{\epsilon_d}{2r^2} \|\underline{q}_i\|^2 - \delta_{(\|\underline{q}_i\| \leq r^2)} \right) \right] \quad (3.11)$$

In cui  $\underline{p} \in Y = \mathbb{R}^{r^2 MN} \times \mathbb{R}^{r^2 MN}$  e  $\underline{q} \in X = \mathbb{R}^{MN}$  sono le variabili duali, ed è definita la funzione indicatrice  $\delta_\Sigma = 0$  per  $x \in \Sigma$  e  $\delta_\Sigma = \infty$  altrimenti,

La soluzione  $\underline{x}_H$  viene trovata a partire dagli  $n$  frame  $\underline{x}_L$  tramite le seguenti iterazioni:

$$\underline{p}^{m+1} = \Pi_{B_0^\lambda} \left( \frac{\underline{p}^m + \sigma \nabla \underline{x}_H^m}{1 + \frac{\sigma \epsilon_u}{\lambda}} \right) \quad (3.12)$$

$$\underline{q}^{m+1} = \Pi_{B_0^{r^2}} \left( \frac{\underline{q}^m + \sigma r^2 (K \underline{x}_H^m - \underline{x}_L)}{1 + \frac{\sigma \epsilon_d}{r^2}} \right) \quad (3.13)$$

$$\hat{x}_H^{m+1} = \hat{x}_H^m - \tau \left( -\text{div}(\underline{p}^{m+1}) + r^2 \sum_{i=1}^n (K^T \underline{x}_L) \right) \quad (3.14)$$

$$\underline{x}_H^{m+1} = 2\hat{x}_H^{m+1} - \hat{x}_H^m \quad (3.15)$$

In cui vengono utilizzati i seguenti operatori di proiezione bidimensionale e unidimensionale:

$$\Pi_{B_0^\lambda}(\underline{p}) = \frac{\underline{p}}{\max\left(1, \frac{\|\underline{p}\|}{\lambda}\right)} \quad (3.16)$$

$$\Pi_{B_0^{r^2}}(\underline{q}) = \frac{\underline{q}}{\|\underline{q}\|} \cdot \min(\|\underline{q}\|, r^2) \quad (3.17)$$

La convergenza al minimo globale di  $E(\underline{x}_H)$  è assicurata dalla scelta dei passi:  $\tau = \frac{r}{\sqrt{\lambda L^2}}$ ,  $\sigma = \frac{1}{\sqrt{\tau L^2}}$ , con  $L = \lambda \|\nabla\| = \lambda \cdot 2\sqrt{2}$ ,

### 3.3 Schemi numerici

L'elaborazione numerica degli algoritmi sopra descritti richiede l'utilizzo nel discreto degli operatori gradiente  $\nabla$  e divergenza  $\text{div}()$  e l'implementazione degli operatori  $K = DHW$  e  $K^T = W^T H^T D^T$ .

Per la definizione di  $\nabla$  e  $\text{div}()$  seguiamo quanto descritto in Unger et al. [36]. Data un'immagine  $\underline{v}$  su una griglia regolare bidimensionale  $M \times N$  con indici  $1 \leq i \leq M$  e  $1 \leq j \leq N$ , definiamo uno spazio vettoriale finito  $X = \mathbb{R}^{MN}$  con prodotto scalare:

$$\langle \underline{v}, \underline{w} \rangle_X = \sum_{i,j} v_{i,j} \cdot w_{i,j} \quad (3.18)$$

Dato lo spazio vettoriale  $Y = \mathbb{R}^{MN} \times \mathbb{R}^{MN}$  l'operatore gradiente  $\nabla : X \rightarrow Y$  viene definito usando differenze finite e condizioni al contorno di Neumann:

$$(\nabla v)_{i,j} = \left( (\delta_x^+ v)_{i,j}, (\delta_y^+ v)_{i,j} \right)^T \quad (3.19)$$

con:

$$\begin{aligned} (\delta_x^+ v)_{i,j} &= v_{i+1,j} - v_{i,j} && \text{se } i < M \\ (\delta_x^+ v)_{i,j} &= 0 && \text{se } i = M \\ (\delta_y^+ v)_{i,j} &= v_{i,j+1} - v_{i,j} && \text{se } j < N \\ (\delta_y^+ v)_{i,j} &= 0 && \text{se } j = N \end{aligned}$$

L'operatore  $div() : Y \rightarrow X$  è definito come aggiunto dell'operatore gradiente, in modo cioè che dato  $\underline{p} \in Y$  sia:  $\langle \nabla \underline{v}, \underline{p} \rangle_Y = -\langle \underline{v}, div(\underline{p}) \rangle_X$ .

Risulta dunque:

$$(div(\underline{p}))_{i,j} = (\delta_x^- p^x)_{i,j} + (\delta_y^- p^y)_{i,j} \quad (3.20)$$

con:

$$\begin{aligned} (\delta_x^- p^x)_{i,j} &= p_{i,j}^x - p_{i-1,j}^x && \text{se } 0 < i < M \\ (\delta_x^- p^x)_{i,j} &= 0 && \text{se } i = 0 \\ (\delta_x^- p^x)_{i,j} &= -p_{i-1,j}^x && \text{se } i = M \\ (\delta_y^- p^y)_{i,j} &= p_{i,j}^y - p_{i,j-1}^y && \text{se } j < N \\ (\delta_y^- p^y)_{i,j} &= 0 && \text{se } j = 0 \\ (\delta_y^- p^y)_{i,j} &= -p_{i,j-1}^y && \text{se } j = N \end{aligned}$$

Come suggerito da Zomet and Peleg [41] implementiamo gli operatori di down-sampling ( $D$ ), warping ( $W$ ), convoluzione ( $H$ ), e i loro aggiunti  $D^T$  (upsampling),  $W^T$  (back-warping) e  $H^T$  tramite le seguenti operazioni su immagini:

- L'applicazione di  $D$  con fattore di downsampling  $r$  a una immagine  $x_H$  di  $rM \times rN$  pixel corrisponde a una operazione di decimazione: al pixel  $(i,j)$  dell'immagine  $x_L$  ( $M \times N$  pixel) in output viene assegnato il valore ottenuto dalla somma dei pixel  $[i', j'] \in [r(i-1)+1 : ir, r(j-1)+1 : jr]$  di  $x_H$ :

$$(x_L)_{i,j} = \sum_{i'=(ri-r+1)}^{ri} \sum_{j'=rj-r+1}^{rj} (x_H)_{i',j'} \quad (3.21)$$

L'operazione inversa  $D^T$  corrisponde a un ingrandimento di un fattore  $r$  senza interpolazione dell'immagine  $x_L$ :

$$(x_H)_{i',j'} = \frac{1}{r^2} (x_L)_{i,j}, \quad i = \lfloor \frac{i'}{r} \rfloor; \quad j = \lfloor \frac{j'}{r} \rfloor \quad (3.22)$$

Dove il fattore  $1/r^2$  è richiesto dalla condizione di conservare il flusso totale dell'immagine.

- Per l'operatore  $W$  consideriamo traslazioni senza rotazione: in questo caso, l'operazione  $u = Wv$  con fattori di traslazione lungo i due assi di  $(i_{off}, j_{off})$  corrisponde all'assegnazione:

$$(u)_{i,j} = (v)_{i+i_{off},j+j_{off}} \quad (3.23)$$

L'operazione inversa  $W^T$  equivale alla precedente operazione con offset  $(-i_{off}, -j_{off})$ .

- L'operatore di convoluzione  $u = Hv$  con un kernel  $h(l, m)$  di dimensioni  $(2k_1+1) \times (2k_2+1)$  esegue la seguente operazione:

$$(u)_{i,j} = \sum_{l=-k_1}^{k_1} \sum_{m=-k_2}^{k_2} (v)_{i+l,j+m} \cdot h(l, m) \quad (3.24)$$

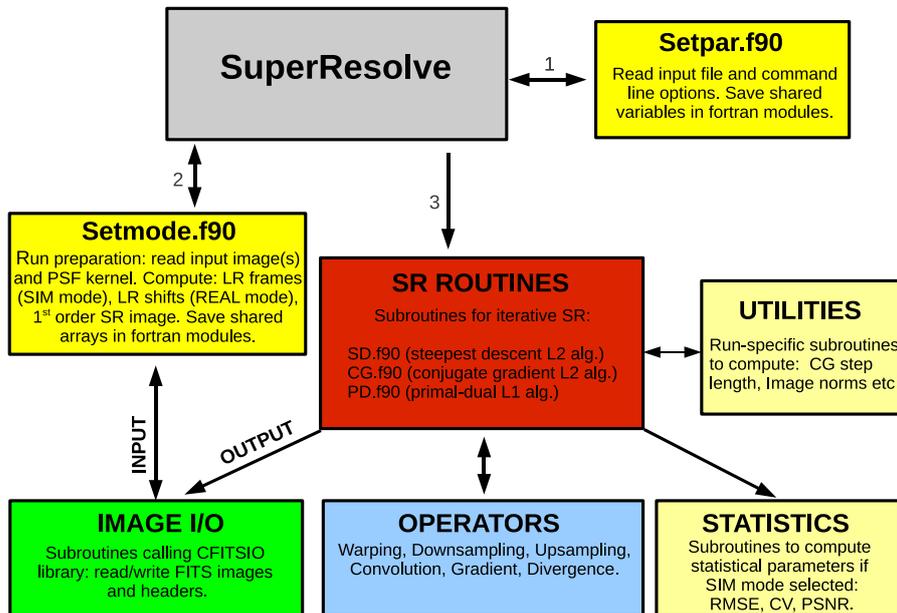
L'operatore aggiunto  $H^T$  esegue l'operazione di convoluzione descritta ma applicando il kernel  $\hat{h}(l, m) = h(-l, -m)$  (nel caso di kernel isotropi ovviamente si ha  $\hat{h}(l, m) = h(-l, -m) = h(l, m)$ ). Affinchè sia rispettata la condizione di conservare il flusso totale dell'immagine è necessario che:

$$\sum_{l,m} h(l, m) = 1 \quad (3.25)$$

### 3.4 Implementazione Software

Gli algoritmi e schemi sopra descritti sono stati implementati in un codice denominato `SuperResolve` appositamente elaborato per l'uso su immagini astronomiche. Sebbene linguaggi quali MATLAB e IDL possano fornire un utile, e rapido, ambiente di sviluppo per software di analisi di immagini, per la scrittura di `SuperResolve` la scelta è ricaduta su un linguaggio di programmazione non interpretato per ottenere una maggiore efficienza computazionale, decisiva per l'uso su dataset astrofisici che possono raggiungere rilevanti dimensioni. In particolare abbiamo scelto il FORTRAN-90 rispetto al C/C++ motivati dall'ampia diffusione di questo linguaggio in ambito astrofisico, in vista tanto dell'inclusione di `SuperResolve` nelle software-pipelines di analisi di immagini già esistenti, quanto di una possibile pubblicazione del codice. La disponibilità delle librerie Fortran FITSIO<sup>1</sup> per la lettura e scrittura di immagini astrofisiche in formato FITS (Cap. 2.3) è un ulteriore importante vantaggio. `SuperResolve` utilizza infatti esclusivamente immagini in formato FITS: normali immagini fotografiche in

<sup>1</sup><http://heasarc.gsfc.nasa.gov/fitsio/>



qualunque formato possono essere convertite in FITS tramite il comando Linux `convert` o il software Gimp. Come verrà discusso più in dettaglio nel Paragrafo 3.5, il codice `SuperResolve` è parallelizzato tramite comandi openMP e può essere utilizzato in modalità seriale o parallela sul numero desiderato di CPU modificando la variabile di Shell `OMP_NUM_THREADS`.

### 3.4.1 Struttura e funzionamento di SuperResolve

In Figura 3.4.1 è riportato lo schema di funzionamento di `SuperResolve`: per favorire future estensioni e modifiche del software esso è concepito con un'architettura modulare suddividendo le diverse funzioni in subroutine Fortran che condividono variabili tramite file MODULE. Gli algoritmi di super-risoluzione presentati nei paragrafi 3.1 e 3.2 sono scritti ognuno come una separata subroutine del programma e fanno a loro volta uso di subroutine e funzioni che implementano gli operatori descritti in 3.3.

Per consentire vari tipi di test delle tecniche di super-risoluzione il programma può funzionare sia in modalità di "simulazione" che in modalità applicativa. Nel primo caso un'immagine fornita in input viene internamente degradata generando frame a bassa risoluzione shiftati tra loro di multipli seminteri di pixel: gli algoritmi di super-risoluzione vengono utilizzati per ricostruire l'immagine originale valutando infine l'accuratezza del risultato. Nel secondo caso deve essere fornita una lista di frame da cui ricavare l'immagine super-risolta, con il primo frame della lista che fornisce la posizione di riferimento per essa.

Per consentire lo svolgimento di test di diverso tipo e l'uso in modalità *batch* il funzionamento del programma viene gestito a run-time specificando una serie di

opzioni da un file ASCII di input o da linea di comando. Tramite il file di input l'utente indica la scelta dell'algoritmo, dei relativi parametri di funzionamento e le diverse opzioni del programma che descriveremo in dettaglio nel paragrafo successivo: il main file `SuperResolve.f90` ha esclusivamente il compito di eseguire le chiamate alle diverse subroutine di gestione dell'I/O e alle subroutine di algoritmo.

Il primo passo eseguito dal programma è la chiamata alla subroutine `Setpar.f90` che legge le opzioni di input. Successivamente `SuperResolve` chiama la subroutine `Setmode.f90` che, in base a quanto richiesto dall'utente, prepara il run di super-risoluzione leggendo i frame a bassa risoluzione o generandoli internamente a partire da un'immagine di riferimento. Infine `SuperResolve.f90` chiama la subroutine relativa all'algoritmo di super-risoluzione scelto e, al termine delle iterazioni, scrive in output il risultato.

### Opzioni del programma

Un esempio di file di input di `SuperResolve` è riportato in Figura 3.1. Nel file sono indicate le opzioni, identificate da parole chiave in lettere maiuscole e, separato da spazi, il relativo valore da assegnare. Le righe che iniziano col carattere `#` o qualunque carattere successivo a `#` viene ignorato dal programma e serve esclusivamente come commento per l'utente: nell'esempio in Figura 3.1 alle righe con indicate le opzioni di `SuperResolve` segue, commentata con `#`, una sintetica spiegazione dell'opzione e del genere di valore richiesto (numero intero INT, reale REAL o stringa di caratteri CHAR).

### Modalità di esecuzione

- **MODE**  
*Valori consentiti:* SIM; REAL.  
 Scelta della modalità di funzionamento: 1) "simulazione" (SIM), da un'immagine fornita in input vengono generati internamente frame a bassa risoluzione; 2) modalità "reale" (REAL), i filename dei frame a bassa risoluzione vengono passati tramite una lista ASCII.
- **METHOD**  
*Valori consentiti:* SD; CG; PD.  
 Scelta dell'algoritmo: super-risoluzione con regolarizzazione L2 e minimizzazione per massima discesa (SD) o minimizzazione col metodo del gradiente coniugato (CG). Super-risoluzione con regolarizzazione L1 secondo l'algoritmo primario-duale di Unger et al. [36] (PD).
- **NGAIN**  
*Valori consentiti:* numeri REAL.  
 L'opzione indica in guadagno risoluzione richiesto (fattore  $r$  in Par. 3.1-3.3). Nel caso `MODE=REAL` ponendo `NGAIN<0` il valore viene regolato interna-

```

#####I/O#####
INPUT      gal.fits  #(CHAR) input HR image for SIM or file with list of REAL images
OUTPUT     prova.fits #(CHAR) output super-resolved image
VERBOSE    N        #(CHAR) prompt to screen while iterating
SAVE_TMP   N        #(CHAR) save all output for LAMBDATEST=LIST ?
HR0_INPUT  INTERNAL #(CHAR) if different from INTERNAL: name of input zero-order HR image
PSF        mask1_0.1.conv #(CHAR) PSF kernel (only used when DECONVOLVE=Y)
CROP_PSF   0        #(INT) if set to N > 0 crop PSF kernel to central 2N+1 rows and columns
N_IMA     -1        #(INT) if set to N < use all images in list (REAL) or N=NGAIN*NGAIN (SIM)
#
#####RUN-OPTIONS#####
#
MODE       SIM      #(CHAR) simulations (SIM) or REAL shifted images
METHOD     SD       #(CHAR) SD=steepest descent L2; CG=conjugate gradient L2; PD=primal dual L1 algorithm.
NGAIN      4        #(INT) res. enhancement. If <0 when mode=REAL NGAIN=sqrt(number of images)
SHIFTS     LIST     #(CHAR) for REAL MODE: FITS: read from FITS header; LIST: read shifts from list
DECONVOLVE N        #(CHAR) Y=apply deconvolution; N=only super-resolution
NORM8BIT   N        #(CHAR) normalize output to [0,255]
#
#####STOP-CRITERIA#####
#
ACC        0.01     #(REAL) stop when ||Residuals||/||LR_FRAMES||<ACC
TOL        0.001    #(REAL) stop when ||X(k+1)-X(k)||/||X(k)||<TOL
MAXITER    1000     #(INT) max number of iterations
#
#####REGULARIZATION#####
#
LAMBDA     1.0      #(REAL) Regularization constant
LAMBDATEST SIMP     #(CHAR) SIMP,LIST
LAMBDAVEC  0.01,1.0,5.0 #(CHAR) Regularization constant
LAMBDA_ACC 0.05     #(REAL) Accuracy on best Regularization constant for test BEST

```

Figura 3.1: File di configurazione di SuperResolve, le opzioni (in lettere maiuscole) sono seguite dal relativo parametro separato da spazi, tutto ciò che è preceduto dal carattere # viene ignorato dal programma e serve esclusivamente come commento per l'utente.

mente dal programma pari alla radice del numero di immagini fornite dal file ASCII indicato con INPUT.

- SHIFTS

*Valori consentiti:* FITS; LIST;

Questa opzione ha valore solo nel caso MODE=REAL usando immagini astronomiche: con SHIFTS=FITS gli offset tra le immagini vengono determinati in base alle keyword astrometriche contenute negli header. Se SHIFTS=LIST il programma richiede una lista ASCII con tre colonne: 1) Nome file immagine; 2) offset lungo asse X; 3) offset lungo asse Y.

- DECONVOLVE

*Valori consentiti:* Y; N.

Ciascuno degli algoritmi indicati con METHOD è disponibile (in subroutine separate) sia in una versione che include l'applicazione dell'operatore di convoluzione/deconvoluzione ( $H$  e  $H^T$  in Par. 3.1-3.3) che in una versione di "pura" super-risoluzione: in questo secondo caso, chiamato con l'opzione DECONVOLVE=N, vengono applicati solo gli operatori di decimazione/upsampling ( $D$  e  $D^T$ ) e warping ( $W$  e  $W^T$ ) per ricostruire un'immagine a risoluzione maggiore senza compensare l'effetto della PSF.

- NORM8BIT

*Valori consentiti:* Y; N.

Opzione per consentire l'uso su immagini fotografiche a 8bit scegliendo NORM8BIT=Y.

### Criteri di arresto

- ACC

*Valori consentiti:* numeri REAL.

Definito l'errore relativo  $ERR$  tra l'immagine super-risolta e i frame in input:

$$ERR = \frac{\|x_{LR} - Kx_{HR}^m\|}{\|x_{LR}\|} \quad (3.26)$$

il programma si arresta appena  $ERR$  risulta inferiore al valore indicato con l'opzione ACC.

- TOL

*Valori consentiti:* numeri REAL.

Il programma si arresta quando l'aggiornamento relativo così definito:

$$\delta x_{HR} = \frac{\|x_{HR}^{m+1} - x_{HR}^m\|}{\|x_{HR}^m\|} \quad (3.27)$$

risulta inferiore al valore indicato con l'opzione TOL.

- MAXITER

*Valori consentiti:* numeri INTEGER.

Massimo numero di iterazioni consentite.

### Parametri di regolarizzazione

- LAMDATEST

*Valori consentiti:* SIMP; LIST.

Se LAMDATEST=SIMP viene eseguito un solo run usando il parametro di regolarizzazione specificato con l'opzione LAMBDA. Se LAMDATEST=LIST vengono eseguiti run di super-risoluzione per ciascuno dei valori indicati tramite l'opzione LAMDAVEC. In quest'ultimo caso, se SAVE\_TMP=Y le immagini super-risolte relative ai diversi parametri di regolarizzazione vengono salvate in output e, se MODE=SIM, viene salvato in output anche un file ASCII con i parametri statistici descritti nel Par. 3.4.2: root-mean-square-error (RMSE), coefficiente di variazione (CV) e peak signal to noise ratio (PSNR).

- LAMDAVEC

*Valori consentiti:* numeri REAL (massimo 20 valori) separati da virgola.

Valori del parametro di regolarizzazione per LAMDATEST=LIST.

- LAMBDA  
*Valori consentiti:* numeri REAL.  
Valore del parametro di regolarizzazione per LAMBDATEST=SIMP.

### Input/Output

- INPUT  
*Valori consentiti:* stringhe (max 60 caratteri).  
Input principale del programma: nome file dell'immagine da usare per le simulazioni (MODE=SIM) o nome del file ASCII contenente la lista (su una colonna) dei frame a bassa risoluzione (se MODE=REAL) e dei loro offset rispetto al primo frame (se anche indicato SHIFTS=LIST).
- OUTPUT  
*Valori consentiti:* stringhe (max 60 caratteri).  
Nome del file di output su cui verrà salvata l'immagine super-risolta.
- ADD\_HEADER  
*Valori consentiti:* Y; N.  
Se ADD\_HEADER=Y vengono inserite nell'header FITS dell'immagine di output le keyword astrometriche relative alla posizione del primo frame (che fornisce il sistema di riferimento per la creazione dell'immagine super-risolta).
- VERBOSE  
*Valori consentiti:* Y; N.  
Se VERBOSE=Y il programma stampa a schermo a ogni iterazione  $m$  l'aggiornamento relativo "acc" e l'errore relativo "tol" (vedere Fig. 3.2) definiti in Eq.3.27 e 3.26.
- SAVE\_TMP  
*Valori consentiti:* Y; N.  
Questa opzione ha valore solo nel caso LAMBDATEST=LIST : se SAVE\_TMP=Y le immagini super-risolate ottenute per i valori del parametro di regolarizzazione indicati con LAMBDAVEC vengono salvate con il nome scelto tramite l'opzione OUTPUT seguito, prima dell'estensione .fits da un numero progressivo. Ad esempio se come OUTPUT è indicato il nome test.fits, e si sceglie di svolgere la simulazione con 10 diversi valori di  $\lambda$ , i relativi file sono denominati test\_1.fits, test\_2.fits ecc.
- HR0\_INPUT  
*Valori consentiti:* stringhe (max 60 caratteri).  
Come default il programma utilizza come approssimazione di partenza dell'algoritmo di super-risoluzione ( $x_{HR}^0$ ) una interpolazione bilineare del primo frame a bassa risoluzione calcolata da una apposita subroutine (BilinUpsample.f90): tale scelta viene mantenuta specificando HR0\_INPUT=INTERNAL. In alternativa si può passare al programma tramite

```

File Edit View Search Terminal Help
marco@node8:~/SR/Devel/NEW> ./SuperResolve.exe lena_input.txt -INPUT lena.fits -METHOD CG -MODE SIM -NGAIN 2 -LAMBDA 0.5 -OUTPUT lena_SRCG_lam05.fits -VERBOSE Y

READING PARAMETERS FILE : lena_input.txt
INPUT      : lena.fits
OUTPUT     : lena_SRCG_lam05.fits
USING METHOD : CG

!WARNING: ALREADY EXISTING FILES WILL BE OVERWRITTEN!

RUNNING IN SIMULATION MODE WITH NGAIN= 2
USING : 4 IMAGES

#Preparing run...
#Generating HR zero-order image
--> RMSE of zero-order HR approximation (bilinear upsampling):
RMSE= 18.484356
CV= 0.12526521
PSNR= 22.794716

#Generating LR images
#Running...
it= 1 acc=0.110601E+00 tol=0.972483E-01
it= 2 acc=0.844238E-01 tol=0.308856E-01
it= 3 acc=0.817666E-01 tol=0.108537E-01
it= 4 acc=0.817372E-01 tol=0.667326E-02
it= 5 acc=0.817453E-01 tol=0.545147E-02
it= 6 acc=0.817435E-01 tol=0.513344E-02

```

Figura 3.2: Esempio di uso del programma SuperResolve da terminale Linux.

$HR\_INPUT$  il nome file di una immagine  $x_{HR}^0$ : ad esempio la mediana, o la somma, upsamplend dei frame a bassa risoluzione, o, nel caso di immagini astronomiche, una immagine della risoluzione desiderata ma ottenuta a una diversa lunghezza d'onda (cfr. Par. 2.3).

- PSF

*Valori consentiti:* stringhe (max 60 caratteri).

Nome del file ASCII contenente la matrice del kernel PSF. Il programma controlla internamente che il numero di elementi  $h(l, m)$  sia dispari e ne normalizza i valori affinché  $\sum_{l,m} h(l, m) = 1$ .

- CROP\_PSF

*Valori consentiti:* numeri INTEGER.

Per velocizzare il programma in fase di test il kernel PSF può venire internamente troncato specificando un numero intero positivo tramite CROP\_PSF. Per mantenere un numero di elementi dispari CROP\_PSF serve a indicare la *semi-ampiezza* desiderata: ad esempio passando con l'opzione PSF un kernel di  $21 \times 21$  elementi, se CROP\_PSF=4 vengono utilizzati soli i  $2 \cdot 4 + 1 = 9$  elementi  $h(l, m)$  centrali.

- N\_IMA

*Valori consentiti:* numeri INTEGER.

Numero di frame a bassa risoluzione da utilizzare. Come default N\_IMA=-1 per indicare che vanno usate tutte le immagini nella lista di input (se MODE=REAL) o generati internamente NGAIN×NGAIN frame a bassa risoluzione (se MODE=SIM). Alternativamente, con N\_IMA=N, (dove N>1) vengono usate solo le prime N immagini nella lista o generati N frame a bassa risoluzione.

### 3.4.2 Analisi statistica

Se utilizzato in modalità di simulazione con l'opzione `MODE=SIM` il programma `SuperResolve` al termine delle iterazioni chiama la subroutine `dostat.f90` che effettua test statistici per giudicare l'accuratezza dell'immagine SR ottenuta rispetto all'immagine ad "alta risoluzione" di partenza. Viene innanzitutto calcolato lo scarto quadratico medio RMSE tra l'immagine SR e l'immagine HR, entrambe di dimensione  $N \times M$ :

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M (HR(i, j) - SR(i, j))^2}{N \cdot M}} \quad (3.28)$$

Assieme al RMSE vengono valutati il coefficiente di variazione CV, definito come rapporto tra l'RMSE e il valore quadratico medio dei pixel HR di partenza, e il Peak-signal to noise ratio (PSNR) cioè il rapporto, espresso in decibel, tra il valore massimo dell'immagine originale e il RMSE:

$$CV = \frac{RMSE}{\sqrt{\langle HR^2 \rangle}} = \frac{RMSE}{\sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M HR(i, j)^2}{N \cdot M}}} \quad (3.29)$$

$$PSNR = 20 \cdot \log_{10} \left( \frac{\max(HR_{i,j})}{RMSE} \right) \quad (3.30)$$

Nel caso venga selezionata l'opzione `SIM` assieme all'opzione `LAMBDATEST=LIST` i test vengono effettuati per ogni immagine super-risolta calcolata con i diversi valori del parametro di regolarizzazione indicati con `LAMBDAVEC`. Al termine dell'esecuzione i valori di RMSE, CV e PSNR ottenuti vengono salvati nel file ASCII `stat.log`.

## 3.5 Note sull'implementazione

Nella programmazione di `SuperResolve` sono stati attuati i seguenti accorgimenti per l'ottimizzazione e la parallelizzazione:

- **Uso di array dinamici.** I valori dei pixel delle immagini sono salvati in array dinamici bidimensionali definiti tramite l'istruzione `FORTRAN 90 ALLOCATABLE` per consentire l'uso efficiente del programma su immagini di diverse dimensioni. Gli array contenenti i valori delle immagini di input sono definiti in appositi `MODULE` condivisi dalle varie subroutine. Ove necessario i cicli che eseguono operazioni di manipolazione delle immagini sfruttano array di appoggio "locali" anch'essi di tipo dinamico che vengono allocati (istruzione `ALLOCATE`) e de-allocati (`DEALLOCATE`) in base alle necessità del programma.

- **Programmazione ridondante per evitare costrutti di tipo “branch” nei loop.** Dal momento che l’operazione di convoluzione è particolarmente onerosa da un punto di vista computazionale, ciascuno degli algoritmi discussi è implementato in due versioni, con e senza deconvoluzione (applicazione degli operatori  $H$  e  $H^T$ ). Ciò consente di usare gli algoritmi per la “pura” super-risoluzione (cioè senza compensare l’effetto convolutivo della PSF) tramite l’opzione DECONVOLVE=N senza sovraccaricare i loop con istruzioni di tipo IF che ne rallenterebbero l’esecuzione.

- **Ottimizzazione dei loop.** Gli operatori downsampling ( $D$ ), upsampling ( $D^T$ ), warping ( $W$  e  $W^T$ ), convoluzione ( $H$  e  $H^T$ ), divergenza ( $div()$ ) e gradiente ( $\nabla$ ), sono implementati tramite specifiche subroutine che eseguono tramite loop do innestati le operazioni indicate in Par. 3.3. I loop sono costruiti tenendo conto dell’accesso in memoria per colonne del FORTRAN, ordinando dunque i cicli innestati dall’indice più esterno al più interno degli array bidimensionali.

All’interno dei loop le operazioni di divisione (e.g. Eq. 3.22) sono sostituite da moltiplicazioni (meno onerose computazionalmente) per fattori inversi calcolati fuori dal loop.

Dato un kernel di dimensioni  $(2k_1+1) \times (2k_2+1)$  l’operazione di convoluzione di un’immagine  $M \times N$  (Eq. 3.24) richiederebbe delle istruzioni IF all’interno del loop in modo da non incorrere in errori di accesso in memoria per i pixel di bordo ( $[1:k_1-1, 1:k_2-1]$  e  $[M-k_1+1:M, N-k_2+1:N]$ ): per evitare ciò l’operazione viene eseguita su un’array temporaneo di dimensioni  $(M+2k_1) \times (N+2k_2)$ , aggiungendo cioè una “cornice” di pixel con valore nullo attorno all’immagine.

- **Uso di librerie pubbliche** La lettura/scrittura delle immagini in formato FITS e degli header viene eseguita sfruttando le subroutine FORTRAN della libreria FITSIO in base a quanto definito nella relativa Fortran Programmer Reference Guide<sup>2</sup>. Per l’algoritmo PD il calcolo dei fattori epsilon della norma di Huber richiede il calcolo della mean absolute deviation, e dunque della mediana: per ottenere un calcolo efficiente di tale parametro SuperResolve utilizza l’algoritmo di “heapsort” [19] nella versione FORTRAN 90 di Numerical Recipes Vol.2 [30].
- **Parallelizzazione shared memory con openMP.** Per l’algoritmo in Par. 3.1 l’operazione  $K^T(K\underline{x}_H^m - x_L)$  in Eq. 3.3 è eseguita calcolando prima gli N “residui”  $R_n = K\underline{x}_H^m - x_L^{(n)}$  e dunque gli N aggiornamenti  $A_n = K^T R_n$ . Per l’algoritmo primario-duale in Par. 3.2 le due operazioni sono naturalmente separate nei passi in Eq. 3.13 e Eq. 3.14. In entrambi i casi questi sono i loop di gran lunga più costosi da un punto di vista computazionale. Essendo gli N passi di ciascuno dei loop indipendenti tra loro (il calcolo di  $R_1$  non

<sup>2</sup><http://heasarc.gsfc.nasa.gov/docs/software/fitsio/c/fuser/fitsio.html>

richiede il calcolo di  $R_2$  ecc) possono essere facilmente parallelizzati tramite istruzioni openMP in modo che vengano eseguiti su unità logiche differenti. Nella pratica si sono strumentati i loop con comandi openMP-FORTRAN nel seguente modo:

```
!$omp parallel do default(private)
    !shared(xshift,yshift,nxHR,nyHR,Nima,
    !uHR,uLR,Res,psf,nxLR,nyLR,nrows,ncols)
!schedule(runtime)
do, ii=1,Nima
    ...
```

in cui sono definite come `shared` le variabili globali necessarie per il calcolo di ciascuno dei passi: `shift`, dimensioni delle immagini, array contenenti  $x_L$  e  $x_H^m$  ecc. L'istruzione `!schedule(runtime)` permette di gestire da terminale il numero di CPU utilizzate tramite la variabile di shell `OMP_NUM_THREADS`. Un accorgimento analogo è stato utilizzato per il calcolo di  $Qd_m = (K^T K - \lambda \Delta)d_m$  nel passo ottimale del gradiente coniugato (cfr. Eq. 3.7).

- **Compilazione del programma e uso delle directory** Per mantenere separati file sorgente, eseguibili e ambiente di lavoro SuperResolve viene compilato con `gfortran` tramite `Makefile` considerando il seguente albero di directory:

```
-\SR_root-\obj
    -\bin
    -\src
```

Tramite l'istruzione `make` il `Makefile` contenuto in `SR_root` cerca i file sorgente (`.f90`) in `src`, salva gli object-files (`.o`) e i file di modulo (`.mod`) in `obj` e l'eseguibile in `bin` creando infine un link simbolico dell'eseguibile nella directory `SR_root` stessa. Nel `Makefile` sono incluse come flag le istruzioni per linkare in fase di compilazione le librerie FITSIO (comando `-lcfitsio -lm`) e openMP (comando `-fopenmp`) e una flag aggiuntiva con l'opzione di ottimizzazione con valore di default `-O3`.



## Capitolo 4

# Esperimenti numerici

Nel presente capitolo esploreremo gli algoritmi di super-risoluzione su immagini di diverso tipo per valutare sia la capacità degli schemi di ricostruire immagini ad alta risoluzione che l'efficienza computazionale del programma `SuperResolve`. Analizzeremo innanzitutto (paragrafo 4.1) simulazioni su immagini fotografiche utilizzando l'opzione `MODE=SIM` del programma `SuperResolve`. Come illustrato nel Cap. 3, scegliendo questa opzione l'immagine HR fornita in input viene internamente utilizzata per generare i frame a bassa risoluzione (LR) a cui applicare l'algoritmo di super-risoluzione. L'opzione `NGAIN` del programma indica il rapporto tra la dimensione del pixel dei frame LR da simulare e il pixel HR originale: internamente al programma vengono generati i frame LR richiesti shiftando l'immagine HR di multipli semi-interi del pixel LR, e dunque degradandola tramite downsampling ed eventuale convoluzione con PSF kernel fornito dall'utente (se tale opzione è specificata nel file di configurazione di `SuperResolve`). I frame LR vengono successivamente utilizzati dal programma per generare l'immagine super-risolta.

Successivamente valuteremo l'applicabilità delle tecniche SR su immagini astrofisiche. Nel paragrafo 4.2.1 verranno presentati test effettuati in modo analogo al caso di immagini fotografiche, cioè tramite la modalità interna di simulazione del programma. Nel paragrafo 4.2.2 testeremo invece la modalità `MODE=REAL` del programma `SuperResolve` utilizzando simulazioni di dataset astronomici. Infine, nel paragrafo 4.3 valuteremo le prestazioni del programma in funzione delle dimensioni e numero delle immagini utilizzate, del guadagno in risoluzione richiesto e del numero di CPU utilizzate tramite parallelizzazione `openMP`. Per semplicità di esposizione ci riferiremo ai due algoritmi descritti nel Par. 3.1 [41, 23] e nel Par. 3.2 [36] rispettivamente come "SR-L2" e "SR-L1".

### Criteri di arresto

In tutti i test qui presentati utilizzeremo i seguenti criteri di arresto sulla convergenza dell'algoritmo specificati tramite i parametri di configurazione `TOL` e

ACC (Par. 3.4):

$$\|\underline{x}_{SR}^{m+1} - \underline{x}_{SR}^m\| < 0.001 \cdot \|\underline{x}_{SR}^m\| \quad (4.1)$$

OR

$$\|\underline{x}_{LR} - K\underline{x}_{SR}^m\| < 0.01 \cdot \|\underline{x}_{LR}\| \quad (4.2)$$

Includiamo inoltre la condizione di un numero massimo di 1000 iterazioni (opzione MAXITER).

## 4.1 Immagini fotografiche

Come primo test del programma applichiamo i due algoritmi implementati in SuperResolve su immagini in scala di grigio con proprietà differenti: un ritratto fotografico (“Lena”, e.g. Fig.4.1) e l’immagine di un codice a barre (“bar-code”, e.g. Fig.4.3).

Nei test che seguono miriamo a comprendere: 1) quanto gli algoritmi di super-risoluzione implementati forniscano risultati più affidabili rispetto alla semplice interpolazione delle immagini a bassa risoluzione. 2) quanto l’accuratezza dell’immagine super-risolta dipenda dal parametro di regolarizzazione  $\lambda$ ; 3) quanto le differenti regolarizzazioni utilizzate dagli algoritmi (norma L2 e norma di Huber L1 del gradiente) siano efficaci su immagini con diverse caratteristiche: l’immagine Lena infatti, pur presentando un buon livello di dettaglio in alcuni punti (occhio, cappello ecc), risulta caratterizzata da ampie zone approssimativamente uniformi e scarsamente contrastate. All’opposto, l’immagine bar-code è fortemente contrastata presentando numeri e linee nere su sfondo bianco.

Applichiamo i due algoritmi di super-risoluzione a entrambe le immagini sperimentando per tentativi diversi valori del parametro di regolarizzazione  $\lambda$  sia per il caso di super-risoluzione “pura”, cioè ricerca del solo guadagno in risoluzione, che super-risoluzione con deconvoluzione utilizzando un kernel PSF gaussiano con FWHM di 3 pixel per degradare ulteriormente le immagini LR e tenendo conto di esso nel ricostruire l’immagine SR. Variamo il numero di frame e l’incremento di risoluzione richiesto secondo i seguenti casi:

1. Guadagno in risoluzione di un fattore  $r=2$  a partire da  $N=4$  frame LR.
2. Guadagno in risoluzione di un fattore  $r=4$  a partire da  $N=16$  frame LR.

Giudicheremo l’accuratezza delle immagini super-risolate dalla misura di: scarto quadratico medio (RMSE, Eq. 3.28), coefficiente di variazione (CV, Eq. 3.29) e “peak signal to noise ratio” (PSNR, Eq. 3.30). Nelle tabelle 4.1 e 4.2 sono riportati i valori migliori di RMSE, CV e PSNR ottenuti dai due algoritmi per le immagini

esaminate, paragonandoli al caso di un semplice upsampling bilineare dei frame LR. I migliori risultati ottenuti per il caso  $r=4$ ,  $N=16$  con e senza deconvoluzione sono mostrati in Fig. 4.1-4.4. Nelle Figure 4.5-4.8 riportiamo, per entrambi gli algoritmi, SR-L2 e SR-L1, l'andamento del RMSE al variare del parametro di regolarizzazione  $\lambda$  nei due casi sopra indicati.

Tabella 4.1: Super-risoluzione

Test	Parametro	SR-L2	SR-L1	Interpolazione bilineare
Lena, $r=2$ , $N=4$	RMSE:	10.92	10.96	18.5
	CV:	0.073	0.074	0.12
	PSNR:	27.37	27.38	22.8
Lena, $r=4$ , $N=16$	RMSE:	18.08	16.35	31.73
	CV:	0.12	0.11	0.21
	PSNR:	23.0	23.9	18.1
bar-code, $r=2$ , $N=4$	RMSE:	16.3	11.07	29.5
	CV:	0.079	0.054	0.14
	PSNR:	23.86	27.2	18.7
bar-code, $r=4$ , $N=16$	RMSE:	25.07	17.1	62.3
	CV:	0.12	0.083	0.3
	PSNR:	20.14	23.47	12.2

Tabella 4.2: Super-risoluzione con deconvoluzione

Test	Parametro	SR-L2	SR-L1	Interpolazione bilineare
Lena, $r=2$ , $N=4$	RMSE:	14.99	15.05	23.4
	CV:	0.10	0.10	0.16
	PSNR:	24.66	24.58	20.8
Lena, $r=4$ , $N=16$	RMSE:	20.5	19.07	32.3
	CV:	0.14	0.13	0.22
	PSNR:	21.9	22.5	18.0
bar-code, $r=2$ , $N=4$	RMSE:	21.5	21.05	37.3
	CV:	0.11	0.10	0.18
	PSNR:	21.48	21.67	16.7
bar-code, $r=4$ , $N=16$	RMSE:	27.03	26.73	63.5
	CV:	0.14	0.13	0.31
	PSNR:	19.5	19.6	12.1



Figura 4.1: Test su immagine Lena: immagine originale, esempio di frame LR, miglior risultato di super-risoluzione ( $r=4$  a partire da 16 frame LR) con regolarizzazioni L2 e L1



Figura 4.2: Test su immagini bar-code: immagine originale, esempio di frame LR, miglior risultato di super-risoluzione ( $r=4$  a partire da 16 frame LR) con regolarizzazioni L2 e L1



Figura 4.3: Come 4.1 ma includendo convoluzione/deconvoluzione con kernel Gaussiano FWHM=3pixel HR



Figura 4.4: Come 4.1 ma includendo convoluzione/deconvoluzione con kernel Gaussiano FWHM=3pixel HR

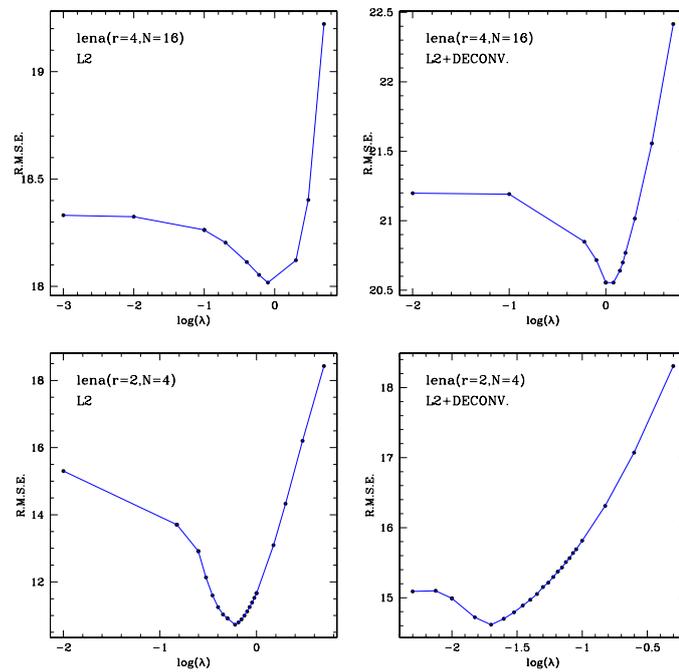


Figura 4.5: Scarto quadratico medio (RMSE) tra l'immagine super-risolta e l'immagine HR in funzione del parametro di regolarizzazione  $\lambda$ .

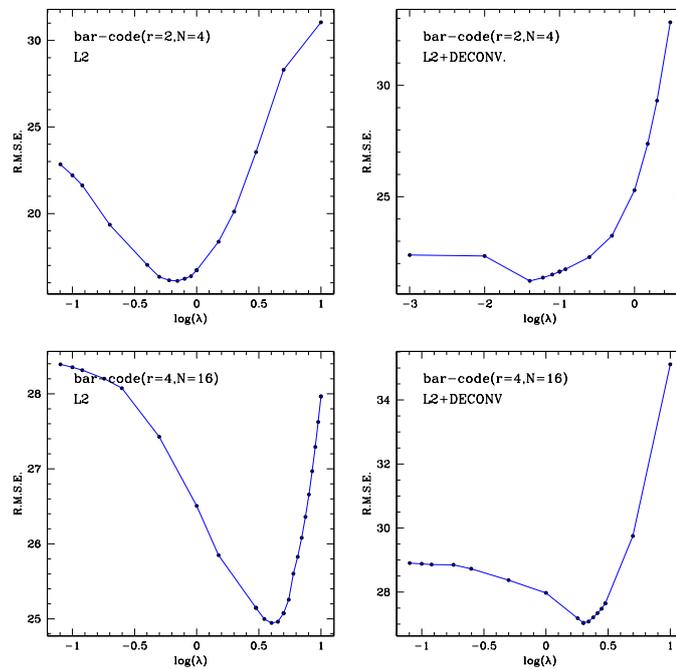


Figura 4.6: Test sull'immagine "bar-code", come Fig. 4.5.

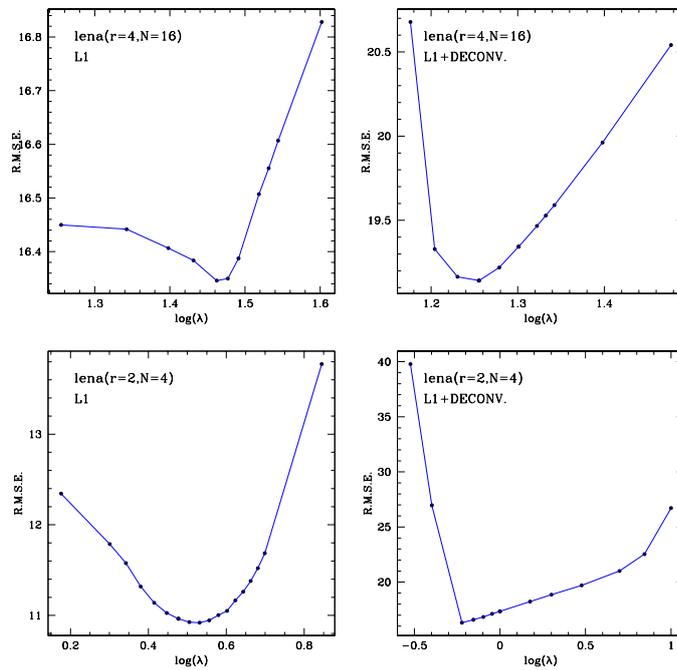


Figura 4.7: Test su immagine "Lena", regolarizzazione con norma L1 del gradiente, cfr Fig. 4.5.

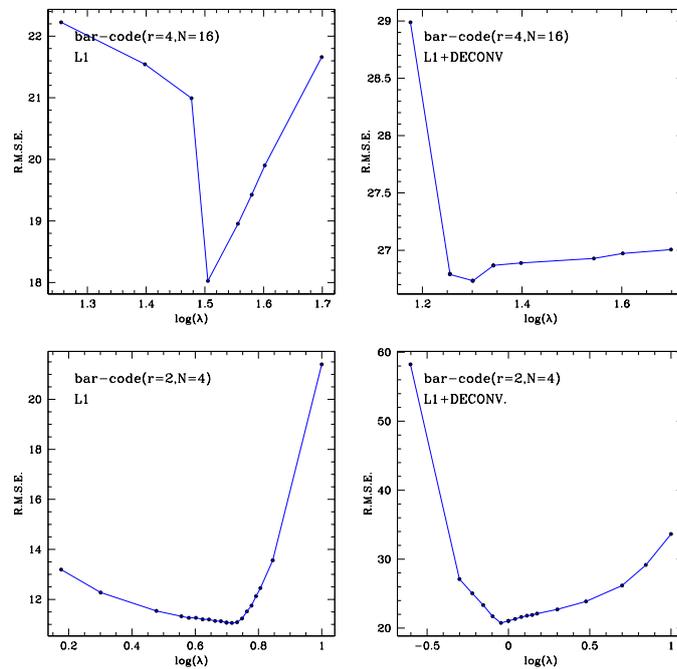


Figura 4.8: Come Fig. 4.7, test su immagine "bar-code"

## Valutazione dei test preliminari

Considerando i grafici dell'andamento del RMSE in funzione di  $\lambda$  e paragonando i migliori risultati ottenuti dai due algoritmi nei casi esaminati possiamo trarre le seguenti conclusioni:

- *Paragone tra Super-risoluzione e interpolazione:* In qualunque caso considerato le tecniche iterative di super-risoluzione ricostruiscono un'immagine molto più accurata di quella ottenibile da interpolazioni di immagini a bassa risoluzione, con un guadagno di un fattore due-tre in RMSE per i casi più complessi. Sebbene questo risultato possa sembrare scontato è bene tenerlo presente per quanto verrà discusso in seguito a riguardo delle immagini astrofisiche per le quali i rebinning con interpolazione sono comunemente usati.
- *Scelta del parametro di regolarizzazione:* in entrambi i casi vi è un intervallo abbastanza ampio (circa 0.5 dex) in  $\lambda$  attorno al valore ottimale in cui le prestazioni risultano accettabili, con un buon compromesso tra gli effetti di denoising e di deconvoluzione. La scelta di  $\lambda$  sembra comunque maggiormente critica negli esperimenti sull'immagine bar-code, e usando la regolarizzazione L1 con deconvoluzione.
- *Paragone tra SR con regolarizzazione L2 e SR-L1:* L'algoritmo SR-L1 mostra un rendimento decisamente migliore di SR-L2 nel caso di sola super-risoluzione sull'immagine bar-code: e.g. RMSE di  $\sim 11$  contro  $\sim 16$  nel caso di guadagno  $r=4$  da  $N=16$  frame. Sebbene SR-L1 dia risultati leggermente migliori anche per sola super-risoluzione sull'immagine Lena, le prestazioni dei due algoritmi sono confrontabili e le differenze decisamente minime nei casi di super-risoluzione con deconvoluzione.

### 4.1.1 Variazione del numero dei frame e casi estremi

Sulle stesse immagini di prova sperimentiamo ora due casi di super-risoluzione più complessi utilizzando sempre la modalità di simulazione del programma. Proviamo innanzitutto a variare il numero di immagini utilizzate nella ricostruzione con guadagno in risoluzione  $r=4$  (per il quale il numero di frame LR ottimale è  $\geq 16$ ). In Fig. 4.10 sono mostrate le immagini super-risolve migliori ottenute per il caso Lena dai due algoritmi utilizzando da 4 a 14 immagini LR e l'andamento del relativo RMSE in funzione del numero di frame LR utilizzati: come si può notare le prestazioni dello schema SR-L2 restano eccellenti anche con un numero basso di frame LR a disposizione. Di contro, SR-L1 presenta risultati leggermente migliori utilizzando almeno 8-10 frame a bassa risoluzione. Entrambi gli algoritmi producono risultati migliori rispetto all'interpolazione bilineare anche utilizzando solo 4 frame LR.

Come ultimo test cerchiamo di ottenere un guadagno in risoluzione di un fattore 8 a partire da 64 frame simulati a bassa risoluzione ulteriormente degradati

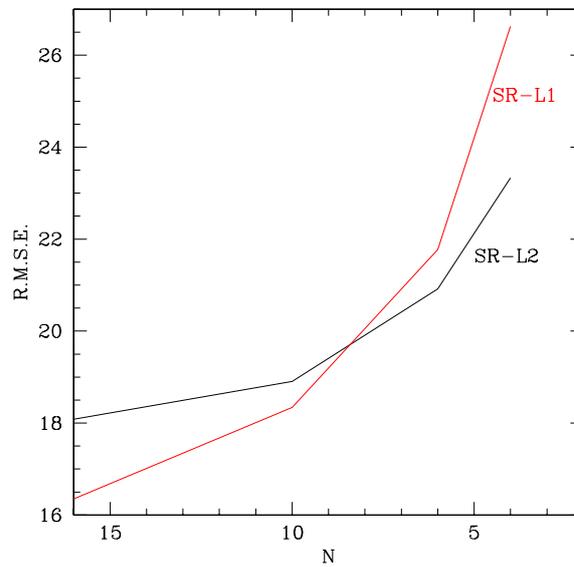


Figura 4.9: RMSE dell'immagine super-risolta al variare del numero N di frame LR a disposizione.



Figura 4.10: Risultati migliori sull'immagine Lena per guadagno in risoluzione  $r=4$  a partire da  $N=4$ , sino a  $N=14$ , frame LR.



Figura 4.11: Test su immagine Lena: super-risoluzione  $r=8$  a partire da 64 frame LR.



Figura 4.12: Test su immagine bar-code: super-risoluzione  $r=8$  a partire da 64 frame LR.

tramite convoluzione. Come si può vedere in Fig. 4.12, dove riportiamo il miglior risultato ottenuto con SR-L2 e SR-L1, anche questo test conferma la caocità delle due tecniche nel ricostruire un'immagine "leggibile". In base al RMSE la regolarizzazione L2 risulta maggiormente efficace su entrambe le immagini. Mentre per l'immagine Lena ciò è confermato anche da un esame visuale, nel caso dell'immagine bar-code, e a dispetto di parametri statistici peggiori, l'immagine ottenuta da SR-L1 risulta più realistica di quella ottenuta da SR-L2 e priva di zone irregolari evidenti in quest'ultima: sebbene ciò possa sembrare inaspettato concorda con quanto illustrato in dettaglio da Wang et al. [37]: il semplice parametro di RMSE può non essere una fedele rappresentazione di quello che è il giudizio umano sulla "somiglianza" tra due immagini, motivando gli autori a proporre un diverso strumento statistico, lo Structure Similarity Index (SSIM), così definito date due immagini (o porzioni)  $x$  e  $y$  di uguali dimensioni:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.3)$$

In cui  $\mu_x$ ,  $\sigma_x$ ,  $\mu_y$ ,  $\sigma_x^2$ ,  $\sigma_y^2$ ,  $\sigma_{xy}$  indicano rispettivamente: valor medio di  $x$ , valor medio di  $y$ , varianza di  $x$ , varianza di  $y$ , covarianza tra  $x$  e  $y$ . Per stabilizzare la divisione per piccoli denominatori sono introdotte le costanti  $c_1 = (L \cdot 0.01)^2$  e  $c_2 = (L \cdot 0.03)^2$  con  $L=2^{\#bit\ per\ pixel}-1$  (range dinamico delle immagini). Lo SSIM varia tra -1 a 1, col valore massimo ottenibile solo da due immagini identiche. A seguito del presente test abbiamo introdotto il calcolo del SSIM nella subroutine `dostat.f90` di `SuperResolve` (Par. 3.4.2): seguendo quanto suggerito da Wang et al. [37] il calcolo dello SSIM globale viene effettuato come valore medio degli SSIM calcolati su finestre di  $8 \times 8$  pixel. Il valore di SSIM ottenuto nel caso dell'immagine bar-code rispecchia quella che è l'impressione visiva, con un valore di 0.62 per l'immagine ottenuta da SR-L1 contro 0.54 per l'immagine super-risolta con l'algoritmo SR-L2.

## 4.2 Immagini astronomiche

I test su immagini fotografiche ci hanno permesso di appurare che gli algoritmi implementati forniscono buoni risultati di super-risoluzione anche in casi particolarmente complessi, quali un numero sub-ottimale ( $< r^2$ ) di frame LR a disposizione o ricerca di guadagno in risoluzione sino a un fattore  $r=8$ . Sebbene in alcuni casi la regolarizzazione L1 dia esiti visivamente e statisticamente migliori, la regolarizzazione L2 produce risultati più "stabili" in termini di minore dipendenza dal parametro  $\lambda$  e dal numero di immagini LR usate in input. In quanto segue studieremo le prestazioni dei due algoritmi su immagini astronomiche valutandone l'efficacia anche, e soprattutto, da un punto di vista di accuratezza delle immagini super-risolve per l'analisi del flusso luminoso dei singoli oggetti.

### 4.2.1 Simulazioni da immagini *Hubble Ultra-Deep Field*

Procedendo in modo analogo a quanto visto nel paragrafo precedente inizieremo usando il programma in modalità di simulazione: come primo test utilizziamo una porzione di 256x256 pixel di quella che è l'immagine astronomica a più alta risoluzione mai ottenuta. Essa è il risultato della somma di 116 immagini (tempo di osservazione complessivo pari a ~38 ore) ottenute dal Telescopio Spaziale Hubble (HST) a una lunghezza d'onda di 0.43nm (filtro B) osservando la zona di cielo denominata Hubble-Ultra Deep Field [HUDF, 3]. La porzione scelta comprende l'immagine di una galassia a "spirale" di cui grazie alla risoluzione di HST (~0.1 secondi d'arco) è possibile distinguere i dettagli anche nelle regioni più esterne. Nelle figure 4.13, 4.14, 4.15 sono mostrate le immagini super-risolve migliori ottenute dai due algoritmi per i casi  $r=2$  (da 4 frame LR di 128x128 pixel),  $r=4$  (da 16 frame LR di 64x64 pixel) e  $r=8$  (da 64 frame LR di 32x32 pixel), paragonate all'immagine ad alta risoluzione e al primo frame a bassa risoluzione generato dal programma. In tutti i casi le immagini LR sono ulteriormente degradate applicando un filtro di convoluzione Gaussiano di FWHM=5 pixel. I valori ottimali del parametro di regolarizzazione per i due algoritmi (ottenuti in base a test analoghi a quelli presentati in Fig. 4.5-4.8) risultano essere  $\lambda = 1.0$  e  $\lambda = 30.0$  rispettivamente per SR-L2 e SR-L1.

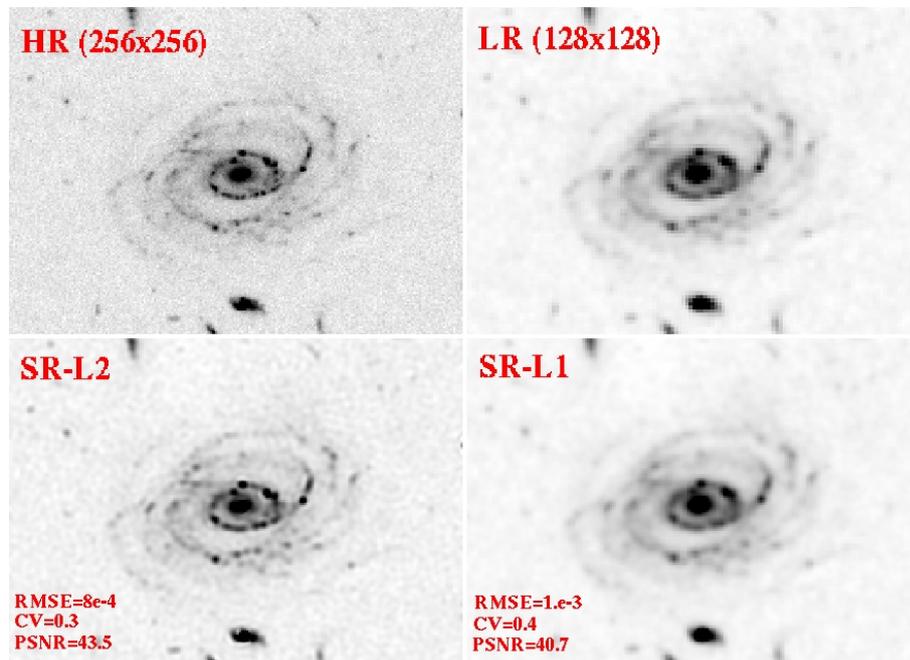


Figura 4.13: Porzione 256×256 dell'immagine HUDF originale (in alto a sinistra), frame simulato a risoluzione 2 volte inferiore (in alto a destra), immagini super-risolve da SR-L2 e SR-L1 (in basso) a partire da 4 frame a bassa risoluzione.

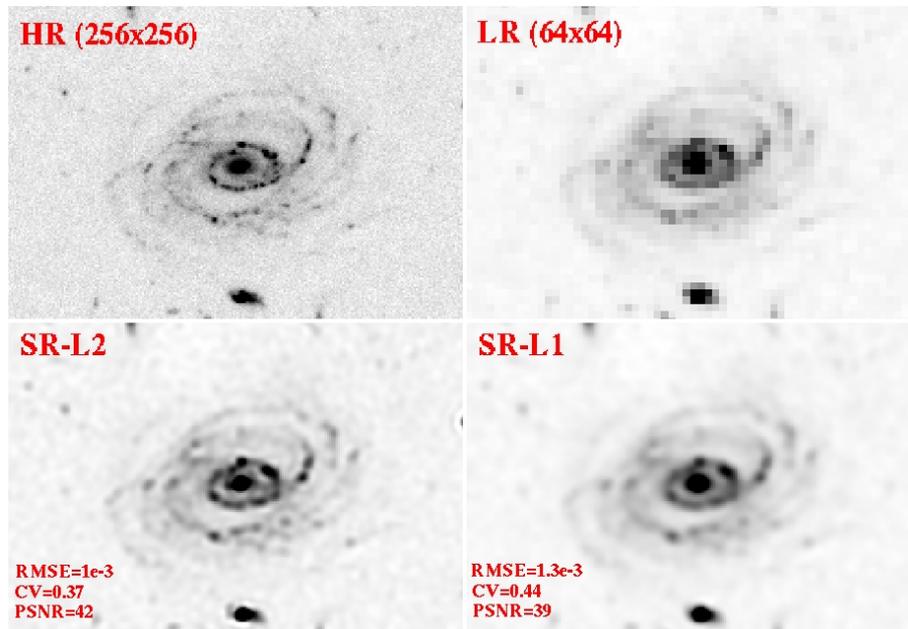


Figura 4.14: Test su HUDF come in Fig. 4.13 per super-risoluzione con guadagno  $r=4$  da 16 frame  $64\times 64$ .

I test statistici (riportati all'interno delle figure) indicano che SR-L2 produce risultati più fedeli in tutti i casi considerati. A un esame visuale le immagini ottenute con l'algoritmo SR-L1 risultano in effetti eccessivamente piatte, come se per effetto della regolarizzazione i dettagli meno contrastati siano stati "diffusi" sui pixel adiacenti. Questo primo test in un certo senso estremizza il paragone effettuato in precedenza nei test sull'immagine bar-code e l'immagine Lena, dove SR-L1 mostrava migliori prestazioni in presenza di forti contrasti e bordi ben definiti all'interno dell'immagine: nelle immagini astronomiche non compaiono quasi mai bordi "netti" ma al contrario oggetti piuttosto regolari i cui contorni si confondono con il rumore che domina le zone "vuote" dell'immagine.

### Effetti della super-risoluzione sulla calibrazione delle immagini

L'utilizzo scientifico delle immagini astronomiche richiede accurate misure del flusso delle sorgenti: affinché le tecniche di super-risoluzione siano utilizzabili su osservazioni astronomiche bisogna dunque verificare che esse non alterino la calibrazione delle immagini. Allo scopo analizziamo con il software SExtractor (cfr. Par. 2.2) una porzione di  $1024\times 1024$  pixel dell'immagine HUDF descritta in precedenza paragonando per ogni sorgente il flusso (espresso in magnitudini) misurato sull'immagine originale e quello misurato sull'immagine super-risolta. Consideriamo il caso di guadagno in risoluzione  $r=4$  a partire da 16 frame a bassa risoluzione generati internamente dal programma in modalità di simulazione e

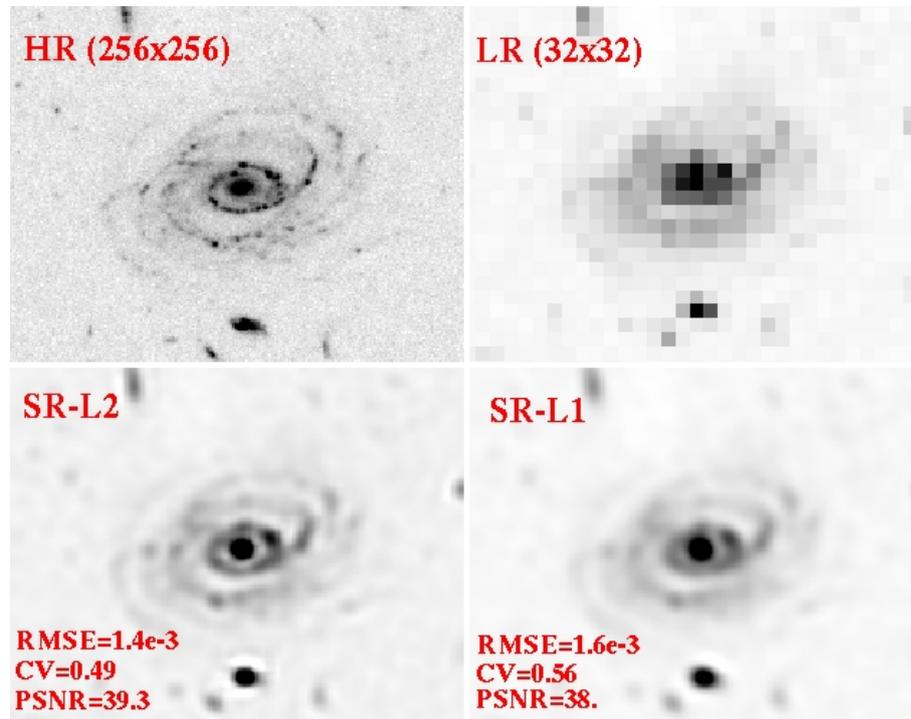


Figura 4.15: Test su HUDF come in Fig. 4.13 per super-risoluzione con guadagno  $r=8$  da 64 frame  $32 \times 32$ .

fissando il parametro di regolarizzazione  $\lambda$  ai valori ottimali ottenuti dai test sulla porzione  $256 \times 256$  discussa nel paragrafo precedente. In Fig. 4.16 sono riportati alcuni esempi di oggetti dall'immagine originale ("HR"), in un frame a bassa risoluzione e nelle immagini super-risolte.

In Fig. 4.17 è mostrato (pannello in basso) il paragone tra le magnitudini misurate sull'immagine originale e sull'immagine super-risolta con l'algoritmo SR-L2 e (pannello in alto) la misura della FWHM di stelle dall'immagine originale, dall'immagine super-risolta e dai frame a bassa risoluzione<sup>1</sup>. I medesimi test relativi all'immagine ottenuta con l'algoritmo SR-L1 sono riportati in Fig. 4.18. Per quanto riguarda SR-L2 questo test fornisce dei risultati estremamente interessanti. Gli oggetti nell'immagine super-risolta *non mostrano alterazioni sistematiche del flusso luminoso (magnitudini)*, la differenza è statisticamente consistente con zero, e variazioni sui singoli oggetti sono attribuibili essenzialmente all'effetto dell'incertezza, intrinseca al processo di misura e dovuta al rumore. La FWHM della PSF, espressa in unità fisiche (secondi d'arco) misurata sull'immagine super-risolta risulta inferiore a quella misurata sui frame a bassa risoluzione per quanto maggiore che sull'immagine originale: *l'algoritmo di super-risoluzione riesce ef-*

<sup>1</sup>la misura della FWHM delle stelle fornisce una stima della PSF apparendo esse effettivamente puntiformi per via delle enormi distanze da noi.

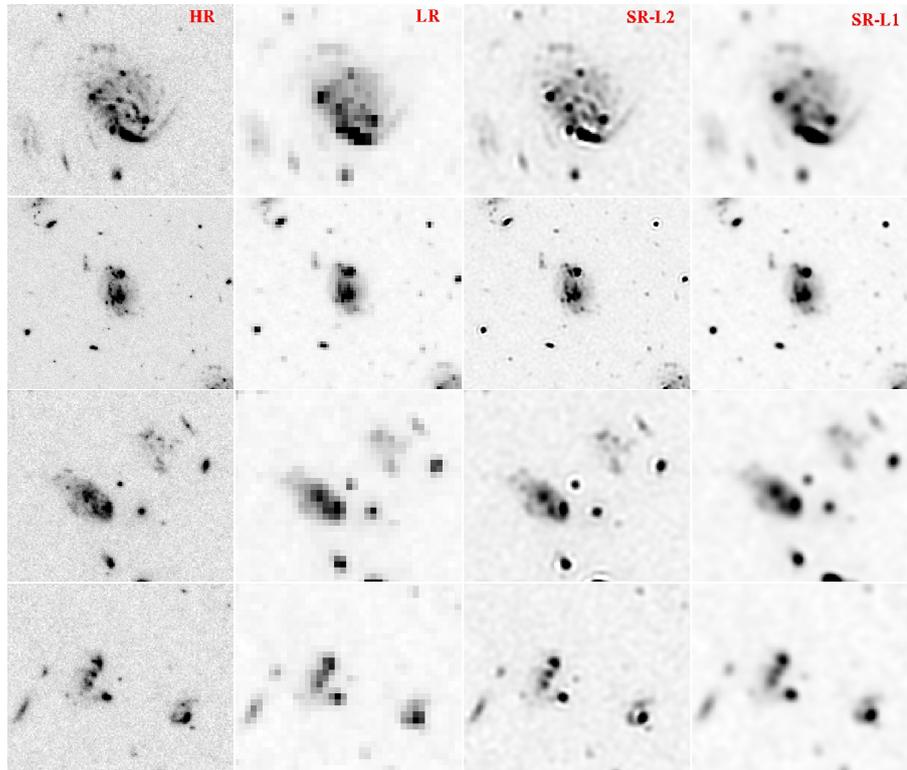


Figura 4.16: Da sinistra a destra: vari esempi di sorgenti nell'immagine  $1024 \times 1024$  HUDF originale, nei frame a bassa risoluzione e nelle immagini super-risolte.

*fettivamente a operare una deconvoluzione, almeno parziale, dell'immagine a bassa risoluzione. L'algoritmo SR-L1 non mostra risultati altrettanto soddisfacenti: statisticamente vi è una differenza di  $\sim 0.25$  magnitudini nel flusso degli oggetti rispetto al valore "vero", e l'effetto di deconvoluzione risulta meno marcato. Comunque l'immagine risulta complessivamente migliorata anche in questo caso, e l'alterazione nel flusso non sembra dipendere dall'intensità dello stesso: può dunque essere interpretata come una variazione dello "zero point", cioè del flusso di riferimento dell'immagine stessa e compensata a posteriori.*

#### 4.2.2 Super-risoluzione di osservazioni astronomiche della missione EUCLID

Nelle simulazioni illustrate nei paragrafi precedenti i frame a bassa risoluzione sono creati internamente a SuperResolve a partire dalla medesima immagine ad alta risoluzione: un possibile problema di questa procedura è dato dal fatto che il rumore, che domina le zone "vuote" delle immagini, non è indipendente tra i diversi frame utilizzati dal programma per ottenere l'immagine super-risolta. Di contro,

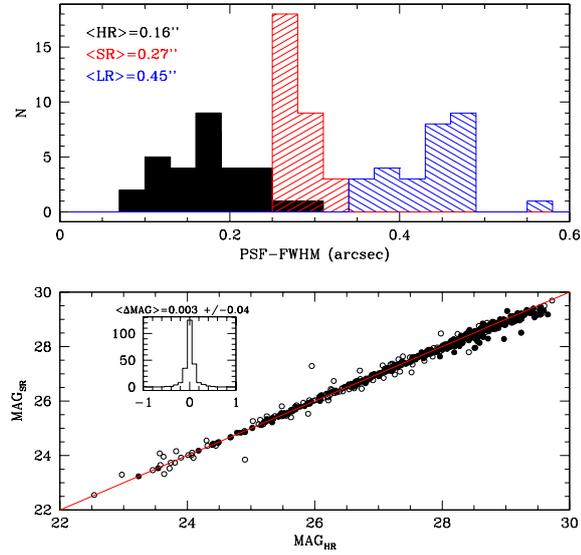


Figura 4.17: **Super-risoluzione SR-L2**. Pannello in basso: magnitudini misurate sull'immagine originale (ascisse) e sull'immagine super-risolta (ordinate), nell'inserito è riportato l'istogramma della differenza sull'intero campione. Pannello in alto: istogramma della FWHM delle stelle misurata sull'immagine originale (HR), sull'immagine super-risolta (SR) e su un frame a bassa risoluzione (LR), con i relativi valori medi.

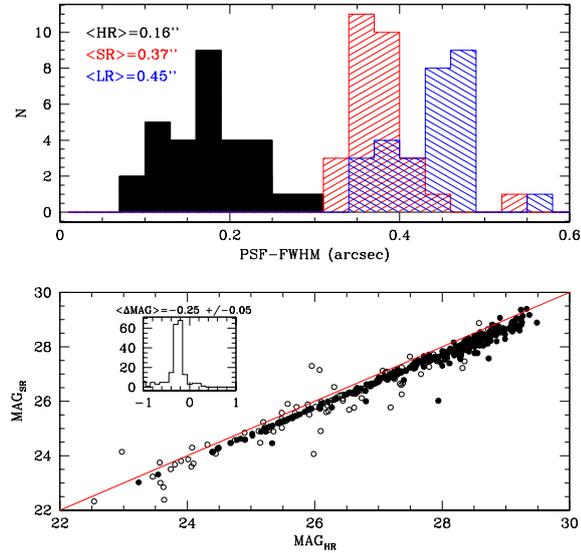


Figura 4.18: **Super-risoluzione SR-L1**. Pannelli come in Fig. 4.17

come descritto nel Cap. 2, le immagini astronomiche sono di norma ottenute sommando più osservazioni della stessa zona di cielo: in tal caso possiamo considerare le diverse immagini alterate da realizzazioni indipendenti della stessa distribuzione di rumore. Nell’ottica di procedere per gradi verso casi via via più realistici sperimentiamo ora gli algoritmi di super-risoluzione in un caso che, seppur in parte “semplificato”, fornisce un ottimo esempio di un vero dataset di imaging astronomico. Le immagini utilizzate per questo test sono accurate simulazioni delle osservazioni che verranno eseguite dal telescopio spaziale EUCLID<sup>2</sup>. Il progetto EUCLID, il cui lancio è previsto per il 2020, è la missione spaziale selezionata da ESA nel contesto “Cosmic Vision 2015-2025” con l’ambizioso obiettivo di migliorare di un fattore  $>100$  la nostra conoscenza dei parametri cosmologici: il telescopio spaziale osserverà oltre metà del cielo per determinare la dinamica dell’espansione cosmica tramite misure di precisione della distanza, del clustering e degli effetti di lente gravitazionale di milioni di galassie sino a quasi 10 miliardi di anni luce da noi [e.g. “Euclid Definition Study Report”, 21].

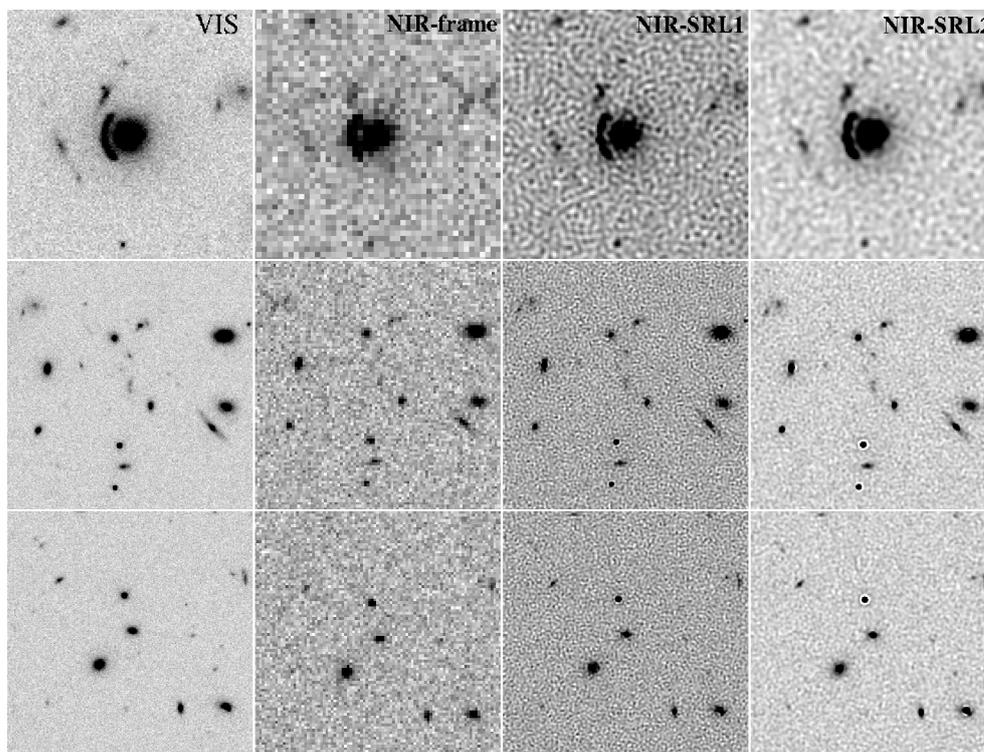


Figura 4.19: Esempi di sorgenti, da sinistra a destra: nell’immagine  $900 \times 900$  EUCLID-VIS, in uno dei frame NIR a risoluzione tre volte inferiore, e nelle immagini super-risolve.

EUCLID sarà dotato di due camere CCD, denominate “VIS” e “NIR”, per

<sup>2</sup><http://sci.esa.int/euclid>

compiere osservazioni rispettivamente nel range spettrale ottico ( $\sim 0.5\text{-}0.9 \mu\text{m}$ ) e infrarosso ( $\sim 0.9\text{-}2.0 \mu\text{m}$ ): le due camere avranno caratteristiche differenti in quanto a risoluzione, tanto in termini di dimensione dei pixel (0.1 arcsec per VIS, 0.3 arcsec per NIR) che in termini di PSF-FWHM : 0.2 arcsec per VIS, 0.3 arcsec, dunque estremamente sottocampionata, per NIR. Le immagini finali NIR saranno costruite come somma di più frame indipendenti e leggermente shiftati tra loro: sembra dunque molto promettente la possibilità di usare tecniche di super-risoluzione sui frame NIR per costruire immagini con risoluzione confrontabile a quella ottenuta da VIS, rendendo così possibili analisi più accurate della morfologia e del flusso luminoso.

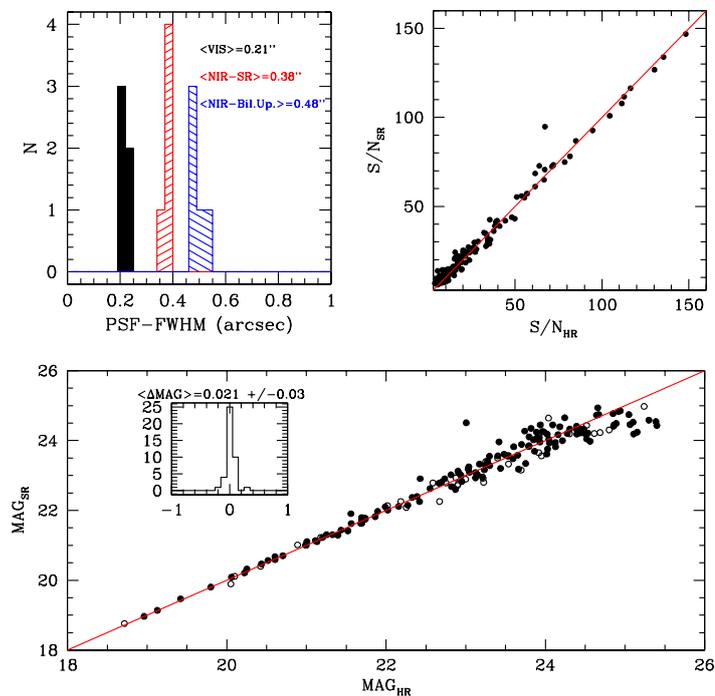


Figura 4.20: **Super-risoluzione SR-L2**. Pannello in basso: magnitudini misurate sull'immagine VIS (ascisse) e sull'immagine NIR super-risolta (ordinate), nell'inserito è riportato l'istogramma della differenza sull'intero campione. Pannello in alto a sinistra: istogramma della FWHM delle stelle misurata sull'immagine VIS, sull'immagine NIR super-risolta (SR) e su un upsampling bilineare di un fattore 3 della somma dei frame NIR, con i relativi valori medi. Pannello in alto a destra: S/N delle sorgenti sull'immagine VIS (ascisse) e sull'immagine NIR super-risolta (ordinate)

Utilizziamo il software Mksimul appositamente concepito per simulare immagini EUCLID [28]: i frame VIS e NIR vengono creati rebinando, aggiungendo

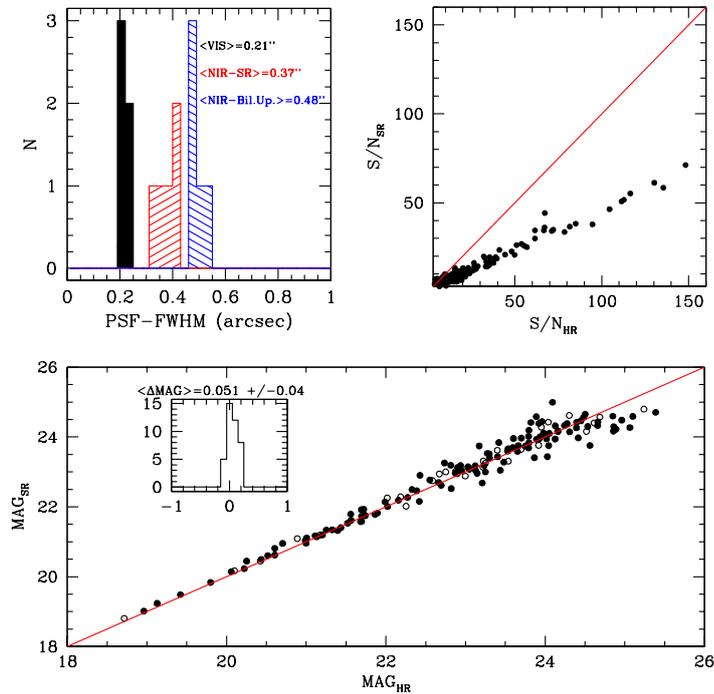


Figura 4.21: **Super-risoluzione SR-L1**. Pannelli come in Fig. 4.20

rumore Gaussiano e convolvendo con appositi kernel osservazioni Hubble Space Telescope, analoghe all’HUDF discusso in precedenza, in modo da riprodurre il flusso limite e la risoluzione attesa delle osservazioni EUCLID. Le immagini EUCLID hanno un flusso limite  $\sim 100$  volte superiore delle immagini HST usate come “base” delle simulazioni, dunque la componente dominante del rumore (aggiunta da `Mksimul`) è indipendente nei diversi frame.

Applichiamo gli algoritmi di super-risoluzione a 9 frame NIR (pixel-scale  $0.3''$ , PSF-FWHM= $0.3''$ ) di dimensioni  $300 \times 300$  arbitrariamente shiftati tra loro, chiedendo in output un’immagine con risoluzione VIS (dimensioni  $900 \times 900$ , pixel-scale  $0.1''$ , PSF-FWHM= $0.2''$ ): a `SuperResolve` viene dunque passata con l’opzione `INPUT` un file ASCII con i filename dei frame NIR, e con l’opzione `PSF` il file contenente il relativo kernel di convoluzione VIS-NIR. Fissiamo il parametro di regolarizzazione a  $\lambda = 1$  e  $\lambda = 30$  rispettivamente per SR-L2 e SR-L1 in base ai valori ottimali trovati nei test HUDF.

In Fig. 4.19 sono mostrati alcuni esempi di oggetti come appaiono sull’immagine finale VIS, su uno dei frame NIR e sulle immagini super-risolve con i due algoritmi SR-L2 e SR-L1. Non solo il miglioramento della risoluzione appare evidente, ma *si nota come gli algoritmi di super-risoluzione abbiano operato un “denoising” delle immagini*: oggetti che sui singoli frame sono completamente im-

mersi nel rumore e appena distinguibili risultano molto più evidenti sulle immagini super-risolte.

Valutiamo quantitativamente l'effetto della super-risoluzione analizzandole con il software SExtractor, in modo analogo a quanto fatto per le simulazioni su immagini HUDF: in Fig. 4.20 e 4.21 è mostrato, rispettivamente per SR-L2 e SR-L1:

- il paragone tra magnitudini misurate sull'immagine VIS e su NIR super-risolta (pannello in basso).
- l'istogramma delle FWHM delle stelle (e dunque la PSF) nell'immagine VIS, nell'immagine NIR super-risolta e su un upsampling bilineare di un fattore 3 della somma dei frame NIR (pannello in alto a sinistra).
- il paragone tra rapporto S/N delle sorgenti misurato da SExtractor su VIS e NIR super-risolta (pannello in alto a destra).

L'analisi delle immagini rivela alcune proprietà interessanti:

1. *le magnitudini degli oggetti restano sostanzialmente inalterate* a seguito della procedura di super-risoluzione. Nel caso dell'algoritmo SR-L1 questo risultato è in contrasto con quanto precedentemente ottenuto sulle simulazioni HUDF (Par. 4.2.1).
2. il rapporto S/N delle sorgenti misurato sull'immagine SR-L2 è paragonabile a quello ottenuto sull'immagine di riferimento VIS e dunque pari a quanto aspettato sulla *somma* dei nove frame NIR usati<sup>3</sup>. Sull'immagine ottenuta con SR-L1 il rapporto S/N risulta invece inferiore a quello dell'immagine somma, seppur leggermente superiore a quello atteso da misure su singoli frame.
3. la PSF misurata su NIR super-risolta è *migliore* di quella misurata sull'immagine somma NIR rebinata (upsampling bilineare) di un fattore 3.

Quanto ottenuto con SR-L2 è piuttosto sorprendente: l'azione di *denoising* dovuta alla regolarizzazione produce un'immagine super-risolta con "profondità" analoga a quella dell'immagine ottenibile sommando i nove frame, cioè una riduzione di un fattore tre del rumore presente, tutto ciò senza alterare il flusso degli oggetti.

Per quanto riguarda SR-L1, l'effetto di *denoising* è comunque presente anche se in misura minore. Il fatto che il flusso degli oggetti resti inalterato, a differenza di quanto ottenuto su HUDF, merita sicuramente ulteriori approfondimenti. Allo stato attuale ci limitiamo a notare quanto dichiarato dagli stessi Unger et al. [36], cioè che la regolarizzazione L1 "preserva i bordi (*strong edges*) presenti sull'immagine diffondendo invece piccoli gradienti". Evidentemente, nel caso delle immagini EUCLID l'effetto della regolarizzazione preserva il flusso delle sorgenti e correttamente "identifica" il rumore, che è indipendente nei diversi frame, diffondendolo.

<sup>3</sup>nel creare i frame NIR con Mksimul abbiamo assunto per semplicità che la profondità finale della loro somma sia pari a quella dell'immagine VIS

In presenza di frame in cui il pattern di rumore è lo stesso, come nel caso delle simulazioni HUDF, la regolarizzazione porta invece ad alterare leggermente anche il flusso delle sorgenti.

L'effetto della super-risoluzione sulla PSF è pure di grande interesse. Come ricordato in precedenza la PSF sulle immagini NIR è fortemente sottocampionata (dimensioni del pixel pari alla FWHM della PSF): in casi simili è noto che la misura della PSF dall'analisi della distribuzione di luce delle stelle risulti sovrastimata, come dimostrato dall'analisi dell'immagine NIR ricampionata con upsampling bilineare (istogramma blu in Fig. 4.20 e 4.21). La PSF misurata sulle immagini super-risolve risulta invece migliore per effetto della deconvoluzione operata dagli algoritmi SR: un risultato di estremo interesse per diversi degli scopi scientifici delle osservazioni EUCLID in particolare, e in generale di qualunque osservazione astronomica.

### 4.3 Prestazioni software

Come discusso nel Par. 3.1 la super-risoluzione con regolarizzazione L2 è implementata sia con minimizzazione per massima discesa a passo fisso che con minimizzazione col metodo del gradiente coniugato: svolgendo con entrambi i metodi le simulazioni sin qui presentate abbiamo verificato che il risultato ottenuto è pressochè identico (in particolare per valori del parametro di regolarizzazione attorno al  $\lambda$  ottimale o maggiori). Da un punto di vista di prestazioni i due metodi di minimizzazione per SR-L2 e l'algoritmo primario-duale di minimizzazione di SR-L1 presentano però notevoli differenze. Per valutarne quantitativamente le prestazioni utilizziamo SuperResolve in modalità di simulazione (MODE=SIM) a partire da porzioni di diverse dimensioni dell'immagine HUDF: 240×240, 480×480, 960×960 e 1920×1920 pixel. Fissando  $\lambda = 1$  (SR-L2) e  $\lambda = 30$  (SR-L1) simuliamo super-risoluzione delle diverse immagini con fattori  $r=2,4,6,8$  a partire da, rispettivamente, 4, 16, 36 e 64 frame a risoluzione inferiore e utilizzando per la convoluzione un kernel Gaussiano di dimensioni 11×11. La macchina usata per i test è un Linux desktop con CPU Quad-core Intel Xeon E5630 (2.53 GHz) con 16Gb RAM, due livelli di memoria cache per ciascun core (256 Kb L1 8-way set associative, 1 Mb L2 4-way set associative), più Cache L3 (12 Mb) condivisa dai 4 core.

Nelle Fig. 4.22-4.24 è riportato il tempo di esecuzione ("elapsed time") del software su singola CPU misurato con il task Linux `time` per le immagini di diverse dimensioni in funzione del guadagno in risoluzione cercato. Le figure fanno rispettivamente riferimento a: SR-L2 minimizzazione col metodo del gradiente coniugato a passo ottimale, SR-L2 minimizzazione col metodo di massima discesa a passo fisso, algoritmo primario duale SR-L1. La super-risoluzione a bassi valori di  $r$  (e dunque ridotto numero di frame in input) risulta in ogni caso efficiente, con tempi di esecuzione dell'ordine di pochi minuti anche per riprodurre immagini con  $\sim 4 \cdot 10^6$  pixel: ad esempio per ricostruire un'immagine 1920×1920 a partire da

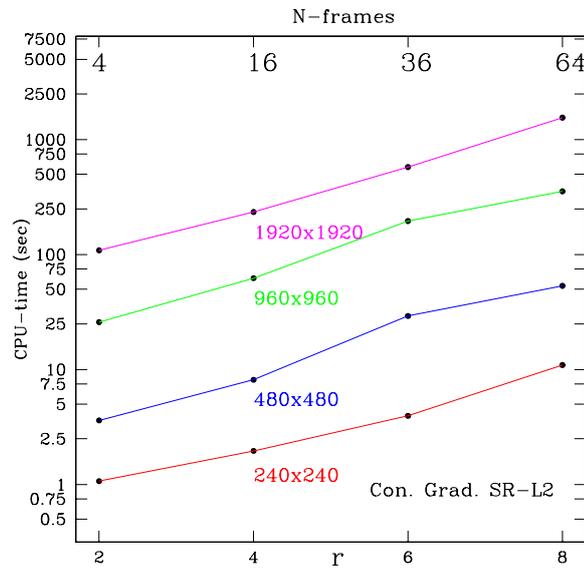


Figura 4.22: Tempi di esecuzione dell'algoritmo SR-L2, minimizzazione col metodo del gradiente coniugato a passo ottimale per immagini di diverse dimensioni in funzione del guadagno in risoluzione  $r$  usando  $r^2$  frame in input.

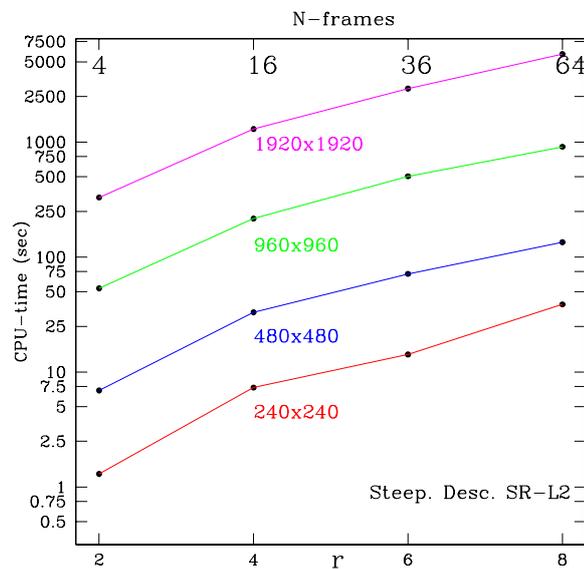


Figura 4.23: Come Fig. 4.22, algoritmo SR-L2, minimizzazione col metodo di massima discesa a passo fisso.

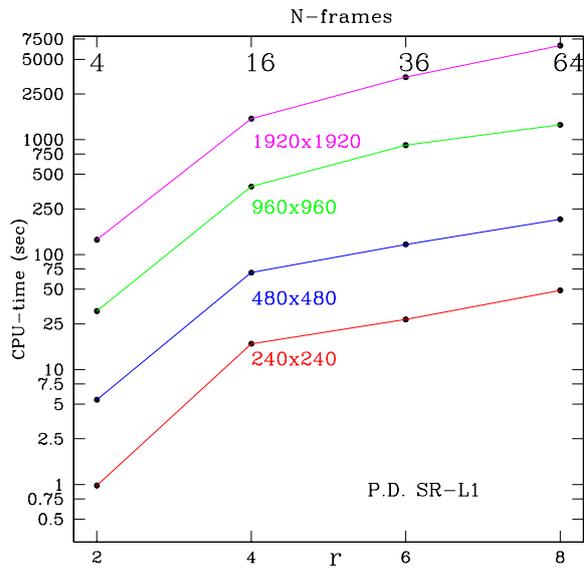


Figura 4.24: Come Fig. 4.22, algoritmo primario-duale con regolarizzazione L1 di Unger et al. [36].

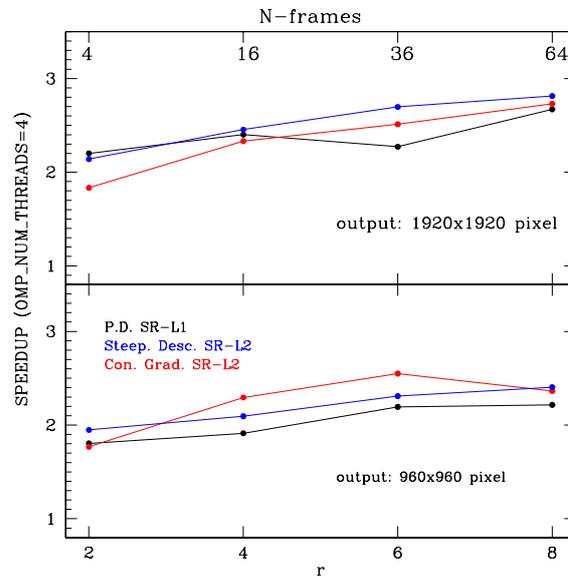


Figura 4.25: Rapporto tra tempo di esecuzione del codice su 1 core e tempo di esecuzione su 4 core per super-risoluzione di immagini 960x960 e 1920x1920 pixel.

4 frame in input servono tra  $\sim 100$  (SR-L2, gradiente coniugato) e i  $\sim 250$  secondi. A fissato guadagno di risoluzione i tempi di esecuzione crescono notevolmente passando da dimensioni di  $240 \times 240$  dell'immagine super-risolta ( $\sim 1$ sec per  $r=2$ , 10-50sec per  $r=8$ ) a immagini di  $1920 \times 1920$  (quasi due ore di CPU per SR-L1 e SR-L2 con massima discesa nel caso  $r=8$  usando 64 frame). Dei tre algoritmi il metodo del gradiente coniugato risulta notevolmente più efficiente con tempi di esecuzione massimo di  $\sim 2000$  secondi per super-risolvere l'immagine  $1920 \times 1920$  da 64 frame in input e tempi CPU inferiori al minuto per tutti i test discussi nel presente capitolo.

Le prestazioni del software migliorano sensibilmente sfruttando la parallelizzazione *shared memory*: lo *speedup*, rapporto tra tempo di esecuzione del codice seriale (1 CPU) e tempo di esecuzione su 4 CPU, è riportato in figura 4.25 per i casi di super-risoluzione di immagini  $960 \times 960$  e  $1920 \times 1920$  pixel con diversi fattori di guadagno  $r$ . I tre algoritmi implementati richiedono tempi di esecuzione da due a quasi tre volte inferiori sfruttando la parallelizzazione. Il maggiore guadagno in tempi di calcolo si ottiene proprio per i casi più onerosi di super-risoluzione da 36-64 frame dell'immagine  $1920 \times 1920$  pixel.

## Capitolo 5

# Conclusioni e Sviluppi Futuri

La presente tesi illustra il lavoro di stage svolto nel periodo Dicembre 2012 - Marzo 2013 presso l'Osservatorio Astronomico di Roma (INAF). Obiettivo del lavoro è stato sperimentare algoritmi di super-risoluzione (SR) su immagini astronomiche per valutare le potenzialità dell'applicazione di tale classe di tecniche di image-restoration su moderni dataset di imaging astrofisico. Una prima parte del lavoro è stata dedicata ad esplorare la varietà di tecniche SR proposte in letteratura (Cap. 1) e individuare possibili vantaggi e svantaggi del loro utilizzo alla luce delle specificità delle immagini astronomiche (Cap. 2). In seguito il lavoro è stato orientato all'implementazione software di due algoritmi appartenenti alla famiglia delle tecniche variazionali (Cap. 3): l'algoritmo discusso in Mitzel et al. [23] e Zomet and Peleg [41], con termine di regolarizzazione dato dalla norma L2 di  $\nabla x_H$  (che abbiamo denominato SR-L2), e l'algoritmo primario-duale di Unger et al. [36] basato sulla norma L1 di Huber (SR-L1). Infine (Cap. 4) sono stati svolti approfonditi test dei due algoritmi SR-L2 e SR-L1 e delle prestazioni del software tanto su immagini fotografiche che su immagini astrofisiche.

I risultati ottenuti possono essere riassunti nei seguenti punti:

- Creazione del software FORTRAN90-OpenMP SuperResolve (Par. 3.4) dotato delle caratteristiche necessarie per sperimentare tecniche SR su immagini astronomiche sia per il lavoro di stage che per sviluppi successivi:
  - 1) architettura modulare con uso di Subroutine e MODULE Fortran in modo da consentire future estensioni (nuovi algoritmi, schemi numerici, ecc);
  - 2) gestione del programma a run-time tramite un file ASCII di input (e da linea di comando) con un esteso numero di opzioni per consentire lo svolgimento di ogni genere di test e il *batch-processing*.
  - 3) possibilità di due modalità di uso: "simulazione" (immagine HR nota) con successiva esecuzione di test statistici sull'immagine super-risolta, e modalità applicativa (set di immagini 'reali', cioè frame HR ignoto).
  - 4) Uso di librerie FITSIO per la manipolazione/lettura/scrittura di immagini astronomiche in formato FITS e dei relativi header.
  - 4) Accorgimenti di ottimizzazione e paralleliz-

zazione per migliorare l'efficienza del codice. Le prestazioni misurate del codice (Par. 4.3) sono più che soddisfacenti: i tempi di esecuzione vanno da pochi secondi a 2-3 minuti per test con guadagno in risoluzione  $r \sim 2-4$  anche su immagini di  $\sim 2000 \times 2000$  pixel e per test con guadagno in risoluzione più elevato ( $r \sim 6-8$ ) su immagini sino a  $\sim 1000 \times 1000$  pixel. L'algoritmo di minimizzazione più efficiente (di un fattore  $\sim 2-3$ ) risulta essere SR-L2 con metodo del gradiente coniugato. L'utilizzo della parallelizzazione OpenMP porta a un miglioramento di un fattore  $\sim 3$  dei tempi di calcolo per i test più onerosi (immagini di milioni di pixel ricostruite a partire da 36-64 frame a bassa risoluzione) utilizzando quattro unità logiche di un processore quad-core.

- I test svolti su immagini fotografiche indicano che entrambi gli algoritmi sono decisamente più efficaci della semplice interpolazione nel ricostruire immagini a risoluzione maggiore, anche usando un numero limitato di frame. L'algoritmo SR-L1 risulta maggiormente efficiente su immagini fortemente contrastate (codice a barre) mentre SR-L2 appare più stabile in termini di scelta del parametro di regolarizzazione  $\lambda$  e usando un numero sub-ottimale ( $< r^2$ ) di frame a bassa risoluzione.
- I test su immagini astrofisiche si sono rivelati estremamente promettenti per l'introduzione di tecniche SR in astrofisica: il miglioramento della qualità dell'immagine, anche in termini di deconvoluzione della PSF, è evidente, e le immagini super-risolte sono decisamente migliori rispetto a quelle ottenibili da upsampling (comunemente utilizzati in campo astronomico). In ogni test effettuato l'algoritmo SR-L2 lascia inalterati i valori in flusso delle sorgenti e produce i risultati migliori in termini di aumento della risoluzione deconvolvendo l'effetto della PSF. Nei test su simulazioni da immagini HUDF SR-L1 produce una leggera alterazione sistematica del flusso delle sorgenti. Tale inconveniente non si presenta nei test su dataset EUCLID. La questione merita sicuramente ulteriori approfondimenti, allo stato attuale si può ipotizzare che il problema potrebbe essere originato dalla correlazione del pattern di rumore dei frame a bassa risoluzione HUDF (tutti ottenuti dalla medesima immagine). Entrambi gli algoritmi producono un denoising dell'immagine super-risolta: SR-L2 produce risultati estremamente interessanti a riguardo, in quanto la riduzione del rumore nell'immagine finale è paragonabile a quanto atteso dalla somma delle  $N$  immagini con rumore indipendente ( $S/N \propto \sqrt{N}$ ).

Alla luce dei test effettuati possiamo dire che le tecniche SR sono di sicuro interesse per l'analisi di immagini astronomiche. Pur essendovi necessità di ulteriori test e approfondimenti, i risultati ottenuti costituiscono una rilevante novità, ed è sicuramente da prendere in considerazione l'opportunità di sottoporre un'analisi dell'uso di tecniche SR su dataset astrofisici a riviste peer-reviewed del settore.

## 5.1 Sviluppi futuri

Il lavoro di sperimentazione sin qui svolto merita ulteriori approfondimenti, in particolare seguendo le seguenti linee di sviluppo:

1. Per come è concepito, il software SuperResolve si presta all'introduzione tra le sue opzioni di nuovi algoritmi, di tipo variazionale con differenti metodi di regolarizzazione e non solo. Altre tecniche SR tra quelle illustrate nel Cap. 1 possono essere implementate come subroutine Fortran e chiamate, a costo di modifiche minori, dal main file, sfruttando il codice già esistente per l'esecuzione di simulazioni, il calcolo di test statistici e la gestione dell'I/O. Il codice può essere ulteriormente migliorato, ad esempio combinando la parallelizzazione OpenMP già presente con comandi di parallelizzazione MPI o CUDA-FORTRAN per effettuare le operazioni su immagini con tecniche di domain decomposition.
2. La scelta di un valore ottimale per il parametro di regolarizzazione  $\lambda$  è stata sin qui effettuata prendendo come riferimento i risultati di simulazioni con immagine ad alta risoluzione nota. Una linea di sviluppo da prendere in considerazione è l'implementazione di tecniche per la scelta di  $\lambda$  su set di immagini reali in cui l'immagine HR non è nota: ad esempio il metodo della Generalized Cross Validation [12], o il criterio della "L-curve" [13, 14].
3. I test astrofisici sin qui effettuati sono stati svolti su casi realistici ancorchè semplificati. Ulteriori test su casi più complessi possono essere svolti allo scopo di: a) individuare il genere di immagini che maggiormente si presta all'uso di tecniche SR; b) comprendere gli effetti sulla SR di incertezze nella calibrazione astrometrica e di presenza di difetti (background, pixel difettosi); c) esplorare casi di variazione spaziale della PSF e di presenza di distorsioni e aberrazioni ottiche; d) individuare il modo migliore di inserire l'uso della SR nel processo di calibrazione e riduzione delle osservazioni astronomiche (Par. 2.1.1).



# Bibliografia

- [1] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer, 2006.
- [2] Vivek Bannore. Iterative-interpolation super-resolution (iisr). *Iterative-Interpolation Super-Resolution Image Reconstruction*, pages 19–50, 2009.
- [3] S. V. W. Beckwith, M. Stiavelli, A. M. Koekemoer, J. A. R. Caldwell, H. C. Ferguson, R. Hook, R. A. Lucas, L. E. Bergeron, M. Corbin, S. Joglee, N. Panagia, M. Robberto, P. Royle, R. S. Somerville, and M. Sosey. The Hubble Ultra Deep Field. *AJ*, 132:1729–1755, November 2006. doi: 10.1086/507302.
- [4] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *A&AS*, 117:393–404, June 1996.
- [5] E. Bertin, Y. Mellier, M. Radovich, G. Missonnier, P. Didelon, and B. Morin. The TERAPIX Pipeline. In D. A. Bohlender, D. Durand, and T. H. Handley, editors, *Astronomical Data Analysis Software and Systems XI*, volume 281 of *Astronomical Society of the Pacific Conference Series*, page 228, 2002.
- [6] M. Castellano, A. Fontana, D. Paris, A. Grazian, L. Pentericci, K. Boutsia, P. Santini, V. Testa, M. Dickinson, M. Giavalisco, R. Bouwens, J.-G. Cuby, F. Mannucci, B. Clément, S. Cristiani, F. Fiore, S. Gallozzi, E. Giallongo, R. Maiolino, N. Menci, A. Moorwood, M. Nonino, A. Renzini, P. Rosati, S. Salimbeni, and E. Vanzella. The bright end of the  $z \sim 7$  UV luminosity function from a wide and deep HAWK-I survey. *A&A*, 524:A28, December 2010. doi: 10.1051/0004-6361/201015195.
- [7] Subhasis Chaudhuri. *Super-resolution imaging*, volume 632. Springer, 2001.
- [8] Dmitry Datsenko and Michael Elad. Example-based single document image super-resolution: a global map approach with outlier rejection. *Multidimensional Systems and Signal Processing*, 18(2):103–121, 2007.
- [9] Michael Elad and Arie Feuer. Super-resolution reconstruction of image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):817–834, 1999.

- [10] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.
- [11] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 349–356. IEEE, 2009.
- [12] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- [13] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, 34(4):561–580, 1992.
- [14] Per Christian Hansen. *The L-curve and its use in the numerical treatment of inverse problems*. 1999.
- [15] Russell C Hardie, Kenneth J Barnard, John G Bognar, Ernest E Armstrong, and Edward A Watson. High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. *Optical Engineering*, 37(1):247–260, 1998.
- [16] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods*, 6(9):813–827, 1977.
- [17] Peter J Huber. Robust statistics. 1981, 1981.
- [18] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.
- [19] Donald E Knuth. The Art of Computer Programming, Vol. 3: Sorting and Searching. *Reading, MA: Addison-Wesley*, 19:496–503, 1973.
- [20] R. G. Kron. Photometry of a complete sample of faint galaxies. *ApJS*, 43: 305–325, June 1980. doi: 10.1086/190669.
- [21] R. Laureijs, J. Amiaux, S. Arduini, J. . Auguères, J. Brinchmann, R. Cole, M. Cropper, C. Dabin, L. Duvet, A. Ealet, and et al. Euclid Definition Study Report. *ArXiv e-prints*, October 2011.
- [22] Peyman Milanfar. *Super-resolution imaging*, volume 1. CRC Press, 2010.
- [23] Dennis Mitzel, Thomas Pock, Thomas Schoenemann, and Daniel Cremers. Video super resolution using duality based TV-L 1 optical flow. *Pattern Recognition*, pages 432–441, 2009.

- 
- [24] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer Science+ Business Media, 2006.
- [25] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.
- [26] Andrew J Patti, M Ibrahim Sezan, and A Murat Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *Image Processing, IEEE Transactions on*, 6(8):1064–1076, 1997.
- [27] Lyndsey C Pickup, David P Capel, Stephen J Roberts, and Andrew Zisserman. Bayesian methods for image super-resolution. *The Computer Journal*, 52(1):101–113, 2009.
- [28] S. Pilo, D. Paris, A. Fontana, and M. Castellano. A simulation tool based on deep HST mosaics for EUCLID OU-MER. *EUCLID OU-MER internal document*, 2012.
- [29] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1133–1140. IEEE, 2009.
- [30] WH Press, SA Teukolsky, WT Vetterling, BP Flannery, and M Metcalf. Numerical recipes in Fortran 90, vol. 2. *Cambridge, ISBN*, 521574390, 1996.
- [31] R Tyrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.
- [32] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- [33] Henry Stark and Peyma Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *JOSA A*, 6(11):1715–1726, 1989.
- [34] AN Tikhonov and V Ya Arsenin. Solutions of ill-posed problems. 1977. *WH Winston, Washington, DC*, 1977.
- [35] RY Tsai and Thomas S Huang. Multiframe image restoration and registration. *Advances in computer vision and Image Processing*, 1(2):317–339, 1984.
- [36] Markus Unger, Thomas Pock, Manuel Werlberger, and Horst Bischof. A convex approach for variational super-resolution. *Pattern Recognition*, pages 313–322, 2010.
- [37] Zhou Wang, Alan C Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

- 
- [38] Jianchao Yang and Thomas Huang. Image super-resolution: Historical overview and future challenges. *Super-resolution imaging*, 2010.
- [39] Dante C Youla and Heywood Webb. Image restoration by the method of convex projections: Part 1 theory. *Medical Imaging, IEEE Transactions on*, 1(2):81–94, 1982.
- [40] Assaf Zomet and Shmuel Peleg. Efficient super-resolution and applications to mosaics. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 579–583. IEEE, 2000.
- [41] Assaf Zomet and Shmuel Peleg. Super-resolution from multiple images having arbitrary mutual motion. *Super-Resolution Imaging*, pages 195–209, 2002.