

**ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA**

SCUOLA DI INGEGNERIA E ARCHITETTURA

Sede Forlì

CORSO DI LAUREA IN INGEGNERIA AEROSPAZIALE

Classe L-9

ELABORATO FINALE DI LAUREA

In Impianti Aerospaziali

**STUDIO PRELIMINARE DI UN ALGORITMO  
PER LA DETERMINAZIONE ORBITALE  
ATTRAVERSO IMMAGINI DI CORPI CELESTI**

CANDIDATO

Federico De Vercelli

RELATORE

Dr. Ing. Modenini Dario

CORRELATORE

Dr. Ing. Zannoni Marco

Anno Accademico 2016-2017

Sessione III

# INDICE

<b>0. INTRODUZIONE</b> .....	5
<b>1. ASTROMETRIA E NAVIGAZIONE OTTICA</b> .....	6
1.1 ORIGINI DELL'ASTROMETRIA .....	6
1.2 NAVIGAZIONE .....	8
1.3 OPTICAL NAVIGATION .....	9
<b>2. IMMAGINI DIGITALI</b> .....	11
2.1 ACQUISIZIONE E ELABORAZIONE DELLE IMMAGINI DA PARTE DELL'UOMO .....	11
2.2 LA LUCE .....	12
2.3 Acquisizione ed elaborazione delle       immagini digitali .....	13
2.3.1 ACQUISIZIONE .....	13
2.3.2 L'IMMAGINE DIGITALE COME FUNZIONE BIDIMENSIONALE .....	15
<b>3. IMAGE PROCESSING</b> .....	17
3.1 L'AMBIENTE DI CALCOLO .....	17
3.2 RILEVAMENTO DEI BORDI IN UN'IMMAGINE: EDGE DETECTION .....	18
3.2.1 FILTRI .....	19
3.2.2 FILTRO GAUSSIANO .....	21
3.2.3 DERIVATA DI UN'IMMAGINE .....	23
3.2.4 GRADIENTE DI UN'IMMAGINE .....	25
3.3 METODO CANNY .....	26
3.3.1 RIDUZIONE DEL RUMORE .....	26
3.3.2 GRADIENTE .....	27
3.3.3 SOPPRESSIONE DEI NON-MASSIMI .....	28
3.3.4 EDGE DETECTION TRAMITE ISTERESI .....	29
<b>4. ELLIPSE FITTING</b> .....	30
4.1 ELLISSE .....	30
4.2 RELAZIONE TRA CORPI CELESTI ED ELLISSI .....	33

4.2.1 PINHOLE CAMERA MODEL .....	33
4.2.2 ELLISSOIDE DI RIFERIMENTO TRIASSIALE .....	37
4.3 RELAZIONE TRA ELLISSOIDE TRIASSIALE E LA SUA PROIEZIONE NEL PIANO DELL'IMMAGINE .....	38
4.4 METODI DI REGRESSIONE AD ELLISSE .....	43
4.4.1 MINIMI QUADRATI FORMULAZIONE GENERALE .....	43
4.4.2 DETERMINAZIONE DELL'ELLISSE CHE MINIMIZZA LA DISTANZA ALGEBRICA.....	46
4.4.3 DETERMINAZIONE DELL'ELLISSE DI APPROSSIMAZIONE IN FORMA PARAMETRICA.....	48
4.5 ALGELLIPSE.....	50
4.6 BOOKSTEIN .....	51
4.7 ELLIPSE DIRECT FIT.....	51
4.8 ELLIPSE FIT BY TAUBIN .....	52
4.9 FIT_ELLIPSE .....	52
4.10 FITELLIPSE.....	53
<b>5. IL PROGETTO.....</b>	<b>54</b>
5.1 CALIBRAZIONE IMMAGINI DELLA SONDA CASSINI.....	54
5.2 DATABASE .....	55
5.3 LETTURA IMMAGINE ED EDGE DETECTION .....	55
5.4 ERRORE E ERRORE MEDIO.....	57
5.5 ALGELLIPSE.....	59
5.5.1 FITTING .....	60
5.6 BOOKSTEIN .....	61
5.6.1 FITTING.....	61
5.6.2 CENTROIDI.....	62
5.7 ELLIPSE DIRECT FIT.....	62
5.7.1 FITTING.....	62

5.7.2 CENTROIDI .....	63
5.8 ELLIPSE FIT by TAUBIN .....	64
5.8.1 FITTING .....	64
5.8.2 CENTRODI .....	65
5.9 FIT_ELLIPSE .....	65
5.9.1 REGRESSIONE AD ELLISSE .....	66
5.9.2 CENTROIDE .....	66
5.10 FIT ELLIPSE .....	67
5.10.1 FITTING .....	67
5.10.2 CENTROIDE .....	68
5.11 CONCLUSIONI CAPITOLO .....	68
<b>6. RISULTATI OTTENUTI DALLE FUNZIONI</b> .....	<b>69</b>
6.1 TABELLA CENTROIDI .....	69
6.2 ERRORE MEDIO (Misurato in PIXEL) .....	71
6.3 IMMAGINI ELABORATE TRAMITE ELLIPSE FITTING .....	74
<b>7.0 CONCLUSIONI</b> .....	<b>80</b>
RINGRAZIAMENTI .....	82
BIBLIOGRAFIA .....	83
FONTE IMMAGINI .....	84
SITOGRAFIA .....	85

## 0. INTRODUZIONE

In questa tesi verrà trattata la navigazione ottica usando i centroidi dell'ellissi risultanti dalle immagini di pianeti e/o satelliti planetari, tale tecnica sta alla base della determinazione orbitale attraverso immagini di corpi celesti. Quest'ultima si è dimostrata di grande utilità per l'orientamento e per le manovre di avvicinamento al corpo in studio, da parte dei satelliti artificiali (o sonde), così da poter adempiere ai compiti della missione.

L'obiettivo principale che questa tesi chiedeva era la compilazione di un algoritmo preliminare, che fosse in grado di effettuare un rilevamento dei bordi da delle immagini relative a corpi celesti (in questo caso sono state utilizzate quelle scattate dalla sonda Cassini<sup>1</sup>) e successivamente di approssimare il limbo individuato con un'ellisse tramite l'ausilio di metodi differenti per ricavarne infine il centroide. Da qui, sono state prese le coordinate del centro della conica approssimante relative a tutte le immagini utilizzate per testare l'algoritmo (cinquantuno) ottenute tramite i vari metodi presenti in esso e confrontate con quelle elencate nell'articolo scientifico [1].

Il tipo di rilevamento dei bordi usato nel programma creato per questa tesi è il metodo Canny, questo perché si è rivelato il più preciso e quello con minore presenza di disturbi all'interno dell'immagine.

Successivamente sono state confrontate le varie funzioni inserite all'interno dell'algoritmo, le quali sono principalmente suddivise in due macro categorie: una è basata sulla risoluzione di un problema di tipo lineare dei minimi quadrati (minimizzazione della distanza algebrica), mentre l'altra sulla risoluzione non lineare del problema dei minimi quadrati (minimizzazione della distanza geometrica). Questo confronto serve per poter individuare quale categoria approssimi al meglio il limbo del corpo celeste e quale è più precisa nel calcolo del centroide, questo anche in base al tipo d'immagine.

---

<sup>1</sup> Prese dal sito: [A]

# 1. ASTROMETRIA E NAVIGAZIONE OTTICA

## 1.1 ORIGINI DELL'ASTROMETRIA

L'astrometrica è uno dei settori più antichi e importanti dell'astronomia, si occupa di comprendere e conoscere le posizioni, le distanze e i movimenti delle stelle e altri corpi celesti, come ad esempio i satelliti planetari o asteroidi. Questa "disciplina" risale ai tempi d'Ipparco, famoso astronomo greco, il quale fu il primo a stilare una lista di stelle a lui visibili grazie al sistema, ancora oggi in uso, della "scala di luminosità magnitudine apparente". Per magnitudine apparente si intende la misura della luminosità di un corpo celeste, rilevabile da un punto d'osservazione, esempio la Terra, il cui valore di magnitudine risulta corretto considerando la luminosità che il corpo celeste avrebbe se venisse osservato senza l'atmosfera terrestre.

Successivamente, un altro importantissimo astrologo, il polacco Niccolò Copernico, elaborò la teoria della "Rivoluzione Copernicana", che cambiò completamente il modo di vedere e studiare l'Universo. Copernico, infatti, servendosi della misurazione precisa dei pianeti, derivanti dai principi dell'astrometria, individuata dall'astronomo danese Tycho Brahe, introdusse per la prima volta la teoria Eliocentrica che si contrapponeva a quella Geocentrica, dove non era più la terra al centro del sistema delle orbite planetarie (Geocentrica) come si era pensato fino ad ora, ma è il Sole.

Spostandoci più avanti nel tempo si incontra uno dei più importanti pionieri della astrometrica moderna, Friedrich Bessel, matematico astronomo tedesco vissuto a fine dell'ottocento, il quale fu il primo ad applicare la parallasse per misurare la distanza di una stella. Il metodo della parallasse è il fenomeno per cui un oggetto sembra spostarsi rispetto allo sfondo cambiando punto d'osservazione, a livello quantitativo la

parallasse rappresenta l'angolo di spostamento. Questo metodo<sup>2</sup> è utilizzato in astronomia per determinare la distanza tra due corpi celesti non troppo lontani da loro (esempio Parallasse Lunare).

Un importante esempio dello sviluppo dell'astrometrica moderna, è stato introdotto nei primi del novecento dal famoso astrofisico americano Edwin Hubble, il quale fece uso del metodo della parallasse sulle “variabili Cefeidi” per misurare la distanza delle nebulose<sup>3</sup> e fu in grado di correlare il loro periodo di variazione con la loro luminosità. Le deduzioni principali derivano dall'applicazione della parallasse, che ha permesso la scoperta di galassie esterne alla nostra e la formulazione della Legge di Hubble<sup>4</sup>, la quale afferma che esiste una relazione lineare tra lo spostamento verso il rosso della luce emessa dalle galassie e la loro distanza, più la distanza è alta più sarà grande il suo spostamento verso il rosso, matematicamente:

$$z = \frac{H_0 D}{c} \quad (1)$$

Dove  $z$  è lo spostamento verso il rosso misurato dalla galassia,  $D$  la sua distanza,  $c$  è la velocità della luce e  $H_0$  è la costante di Hubble.

A fine del novecento, tra il 1989 e 1993, l'astrometria fece il suo ingresso nella tecnologia spaziale, dove per la prima volta l'ESA<sup>5</sup> compì una missione chiamata appunto “Missione d'astrometria spaziale: Hipparcos<sup>6</sup>”, la quale si basava sulla misura delle parallassi stellari e questo consentì poi di ricavare la distanza di una stella e il loro moto proprio. La missione sostanzialmente era divisa in due parti, la prima aveva l'obiettivo di misurare i parametri astrometrici di circa 120.000 stelle con alta precisione e la seconda era di misurare le proprietà

---

<sup>2</sup> Misurando l'angolo della parallasse e la distanza dei due punti d'osservazione, grazie alla trigonometria è possibile calcolare la distanza dall'oggetto

<sup>3</sup> Prima di Hubble per nebulose si intendeva un oggetto astronomico di grandi dimensioni di natura non stellare, planetaria e cometaria. Hubble fu in grado di provare che le macchie di luce in lontananza erano miriadi di stelle, cioè Galassie e non le nebulose che conosciamo oggi come agglomerati interstellari di idrogeno, polveri e plasma, che possono anche essere i resti di stelle o supernove.

<sup>4</sup> La legge di Hubble è alla base della teoria dello “Spostamento dell'Universo”. Questa legge si basa sull'effetto doppler, il quale per una sorgente che si allontana le righe spettrali delle onde hanno uno spostamento verso il rosso e verso il blu se si avvicina, con ovvie variazioni di lunghezza d'onda e frequenza. Grazie a questo è stato possibile fare una stima dell'età dell'Universo che è di circa 13.7 miliardi

<sup>5</sup> Agenzia Spaziale Europea

<sup>6</sup> In onore di Ipparco di Nicea

astrometriche e di fotometria in due colori di 400.000 stelle. Questo premise la creazione del catalogo Hipparcos, contenente più di un milione di stelle con posizioni accurate e come ultimo risultato fu possibile elaborare “l’Atlante Stellare del Millennio”.

Un altro esempio d’applicazione di astrometrica moderna è tutt’ora in corso tramite il Satellite Gaia, che ha lo scopo di continuare la missione Hipparcos con la compilazione di una lista di stelle e delle loro posizioni. Lo *spacecraft imaging* è diventato sempre più parte dell’astrometria moderna e le immagini ottenute dalle missioni spaziali vengono spesso utilizzate per misurare le posizioni astrometriche e i movimenti orbitali dei satelliti planetari, per esempio tramite l’*astrometric reduction* è stato possibile conoscere il comportamento Phobos e Deimos (satelliti marziani) grazie alle immagini prese da Mars Express<sup>7</sup>. Grazie alla sonda Cassini inoltre è stato possibile conoscere le proprietà astrometriche dei satelliti di Saturno, Mimas e Enceladus, le quali saranno prese d’esempio in questa tesi, trattando le loro immagini.

## 1.2 NAVIGAZIONE

Con l’aumentare dell’importanza degli obiettivi da parte delle Agenzie Spaziali, le quali puntano a spedire navicelle con equipaggio su Marte e sulla Luna, è cresciuta anche la necessità di aggiornare il sistema di navigazione autonoma, per far sì che l’equipaggio in caso di problemi di comunicazione possa avvalersi di un metodo che gli consenta il ritorno sulla Terra. I metodi utilizzati fino ad oggi si basano principalmente su una combinazione di misure radiometriche e ottiche (OPNAV) e sul “ground-based tracking”, i quali possono andare in failure in caso di problemi di comunicazione con la Terra, perché i dati che vengono inviati dalle stazioni di terra potrebbero non raggiungere le navicelle e di conseguenza il sistema di ricezione non sarebbe in grado di scaricarli, lasciando disorientato l’equipaggio.

---

<sup>7</sup> La Mars Express è una sonda dell’Agenzia Spaziale Europea lanciata nello spazio per studiare Marte nel giugno 2003.



## 1.3 OPTICAL NAVIGATION

L'idea che sta dietro al nuovo sistema di navigazione autonoma, prende il nome di *Optical Navigation Using Planet's Centroid* (OPNAV) o *Astrometric Reduction*, a seconda che si voglia determinare la posizione del satellite nota quella del pianeta o viceversa, la quale si basa sulle immagini che la telecamera/fotocamera cattura durante il suo viaggio interplanetario, dove i soggetti rappresentati sono dei corpi celesti. Molte di queste foto in apparenza sono buie oppure si nota solo un limbo dell'oggetto in questione, da qui tramite l'elaborazione dell'immagine avvalendosi di funzioni di identificazione del limbo (*egde detection*) è possibile individuare anche nelle foto buie il bordo del corpo celeste eliminando il rumore dell'immagine. Dopodiché usando dei metodi *d'ellipse fitting* il limbo visibile diventa parte d'ellisse che approssima la forma del pianeta, e di quest'ultima ne viene calcolato il centroide, che sarà poi utilizzato dalla navicella o sonda per capire la sua posizione e orientamento se si conosce la posizione del corpo celeste, viceversa se si conosce la posizione della sonda è possibile calcolare la posizione del Satelliti o Pianeta sconosciuto.

OPNAV si è dimostrata di grande utilità per l'orientamento e per le manovre d'avvicinamento al corpo in studio, da parte dei satelliti artificiali, sonde e cruiser per portare a termine i compiti della missione. Viene utilizzato questo tipo di navigazione autonoma soprattutto quando si devono effettuare missioni di tipo interplanetario come ad esempio la missione Cassini-Huygens. Questa missione iniziata ufficialmente nel 1997, aveva il compito di studiare l'intero sistema<sup>8</sup> di Saturno, per farlo è stata creata una sonda composta principalmente da due elementi: Cassini<sup>9</sup>, l'*orbiter* e Huygens<sup>10</sup>, il *lander*. Quest'ultimo si occupò, circa un anno dopo l'entrata in orbita della sonda, anno 2005, di studiare l'atmosfera, i rumori e di raccogliere immagini di Titano, questo durante la sua discesa

---

<sup>8</sup> Per intero sistema si intendono sia le sue lune che i suoi anelli, oltre che il pianeta stesso.

<sup>9</sup> Prende il nome dal famoso astronomo italiano della fine del Seicento che ebbe una rilevante importanza sullo studio di Saturno e i suoi anelli; costruttore NASA.

<sup>10</sup> Prende il nome dal famoso astronomo olandese dell'inizio del seicento che scoprì, utilizzando il telescopio, Titano, luna saturniana; costruttore ESA.

durata circa due ore e mezza, prima che il segnale sparì definitivamente. Mentre l'*orbiter* ha avuto molti compiti nei suoi vent'anni di vita (la quale finì con la sua distruzione programmata, nell'atmosfera saturniana il 15 settembre 2017), uno dei quali è stato collezionare numerose immagini di Saturno, Mimas e Encelados. Le immagini di queste lune, prese dal sito PDS IMAGING<sup>11</sup>, verranno ampiamente trattate in questa tesi, perché utilizzate come immagini campione per testare l'algoritmo Matlab creato per lo studio preliminare della determinazione orbitale attraverso immagini di corpi celesti, i cui risultati sono stati comparati con quelli presenti nell'articolo [1].

---

<sup>11</sup> <https://pds-imaging.jpl.nasa.gov/volumes/iss.html>

## 2. IMMAGINI DIGITALI

### 2.1 ACQUISIZIONE E ELABORAZIONE DELLE IMMAGINI DA PARTE DELL'UOMO

Per poter capire come visualizzare e elaborare le immagini digitali tramite calcolatore sarebbe meglio capire come l'uomo acquisisce ed elabora le immagini dell'ambiente circostante. Noi tutti abbiamo un ottimo sistema visivo e di processamento delle immagini, il fulcro è il collegamento che c'è tra gli occhi e il cervello.

L'occhio è composto da varie parti, ed ha il compito di ricavare le informazioni esterne attraverso la luce, la cui intensità viene regolata attraverso l'iride<sup>12</sup> che funziona come un diaframma<sup>13</sup>, questa viene focalizzata attraverso un sistema di lenti regolabili, chiamato cristallino, il quale restituisce un'immagine sulla retina<sup>14</sup>, che a sua volta trasforma l'immagine in una serie di segnali elettrici e attraverso il nervo ottico vengono inviati al cervello per l'elaborazione e l'interpretazione .

I muscoli oculari vengono impiegati principalmente per poter modificare la forma del cristallino, cioè modulare la messa a fuoco dell'immagine sulla retina a seconda della distanza dell'oggetto e successivamente viene trasferita al cervello. L'occhio può distinguere circa  $10^{10}$  livelli d'intensità luminosa, quella percepita risulta differente da quella reale poiché non dipende solo dalla luminosità, ma anche dallo sfondo sul quale il soggetto si trova. Un esempio di questo fenomeno può essere esplicitato prendendo un rettangolo o una qualsiasi figura geometrica di una certa tonalità di grigio e messo su tre sfondi con tonalità differenti, da qui è possibile notare che il rettangolo posto sullo sfondo più scuro viene percepito con un'intensità di grigio più chiara rispetto alle altre configurazioni, anche se in realtà è lo stesso identico colore. A seguito l'esempio figurativo:

---

<sup>12</sup> E' una membrana del bulbo oculare di colore variabile, la parte colorata dell'occhio, situata davanti al cristallino e dietro alla cornea.

<sup>13</sup> E' il meccanismo usato in ambito fotografico, per regolare la quantità di luce che deve attraversare l'obbiettivo della camera.

<sup>14</sup> La retina è coperta da recettori della luce: coni e bastoncelli, i primi sono più sensibili ai colori mentre gli altri sono sensibili ai bassi livelli d'illuminazione.

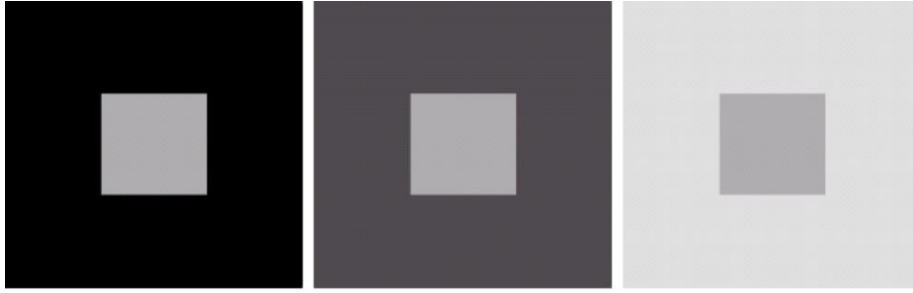


Figura 1: Intensità percepita (fonte[a]).

## 2.2 LA LUCE

Ogni corpo che presenta una temperatura superiore allo zero assoluto (0 [K])<sup>15</sup> emette radiazioni elettromagnetiche<sup>16</sup>, che sono costituite dalla parte ondulatoria e una parte corpuscolare<sup>17</sup>. La prima è creata dalle onde elettromagnetiche, le quali sono strutturate da un campo di forza elettrica e magnetica oscillante, mentre la seconda la si può descrivere come un flusso di fotoni che viaggiano nel vuoto a una velocità prossima a quella della luce. Tutto questo può essere elaborato tramite le equazioni di Maxwell sull'elettromagnetismo.

Qualsiasi oggetto o materia è composta da atomi, i quali emanano radiazioni elettromagnetiche sotto forma di fotoni, ognuno con una specifica frequenza e lunghezza d'onda. In fisica per poter studiare la luce si è dovuto definire l'insieme di queste radiazioni elettromagnetiche in tutte le possibili frequenze, come Spettro Elettromagnetico. Anche se nella realtà è continuo per convenzione viene suddiviso in varie bande a seconda della frequenza e lunghezza d'onda.

Questo spettro è suddiviso in:

- Spettro visibile, il quale ricade tra il rosso e il violetto includendo tutti i colori che l'occhio umano può percepire, da questo si ha la luce, infatti ogni corpo colpito dalla luce assorbe radiazioni elettromagnetiche e ne riflette in parte (l'assorbimento e la riflessione dipendono dal colore).

---

<sup>15</sup> Legge di Stefan - Boltzmann

<sup>16</sup> La luce ne è un esempio

<sup>17</sup> Corpuscolare o particella, nel caso della luce si parla di Fotoni, cioè è la quantità discreta e indivisibile (quanto) del campo elettromagnetico

- Spettro non visibile, suddiviso a sua volta in due regioni: nella prima si trovano le onde la cui lunghezza d'onda è maggiore rispetto a quelle visibili, che sono onde radio, microonde e infrarossi; mentre la parte dello spettro in cui sono presenti quelle con lunghezza d'onda minore rispetto a quelle intermedie (visibili), è costituito da ultravioletti, raggi gamma e raggi X, che sono quelle a più alto contenuto d'energia.

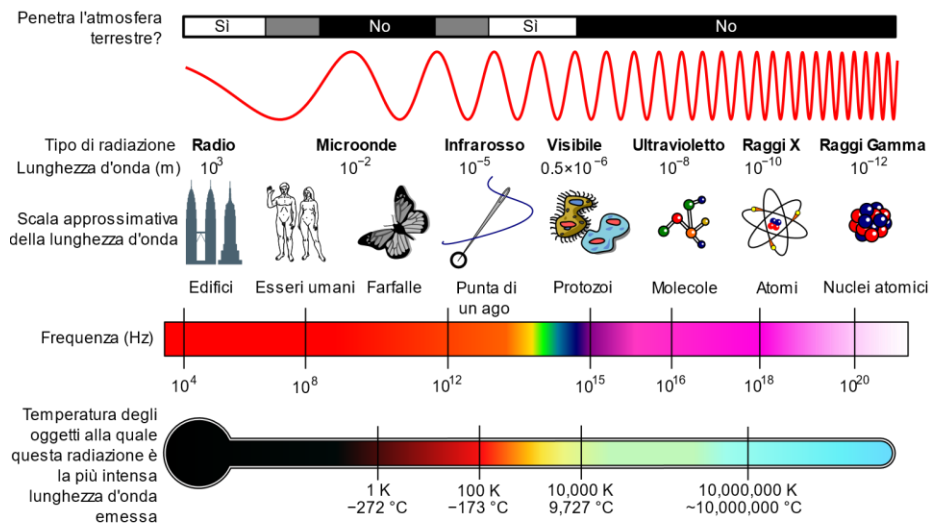


Figura 2: Spettro elettromagnetico (fonte[b])

Lo spettro ha un vasto campo d'utilizzo, per esempio nelle telecomunicazioni, a seconda del tipo di frequenza, vengono modulate le onde radio contenenti il messaggio da inviare dal trasmettitore, con la *carrier*<sup>18</sup>, dopodiché il ricevitore, demodula queste onde isolando il messaggio e l'onda di trasporto.

## 2.3 ACQUISIZIONE ED ELABORAZIONE DELLE IMMAGINI DIGITALI

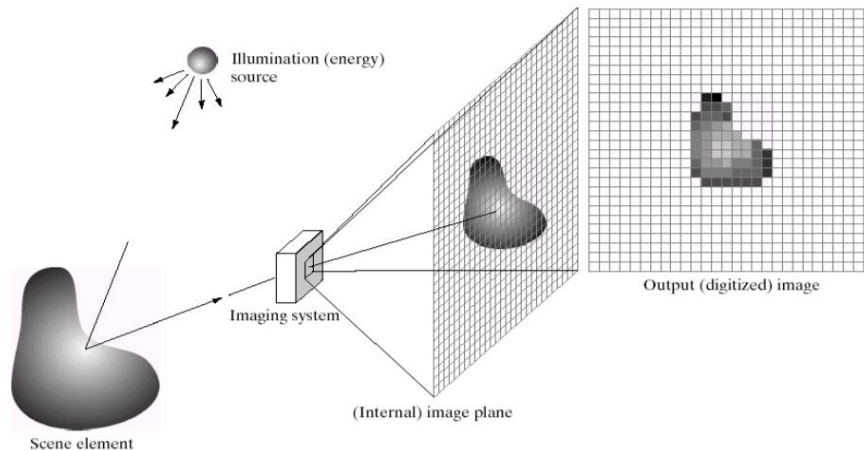
### 2.3.1 ACQUISIZIONE

L'acquisizione di un'immagine è il processo di rilevamento e di "registrazione" che ferma l'immagine inquadrata (o meglio che cattura

<sup>18</sup> E' l'onda trasportatrice del segnale, il modulatore non fa altre che sovrapporre l'onda radio contenente l'informazione da trasportare a quella carrier.

l'attimo) su un supporto adatto agli usi successivi, un tempo si utilizzava la pellicola ora qualsiasi tipo di memoria di una fotocamera. L'energia che arriva all'obiettivo è quella riflessa dal corpo che è investito dalla luce (esempio quella del sole), quest'ultima colpisce i sensori<sup>19</sup> della fotocamera e genera un segnale elettrico, dove tramite la quantizzazione si ha una conversione del segnale analogico continuo in una sua versione digitale, le cui grandezze in questione vengono discretizzate. Questa discretizzazione varia in base al numero di pixel, cioè alla sua risoluzione. Può essere d'interesse la seguente precisazione, se un oggetto di colore verde, viene colpito da una luce bianca, la maggior parte delle lunghezze d'onda vengono assorbite, tranne la luce verde che viene completamente riflessa.

L'immagine digitale una volta acquisita restituisce come risultato un'immagine numerica che viene salvata all'interno di un dispositivo, esempio la memoria di una fotocamera, da qui è possibile trasportarla e successivamente interagire con essa tramite monitor, tablet o calcolatori.



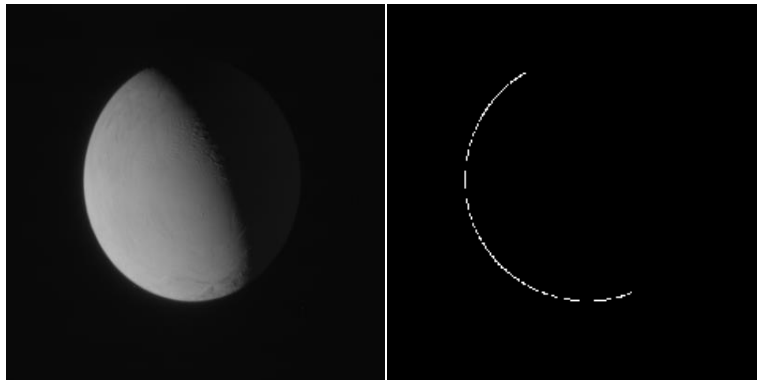
*Figura 3: Schema d'acquisizione di un'immagine digitale. Essendo il mondo uno spazio continuo digitalizzare un oggetto, può essere visto come un'approssimazione del mondo reale. (fonte[a])*

<sup>19</sup> Questi sensori vengono costruiti apposta per acquisire quel tipo d'energia.

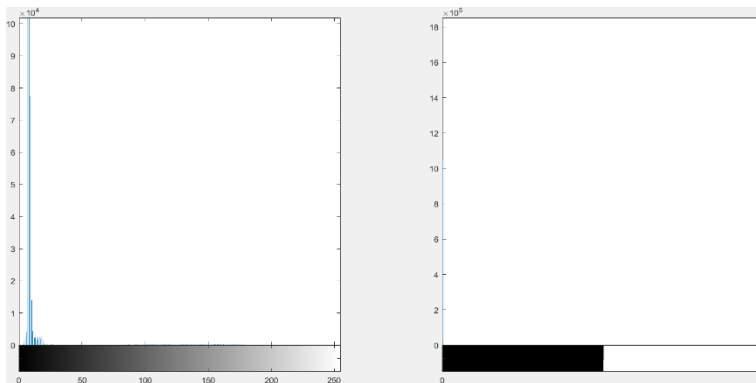
### 2.3.2 L'IMMAGINE DIGITALE COME FUNZIONE BIDIMENSIONALE

Un'immagine digitale è una funzione bidimensionale  $f(x,y)$  che rappresenta una misura di una o più caratteristiche come ad esempio la luminosità e questa funzione restituisce il valore d'intensità ad ogni pixel di coordinata  $(x,y)$ .

Un'immagine digitale monocromatica è una matrice  $I = f(x,y)$  di valori discreti d'intensità luminosa sui vari livelli di grigio, presi da un intervallo  $[0, L - 1]$  essendo  $L = 2^k$ , dove  $k$  è il numero di bit utilizzato per codificare ciascun pixel. Normalmente si utilizzano 8 bit perciò i valori vanno da 0 (nero) a 255 (bianco). In generale per l'elaborazione delle immagini si usa un  $k = 1$ , perciò l'intervallo va da 0 quando il pixel è spento a 1 quando il pixel è acceso. Si potrà notare infatti la differenza tra i due istogrammi rappresentativi dell'immagine in scala di grigi (da 0 a 255) e quella elaborata tramite edge detection (da 0 a 1), in *figura 5*.



*Figura 4: foto d'Encelado scattata dalla sonda Cassini, sulla sinistra si ha l'immagine originale, a destra l'elaborato tramite il metodo Canny. (fonte[c])*



*Figura 5: istogramma, a sinistra si ha quello relativo all'immagine originale, mentre quello a destra si riferisce all'edge detection.*

Un'immagine a colori può essere decomposta tramite diverse tecniche o applicando dei filtri nei tre colori base, che se miscelati tra loro possono dare quasi tutti i colori dello spettro visibile dall'uomo. Questo modello di colori prende il nome di RGB<sup>20</sup> ed è di tipo additivo, cioè le combinazioni derivano dalla somma di queste tre basi: rosso, verde e blu, a ogni pixel quindi vengono forniti tre valori d'intensità ognuno che rappresenta i tre colori sopracitati.

Invece per le immagini in scala di grigio è sufficiente fornire un solo valore d'intensità, che può andare da 0 a 255 o da 0 a 1, a seconda dei bit usati per ogni pixel, questo ha il vantaggio che ogni immagine per essere elaborata ha un costo minore in termini di memoria.

I pixel<sup>21</sup> non sono altro che i punti d'informazione di un'immagine, maggiore è il loro numero all'interno della figura, maggiore sarà la risoluzione spaziale<sup>22</sup>. Come è facile intuire più un'immagine è risolta più è alto il suo livello di dettaglio e di qualità digitale.

La risoluzione viene misurata in dpi ("dots per inch") oppure in pixel su centimetro, per apparecchi da ripresa come le fotocamere, si devono contare quanti sensori ci sono per unità lineare di misura, MEGAPIXEL.

Un esempio per capire il costo in bit per la memorizzazione di un'immagine a colori è il seguente: presa un'immagine quadrata di lato 200 pixel, con 1 pixel su cm:

$$200 * 200 * 24bit = 960\ 000\ bit$$

Dove i 24 bit sono composti da: 8bit per indicare l'intensità del rosso, otto per il verde e otto per il blu.

---

<sup>20</sup> Red, Green e Blue, è il nome di uno dei modelli di colori formato dalla CIE nel 1931.

<sup>21</sup> Picture Elements.

<sup>22</sup> Definizione: si dice risoluzione il numero di pixel per unità di misura.



### 3. IMAGE PROCESSING

Il processamento delle immagini è una disciplina digitale, che comporta l'utilizzo di software, con specifici linguaggi di compilazione, i quali tramite l'ausilio di algoritmi di elaborazione numerica, possono interagire con essa. Presa un'immagine, attraverso varie funzioni che dipendono dal tipo di programma utilizzato, è possibile estrapolarne le sue caratteristiche principali, come ad esempio il numero e l'intensità dei pixel, la grandezza dell'immagine, l'individuazione del contorno e molte altre peculiarità.

#### 3.1 L'AMBIENTE DI CALCOLO

Partiamo da uno dei rami fondamentali della matematica, cioè *l'analisi matematica* che riguarda le proprietà ottenute dalla scomposizione infinita di un oggetto denso. Si basa sul calcolo infinitesimale, cioè tramite le definizioni di continuità e di limite studia il comportamento locale di una funzione, avvalendosi di strumenti come derivata e integrale. Facendo un passo in avanti è possibile entrare in una branca della matematica applicata chiamata *analisi numerica*<sup>23</sup>, quest'ultima si occupa di risolvere i modelli ottenuti dall'analisi matematica alle scomposizioni finite normalmente praticabili, introducendo un concetto fondamentale per l'ingegneria, l'approssimazione. Gli algoritmi che fanno parte di questo campo sono caratterizzati in base alla velocità di convergenza.

Il software utilizzato come anticipato nell'introduzione è Matlab<sup>24</sup>, cioè un ambiente di calcolo numerico, scritto in C, il quale consente la manipolazione e la visualizzazione di funzioni, matrici, vettori, l'implementazione di algoritmi, la creazione d'interfacce con l'utente e l'elaborazione delle immagini.

---

<sup>23</sup> Calcolo Numerico.

<sup>24</sup> Abbreviazione di **Matrix Laboratory**.

## 3.2 RILEVAMENTO DEI BORDI IN UN'IMMAGINE: *EDGE DETECTION*

L'*edge detection* è uno dei rami del riconoscimento delle caratteristiche di un'immagine ed è utilizzato per rilevarne i bordi, cioè marcare i punti di una qualsiasi immagine digitale in cui l'intensità dei pixel varia rapidamente (dove si ha un gradiente elevato dell'intensità dei pixel). Questi bruschi cambiamenti possono essere dovuti all'ambiente circostante al soggetto in questione, alla discontinuità della profondità o all'orientamento delle superfici, dalle proprietà dei vari materiali presenti nell'immagine e soprattutto dal tipo di illuminazione presente. Queste cause portano ad avere, nel mondo fisico dell'immagine, delle rapide variazioni d'intensità nelle regioni prossime al "passaggio" da un soggetto (esempio: un pianeta, una persona, una macchina, ...) ad un altro presente in essa (esempio lo sfondo).

Il riconoscimento dei bordi si può pensare come un'operazione che ha il compito di creare un'immagine "sintetica", nel quale vengono evidenziate solo le informazioni di cui l'utente ha bisogno a seconda dei suoi obiettivi. In sintesi questa funzione elimina la maggior parte dei dettagli non rilevanti lasciando solo le informazioni essenziali contenute nell'immagine originale, le quali descrivono la forma e le caratteristiche strutturali e geometriche dei corpi presenti al suo interno.

I metodi del rilevamento dei bordi sono per la maggior parte raggruppati in due macro categorie: *search-based* (basati sulla ricerca degli zeri) e *zero-crossing* (basati sull'attraversamento dello zero). I primi come ad esempio il metodo Prewitt o Sobel, riconoscono i contorni cercando i massimi e i minimi avvalendosi delle derivate del primo ordine dell'immagine, essendo essa una funzione bidimensionale  $f(x,y)$ . Il secondo metodo, invece cerca i punti in cui la derivata del secondo ordine passa per lo zero.

Un esempio molto semplice che spieghi cosa sia l'individuazione dei contorni tramite una derivata è il seguente: come detto in precedenza la regione limite tra due corpi (corpo e sfondo ad esempio) di un'immagine, è identificato attraverso un cambiamento d'intensità dei pixel in quella

regione. Perciò prendendo una sola linea di pixel per semplicità, è possibile supporre che il bordo si trovi tra il nono e il decimo pixel:

12   14   11   19   16   24   27   16   16   125   140   133   119   131   127   140
--------------------------------------------------------------------------------------

ovviamente bisogna stabilire precedentemente i criteri per i quali la differenza si può considerare tale da essere identificata come un contorno. Prima di parlare del metodo utilizzato in questa tesi per l'individuazione del bordo, per una maggior chiarezza verranno definiti i filtri e quali tipi esistono per le immagini.

### 3.2.1 FILTRI

Nell'*image processing* è spesso utile migliorare l'immagine per raggiungere lo scopo prefissato, attraverso l'eliminazione o l'attenuazione di alcune delle sue caratteristiche, per farlo si limano determinati componenti frequenziali dell'immagine.

Principalmente è possibile filtrare l'immagine nel dominio delle frequenze e nello spazio.

Il filtraggio di un'immagine nel dominio della frequenza consiste nel modificare la sua trasformata di Fourier e subito dopo calcolarne la sua antitrasformata, ottenendo così il risultato desiderato.

Come già detto in precedenza un'immagine digitale è una funzione bidimensionale  $f(x, y)$  di dimensioni  $M \times N$ , per filtrare questa immagine bisogna applicare l'equazione:

$$r(x, y) = F^{-1}[H(u, v)F(u, v)]$$

dove  $F^{-1}$  è l'antitrasformata di Fourier,  $F(u, v)$  è la trasformata dell'immagine originale  $f(x, y)$  e  $H(u, v)$  è la funzione filtro, mentre  $r(x, y)$  è l'immagine di partenza con l'aggiunta del filtro. Sia  $F$  che  $H$  hanno sono matrici con le dimensioni dell'immagine di partenza.

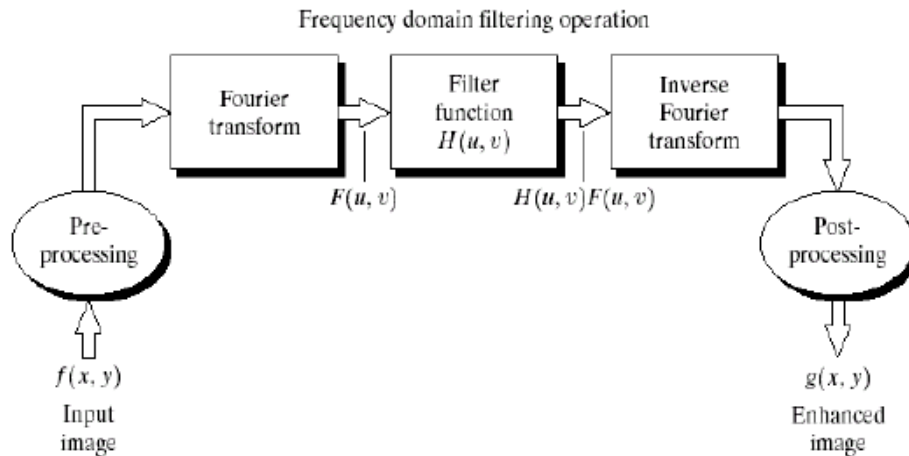


Figura 6: Schema del filtraggio di un'immagine nel dominio della frequenza. (fonte[d])

Le frequenze nella trasformata di un'immagine sono suddivise in basse e in alte, le prime hanno una relazione con le componenti dell'immagine che variano in maniera graduale la propria intensità. Le alte invece, sono relazionate con le parti in cui si hanno brusche variazioni d'intensità, come ad esempio i bordi e il rumore.

Esistono tanti tipi di filtri, uno di questi è *high pass gaussiano*, il quale aumenta la nitidezza digitale e riduce il contrasto dell'immagine, un altro tipo è il *low pass gaussiano* che al contrario del primo sfoca le immagini. Esiste un legame tra il filtraggio nel dominio spaziale e quello della frequenza e si basa sul teorema della *convoluzione*.

Il teorema della convoluzione (bidimensionale), dice che sotto opportune condizioni la trasformata di Laplace e di Fourier della convoluzione di due funzioni è il prodotto delle trasformate delle rispettive funzioni, date:

$$f(x, y), h(x, y) \quad (2)$$

abbiamo che

$$f(x, y) * h(x, y) \stackrel{\text{def}}{=} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n) \quad (3)$$

dove

$$x = 0, \dots, M - 1$$

$$y = 0, \dots, N - 1$$

Il teorema è dato dalle espressioni:

$$f(x, y) * h(x, y) \leftrightarrow F(u, v)H(u, v)$$

viceversa

$$f(x, y)h(x, y) \leftrightarrow F(u, v) * H(u, v)$$

Tornando al legame tra i due domini, tramite un filtro spaziale è possibile ottenere la sua rappresentazione nel dominio della frequenza grazie alla trasformata di Fourier del filtro spaziale e da qui è applicabile come filtro di frequenza all'immagine. Infatti si ha che i due filtri, quello di partenza e la sua trasformata, formano una coppia di Fourier:

$$h(x, y) \leftrightarrow H(u, v)$$

dove  $h(x, y)$  è un filtro spaziale.

Il filtraggio nel dominio dello spazio invece, è basato sulla convoluzione tra l'immagine (nelle sue coordinate spaziali) ed un filtro avente una funzione di risposta impulsiva chiamato kernel<sup>25</sup>:

$$r(i, j) = h(i, j) \odot f(i, j) \quad (4)$$

dove  $f(i, j)$  è la matrice dell'immagine originale,  $\odot$  è il prodotto di convoluzione tra due matrici,  $h(i, j)$  è il filtro o kernel che viene applicato e  $r(i, j)$  è il risultato finale, cioè l'immagine filtrata o elaborata. Solitamente le matrici di filtraggio sono dispari per identificare meglio il centro (il pixel che verrà modificato tramite questo filtro è posizionato al centro della matrice) e vengono usate per sfocatura, goffatura, riconoscimento dei bordi.

### 3.2.2 FILTRO GAUSSIANO

Un esempio di corrispondenza tra i due domini è il filtro Gaussiano<sup>26</sup> nel dominio della frequenza (tipo monodimensionale), le cui trasformate e antitrasformate sono tutte funzioni reali:

$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}} \quad (5)$$

dove  $\sigma$  è la deviazione standard<sup>27</sup> della curva gaussiana. Ora per ottenere il filtro nel dominio spaziale basta applicare l'antitrasformata di Fourier:

$$h(x) = \sqrt{2\pi}Ae^{-2\pi^2\sigma^2x^2} \quad (6)$$

<sup>25</sup> In italiano significa Nucleo, a volte il kernel viene chiamato anche "maschera di un'immagine"

<sup>26</sup> Le funzioni Gaussianhe sono chiamate così in memoria del grande matematico tedesco Carl Friedrich Gauss, hanno tante applicazioni e una forma che ricorda quella di un impulso. Tutte quelle che hanno  $\sigma^2$  sono delle funzioni della trasformata di Fourier.

<sup>27</sup> Ampiezza della Curva

si può osservare che se  $\sigma \rightarrow \infty$ ,  $h(x)$  è una funzione costante e la sua trasformata  $H(u)$  è un impulso (viceversa).

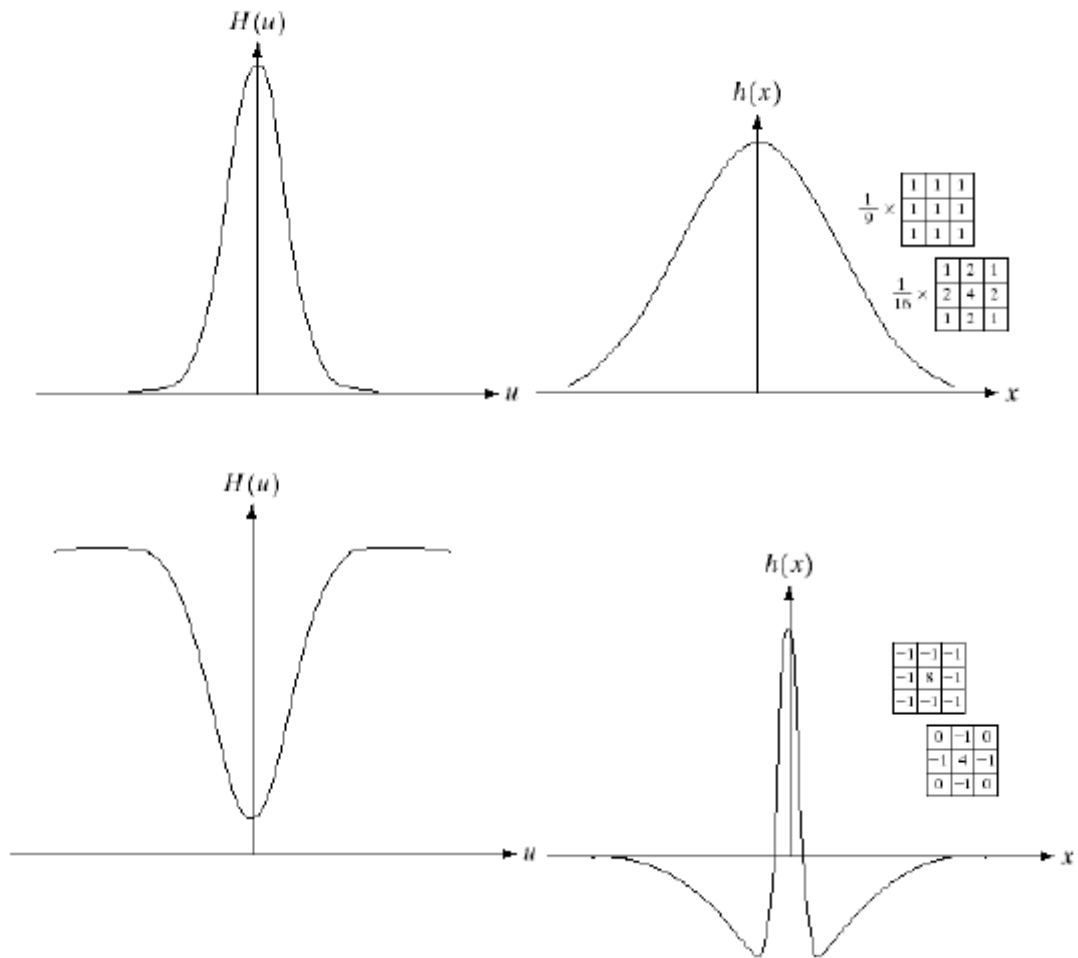


Figura 7: Antitransformate di un filtro gaussiano lowpass in alto, dal dominio delle frequenze a quello spaziale, mentre in basso si ha l'antitransformata di un filtro highpass. (fonte[e])

È possibile creare sia filtri passa basso che passa alto come già accennato in precedenza e come visibile in figura 7. La forma di un *high pass gaussiano* è la seguente:

$$H(u) = Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}} \quad (7)$$

con  $A \geq B$  e  $\sigma_1 > \sigma_2$ , è possibile notare che è il risultato della differenza di due filtri gaussiani. Il suo corrispondente nel dominio spaziale è:

$$h(x) = \sqrt{2\pi}Ae^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}Be^{-2\pi^2\sigma_2^2x^2} \quad (8)$$

Uno dei tanti vantaggi di poter lavorare nel dominio delle frequenze è quello di poter sopprimere le frequenze indesiderate e ridurre lo spazio occupato dei dati.

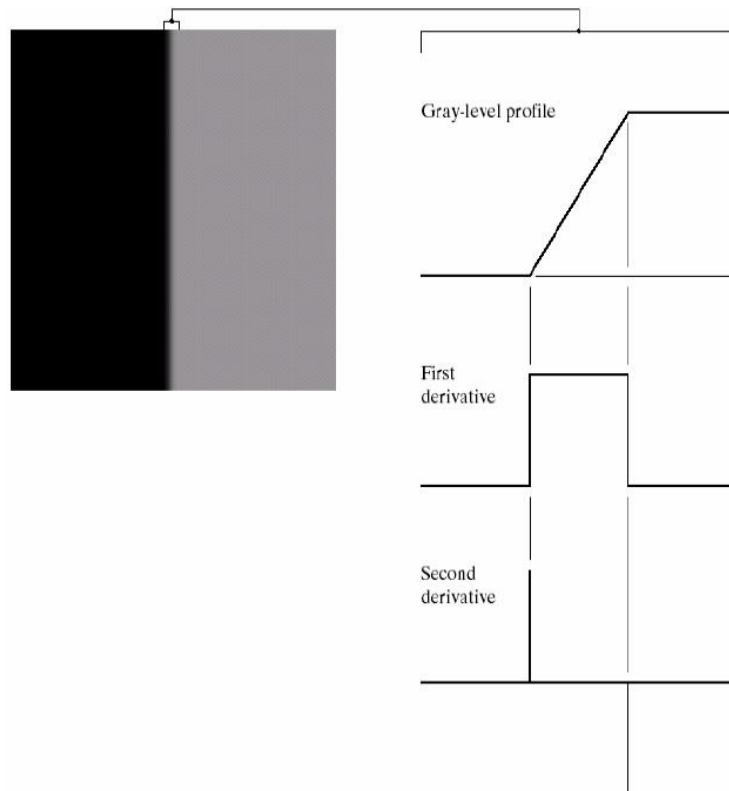
### 3.2.3 DERIVATA DI UN'IMMAGINE

La derivata di un segnale denota la sua variabilità a fronte di forti variazioni locali (esempio d'intensità) e li assume valori elevati, mentre se è uguale a zero il segnale è costante. Se utilizziamo dei segnali bidimensionali, come le immagini (matrice  $M \times N$ ), dobbiamo utilizzare le derivate parziali. Ricordando le definizioni di derivate parziali:

$$\frac{\partial f(x, y)}{\partial x} = \lim_{k \rightarrow \infty} \frac{f(x + k, y) - f(x, y)}{k} \rightarrow f(x + 1, y) - f(x, y)$$

$$\frac{\partial f(x, y)}{\partial y} = \lim_{h \rightarrow \infty} \frac{f(x, y + h) - f(x, y)}{h} \rightarrow f(x, y + 1) - f(x, y)$$

Si può dire che fare la derivata prima di un'immagine equivale a fare l'operazione di convoluzione di quest'ultima tramite i filtri derivativi. È possibile fare un esempio prendendo la riga di separazioni tra nero e grigio, dell'immagine in *figura 8* mostrando la sua funzione e le sue derivate:



*Figura 8: La riga evidenziata viene rappresentata tramite la prima funzione che è costante, lineare e nuovamente costante. Successivamente sono raffigurate la derivata prima e seconda. (fonte[f])*

Rispetto a  $x$  si applica il filtro:

$$-1 \quad 1$$

rispetto a  $y$ :

$$\begin{matrix} -1 \\ 1 \end{matrix}$$

L'operazione di convoluzione come già visto precedentemente nell'Eq. (3) è:

$$f(x, y) * h(x, y) \stackrel{\text{def}}{=} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n).$$

Come già precedentemente detto, la derivata di un segnale denota la sua variabilità, questo è possibile vederlo con l'esempio nelle *figure 9 e 10*, in cui si è presa una regione irregolare come una zona montuosa e ci si è posti a una certa coordinata  $y$  (segnata in rosso), *figura 9*:

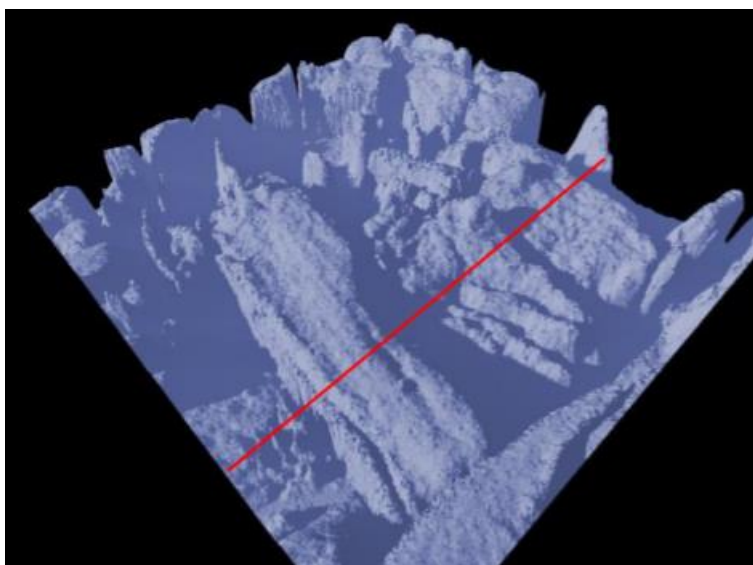


Figura 9: Rilevamento tipo Radar di una zona montuosa. (fonte[g])

è possibile notare in *figura 10*, come la derivata lungo  $x$  (essendo in  $y$  uguale a zero) del segnale differisca dalla funzione associata all'immagine della zona montuosa.

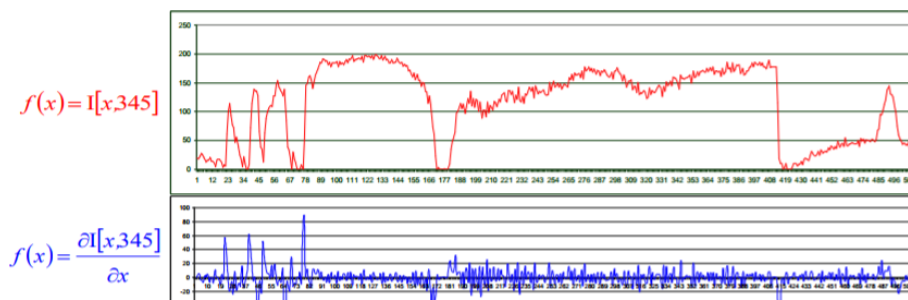


Figura 10: Viene messa in mostra la differenza tra il segnale che identifica l'immagine lungo la linea rossa e la sua derivata. (fonte[g])



### 3.2.4 GRADIENTE DI UN'IMMAGINE

Il gradiente di un'immagine è il vettore che indica la direzione in cui si ha la maggiore variazione d'intensità in quel punto dell'immagine e le sue componenti sono le derivate parziali nelle diverse direzioni. Quanto detto è facilmente intuibile nelle seguenti immagini in *figura 11*:

$$\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}; \frac{\partial f(x, y)}{\partial y} \right) \quad (9)$$



*Figura 11: da sinistra a destra abbiamo l'immagine a), b) e c) (fonte[h])*

I gradienti relativi a ogni figura sono i seguenti:

- a)  $\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}; 0 \right)$ .
- b)  $\nabla f(x, y) = \left( 0; \frac{\partial f(x, y)}{\partial y} \right)$ .
- c)  $\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}; \frac{\partial f(x, y)}{\partial y} \right)$ .

Il modulo è calcolato nella stessa maniera in cui si calcola quello di un normale vettore, come si può vedere nell'Eq. (10):

$$\|\nabla f(x, y)\| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} = |\nabla_x| + |\nabla_y| \quad (10)$$

Per l'orientamento del gradiente, quando si lavora con le immagini, bisogna prestare attenzione alla direzione  $y$ , perché solitamente i software tipo Matlab ma più in generale i PC considerano la coordinata d'origine  $f(0,0)$  in alto a sinistra. L'orientazione del gradiente viene quindi calcolata tramite l'arcotangente a due quadranti<sup>28</sup>:

$$\theta = \tan^{-1}_2(\nabla_y, \nabla_x) \quad (11)$$

<sup>28</sup>  $\tan^{-1}_2(b, a)$  restituisce l'angolo del vettore da  $(0,0)$  al punto di coordinata  $(a, b)$ .

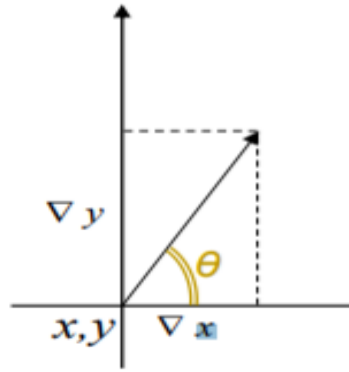


Figura 12: Gradiente. (fonte[g])

### 3.3 METODO CANNY

In questa tesi l'unico operatore utilizzato per la realizzazione dell'algoritmo per la determinazione dei centroidi d'immagini di corpi celesti è il *Canny Method*. Il codice per l'applicazione è spiegato nel Capitolo 5.3.

Il metodo Canny è un algoritmo di individuazioni e tracciamento dei bordi creato dallo scienziato australiano Jhon F. Canny nel 1986 ed è al giorno d'oggi uno degli operatori più utilizzati e più efficienti per questo tipo di tecniche di riconoscimento delle caratteristiche delle immagini. Lo si può vedere come composto da due macrofasi: la prima consiste nell'applicazione di un filtro di tipo gaussiano (vedi Paragrafo 3.3.2) per l'eliminazione del rumore nell'immagine, mentre la seconda fase si occupa del rilevamento dei bordi, in maniera simile a i metodi Sobel e Prewitt<sup>29</sup> (questi metodi non saranno trattati in questa tesi, perché esulano dall'obiettivo imposto). Queste fasi principali possono essere riassunte in quattro step:

#### 3.3.1 RIDUZIONE DEL RUMORE

Questo stadio viene compiuto prima dell'elaborazione dell'immagine e serve per l'eliminazione di influenze negative presenti dovute al rumore, quest'ultimo si crea al momento dell'acquisizione. Il suo compito è perciò

<sup>29</sup> Questi due metodi sono stati accennati in precedenza e fanno parte della "macrocategoria" search-based, i cui metodi riconoscono i contorni cercando i massimi e i minimi della derivata prima, non verranno discussi in questa tesi.

quello di evitare i bordi generati dal disturbo e cercare solo quelli appartenenti agli elementi di scena. Per poter compiere questa operazione di “purificazione” è necessario avvalersi di un operatore matriciale chiamato *kernel*, che è basato sulla derivata prima di un filtro gaussiano, la quale viene calcolata nel pixel in analisi di coordinate  $(x_0, y_0)$ . Questo *kernel* viene sovrapposto alla matrice dei pixel relativi all’immagine originale, facendo in modo che il suo centro corrisponda con  $(x_0, y_0)$ , il quale durante l’operazione rimane inalterato, mentre ogni posizione  $(x, y)$  del filtro moltiplica la rispettiva coordinata della matrice dei pixel dell’immagine originale per valori che dipendono dalle caratteristiche del *kernel*. Fatto questo si ottiene una nuova matrice con la stessa dimensione del filtro usato, il cui determinante sarà il nuovo valore che verrà sostituito al pixel di coordinate  $(x_0, y_0)$ . Questo processo viene applicato a ogni pixel appartenente alla funzione originale. Il risultato che si ottiene è un’immagine con le stesse dimensioni di quella autentica, con una leggera sfocatura gaussiana, in cui non sono presenti pixel con disturbi di livello significativo. Questa operazione è detta anche *smoothing* e dipende dalla dimensione del filtro (che per individuare con facilità il centro solitamente è dispari) e dalla sua ampiezza  $\sigma$ . Riguardo alla prima più è piccola e più si ha una minore sfocatura ed è possibile identificare bordi più netti, mentre se più è grande maggiore sarà la sfocatura finale sulle immagini (con contorni ben più ampi e sfumati)

Un esempio di filtro gaussiano bidimensionale che può essere applicato dal metodo Canny è il seguente:

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

### 3.3.2 GRADIENTE

Il secondo step è il calcolo del gradiente, in questo algoritmo vengono utilizzati quattro tipi di filtri differenti per individuare i contorni: orizzontali, verticali e i due diagonali i quali vengono sovrapposti all’immagine ottenuta dopo l’operazione di *smoothing*. Ognuno di questi filtri restituisce un valore d’intensità per ogni pixel e quello con il valore maggiore, da un’indicazione sulla direzione da considerare. Il valore e

l'orientamento insieme corrispondono alla direzione in cui si ha il massimo gradiente di luminosità in ciascun punto dell'immagine. Gli operatori più semplici per calcolare il gradiente che tengono conto della rotazione degli assi sono quelli Prewitt.

### 3.3.3 SOPPRESSIONE DEI NON-MASSIMI

Dopo aver calcolato il gradiente per ogni pixel si ottiene come risultato una mappa di questi vettori, la quale fornisce il valore dell'intensità in ciascun punto dell'immagine. Se si è in presenza di alti valori, significa che ci sono buone possibilità che il pixel in analisi sia parte di un bordo, però non si può avere la certezza di questo. Per aumentare le probabilità che la predizione sia corretta, lo step successivo prenderà in considerazione solo i pixel in cui la derivata del gradiente è uguale a zero, quindi si sarà in presenza dei massimi locali.

Un esempio di verifica della condizione di massimo locale nell'intorno  $3 \times 3$  del pixel  $P$  rappresentato nella *figura 13*, è il seguente:

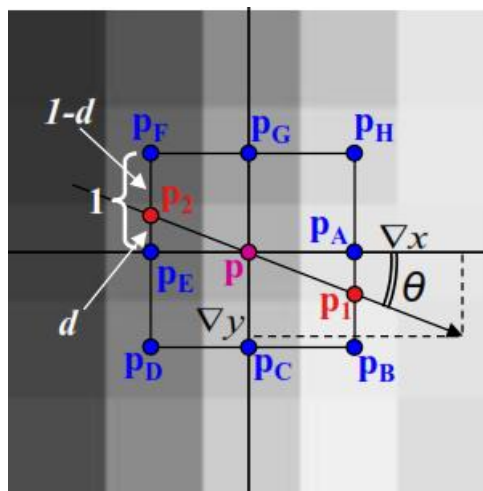


Figura 13. (fonte[g])

bisogna fare una stima del gradiente nei punti  $P_1$  e  $P_2$  (la direzione coincide con l'aumento d'intensità dei pixel) tramite interpolazione lineare, da qui il pixel  $P$  verrà conservato se e solo se:

$$\|\nabla P\| \geq \|\nabla P_1\| \wedge \|\nabla P\| \geq \|\nabla P_2\|$$

Se rispetta questa condizione il pixel  $P$  verrà identificato come massimo locale, nel caso in figura questo non risulta perché il gradiente che si ha nel punto  $P_1$  è maggiore di quello in  $P$ .

### 3.3.4 EDGE DETECTION TRAMITE ISTERESI

A differenza di altri metodi nel metodo Canny si utilizza un'estrazione dei contorni tramite una *sogliatura*<sup>30</sup> con *isteresi*<sup>31</sup> al fine di selezionare solo gli edge significativi ma evitando la frammentazione. Quindi vengono definite due soglie una alta  $T_1$  e una bassa  $T_2$ , con  $T_1 > T_2$ , le quali vengono confrontate con il gradiente di ciascun punto ottenuto nella fase precedente, per scremare i massimi locali. Se il valore del gradiente è:

- Inferiore alla soglia bassa  $T_1$ , il pixel viene scartato
- Superiore alla soglia alta  $T_2$ , il punto è accettato come parte di un contorno.
- Tra le due soglie, il punto viene accettato solo se contiguo ad un punto che è stato precedentemente accettato.

Le due soglie vengono utilizzate perché non esiste un unico valore del gradiente d'intensità luminosa che ci permetta di dire se un pixel fa parte o no di un bordo. I valori degli estremi di questo intervallo che sono stati utilizzati in questa tesi, variano da immagine a immagine, ma il range massimo considerato va da  $T_2 = 0.05$  a  $T_1 = 0.99$ . La scelta dei valori è molto importante, perché il risultato che si ottiene è differente da intervallo a intervallo e di conseguenza anche l'ellisse approssimante può risultare molto diversa.

Alla fine di questo ultimo passo si otterrà un'immagine binaria, dove ciascun pixel è marcato come appartenente all'immagine o no. La mappa rimanente la si può vedere come l'insieme di curve di contorno che possono essere approssimate tramite una poligonale.

---

<sup>30</sup> La sogliatura o thresholding nell'elaborazione digitale delle immagini è un metodo per segmentare un'immagine, cioè durante questo processo ogni pixel dell'immagine viene catalogato come pixel oggetto se è sopra una certa soglia o pixel sfondo se è sotto.

<sup>31</sup> Il riferimento viene fatto perché vengono applicati due valori di soglia

## 4. ELLIPSE FITTING

Come già spiegato nel Capitolo 1, per via degli obiettivi sempre più grandi e importanti delle missioni spaziali, come ad esempio inviare l'uomo su Marte o nuovamente sulla Luna, si è incentivata la ricerca per la realizzazione di nuovi di sistemi di navigazione che possano essere indipendenti dai segnali di Terra. Uno di questi sistemi è *la navigazione ottica usando il centroide e il diametro apparente dei pianeti*, la quale si basa sulle immagini scattate dalla fotocamera presente sulla navicella o sonda in cui tramite vari metodi di identificazione del limbo (*edge detection*) verrà identificato il limbo del corpo celeste visibile dall'obiettivo della camera. Fatto questo tramite funzioni di regressione ad ellisse (*ellipse fitting*), il software di navigazione approssima il limbo individuato ad un'ellisse che ha il compito di rappresentare la stima della forma reale del corpo celeste. Ognuna di queste funzioni di regressione si basa su differenti metodi d'approssimazione i quali verranno discussi in questo Capitolo e in cui verranno mostrati ampiamente i risultati nei Capitoli 5 e 6. Dopo aver trovato una forma più o meno veritiera del Satellite o Pianeta in studio, il software inoltre ne calcolerà le coordinate del centroide, questo dato servirà poi al computer di bordo per capire l'orientamento e/o la posizione del satellite artificiale (navicella, sonda, ...) rispetto al corpo celeste di cui è nota la posizione. Oppure si può ragionare in senso contrario, se è conosciuta la posizione della sonda, è possibile conoscere la posizione relativa ad essa del soggetto in analisi di cui non si conosce la posizione. Prima di passare ad analizzare i vari metodi di regressione ad ellisse, può risultare utile fare un breve riassunto sulle caratteristiche geometriche dell'ellisse e sulla sua definizione, inoltre verrà data una motivazione sul perché è possibile approssimare i corpi celesti ad un'ellisse.

### 4.1 ELLISSE

L'ellisse è il luogo geometrico dei punti del piano per i quali la somma delle distanze di un punto rispetto ai due fuochi, rimane costante. Questa

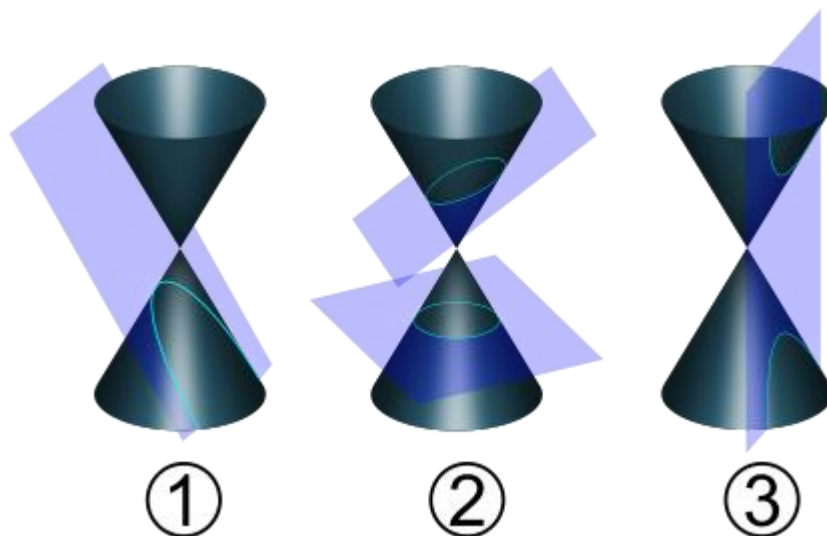


Figura 14: In 1) si ha una parabola, nella parte in alto della figura 2) si ha l'ellisse, in basso una circonferenza, mentre nella 3) si hanno due iperboli.  
(fonte[i])

curva piana fa parte della famiglia delle coniche<sup>32</sup> ed è possibile ottenerla, intersecando un cono con un piano avente inclinazione superiore a quella della generatrice del cono rispetto al suo asse com'è possibile vedere in figura 14 nella parte 2) in alto, mentre le altre due figure 1) e 2) sono Parabola e Iperbole le quali vengono ottenute quando l'inclinazione del piano è pari o inferiore a quella della retta generatrice rispetto l'asse del cono. Il cerchio è una particolare ellisse che si ottiene quando il piano ha una direzione ortogonale all'asse del cono. Inoltre se si prende una circonferenza che giace su un piano inclinato, è possibile ottenere una proiezione verticale di un'ellisse su un piano orizzontale, chiamando con  $\theta$  l'angolo tra il piano orizzontale e quello inclinato e  $\sin \theta$  l'eccentricità dell'ellisse. L'eccentricità  $e$  di un'ellisse da un'indicazione ben precisa del suo schiacciamento, questo valore è compreso tra 0 e 1 ed è il rapporto tra le distanze dei due fuochi  $P_1 = (c; 0)$  e  $P_2 = (-c; 0)$  e la lunghezza del semiasse maggiore  $2a$ :

<sup>32</sup> Sono delle particolari curve piane che vengono ottenute da un cono a doppia falda (come in figura 14) e sostanzialmente si dividono in due grandi famiglie: non degeneri e degeneri. Nella prima fanno parte: parabola, iperbole, ellisse e circonferenza; mentre nella seconda: punto, retta e due rette incidenti.

$$e = \frac{2c}{2a} = \frac{c}{a} \quad (12).$$

Quando l'eccentricità è uguale a zero, vuol dire che la distanza tra i due fuochi è nulla e quindi degenera in una circonferenza di raggio  $a$ .

Uno dei metodi più utili per descrivere un'ellisse dal punto di vista matematico, è trovare le sue caratteristiche partendo dall'equazione delle coniche la quale è un'equazione quadratica<sup>33</sup>:

$$Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Fy_i + G = 0 \quad (13)$$

dove  $[x_i, y_i]$  sono le coordinate geometriche dei punti dell'ellisse se e solo se vale:

$$4AC > B^2$$

Quest'equazione è possibile riscriverla in forma matriciale, che è la stessa forma introdotta anche nel software Matlab in questa tesi:

$$[x_i \ y_i \ 1] \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & F/2 \\ D/2 & F/2 & G \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0 \quad (14)$$

Oppure è possibile utilizzare un'altra notazione matriciale che semplificherà la lettura, definendo:

$$m_i = [x_i^2 \ x_iy_i \ y_i^2 \ x_i \ y_i \ 1]^T$$

$$a = [A \ B \ C \ D \ F \ G]^T$$

l'Eq. (14) diventa:

$$m_i^T a = 0 \quad (15)$$

Dopo aver definito l'ellisse nella sua forma implicita e nella sua forma matriciale, verranno ora esplicitate le equazioni per il calcolo delle proprietà principali di questa curva:

- Centroide:

$$x_c = \frac{2CD - BF}{B^2 - 4AC}; \quad y_c = \frac{2AF - BD}{B^2 - 4AC}; \quad (16)$$

- Semiasse maggiore:

$$a = \frac{\sqrt{2[AF^2 + CD^2 - BDF + G(B^2 - 4AC)]}}{(B^2 - 4AC) \left[ \sqrt{(A - C)^2 + B^2} - A - C \right]} \quad (17)$$

- Semiasse minore:

---

<sup>33</sup> Un'equazione quadratica è un'equazione di secondo grado con due variabili reali.



$$b = \frac{\sqrt{2[AF^2 + CD^2 - BDF + G(B^2 - 4AC)]}}{\sqrt{(B^2 - 4AC) [-\sqrt{(A - C)^2 + B^2} - A - C]}} \quad (18)$$

- Angolo d'orientamento:

$$\alpha = \begin{cases} 0 & B = 0 \wedge A < C \\ \frac{\pi}{2} & B = 0 \wedge A > C \\ \frac{1}{2} \cot^{-1} \left( \frac{A - C}{B} \right) & B \neq 0 \wedge A < C \\ \frac{\pi}{2} + \frac{1}{2} \cot^{-1} \left( \frac{A - C}{B} \right) & B \neq 0 \wedge A > C \end{cases} \quad (19)$$

## 4.2 RELAZIONE TRA CORPI CELESTI ED ELLISSI

In questo paragrafo verrà spiegato com'è possibile partendo da un insieme di punti trovati all'orizzonte dalla fotocamera di un satellite artificiale, poter calcolare la posizione corpo celeste conoscendo quella da dove è stata scattata la foto. Oppure, come verrà trattato nel seguito, sarà possibile conoscere la posizione del satellite artificiale conoscendo quella del Pianeta o Satellite Naturale (come ad esempio la Terra, Saturno, Encelado, Mimas, ...). La tecnica utilizzata per conoscere la posizione del satellite, è come già detto in precedenza, basata sull'ellisse d'approssimazione. È stato ritenuto fondamentale, fare un'introduzione del *Pinhole Camera Model* per dare al lettore una maggiore chiarezza durante lo studio dei Paragrafi successivi.

### 4.2.1 PINHOLE CAMERA MODEL

*Pinhole Camera* o *Stenoscopia*, è il procedimento fotografico che sfrutta il principio della camera oscura per la riproduzione delle immagini. La fotocamera utilizza questo foro stenopeico, per proiettare la luce, la quale viaggia in linea retta, sul materiale fotosensibile interno alla camera, formando dei piccoli puntini. Un tempo era utilizzato la pellicola mentre al giorno d'oggi per le fotocamere digitali è spesso utilizzato il *Focal Plane Array* (FPA), del quale ne verrà accennato il funzionamento a breve. L'immagine che si viene a creare dai puntini impressi nel materiale fotosensibile è invertita di 180° rispetto a quella reale, inoltre la sua qualità

dipende dalle dimensioni del foro e dal tempo di esposizione; per esempio con un foro piccolo si avrà un'immagine più nitida ma necessita di un tempo d'esposizione più elevato alla luce.

Il FPA o *Array di Fissaggio* è un dispositivo utilizzato per la rilevazione e acquisizione d'immagini digitali ed è costituito da un array di pixel sensibili alla luce sul piano focale di una lente<sup>34</sup>, solitamente ha una forma di tipo rettangolare.

Il *Pinhole Camera Method* si occupa di trovare una relazione tra le coordinate spaziali 3D del corpo in studio e la sua proiezione sul piano d'immagine di una ideale *Pinhole Camera*. Il foro (considerato idealmente come un punto) è l'unico sistema che ha la macchina di mettere a fuoco la luce senza l'ausilio di lenti. Quindi l'idea che sta dietro a questo studio è che i raggi di luce passino attraverso il centro ottico dell'obbiettivo senza essere distorti dalla lente della fotocamera e che continuino il loro percorso fino a intersecare il FPA. Essendo un modello ideale non tiene conto dei disturbi e del fatto che molte fotocamere hanno solo coordinate discrete e non continue come nella realtà e quindi è possibile considerarlo solo come un'approssimazione dell'oggetto reale. Più la qualità è alta della fotocamera, più questi effetti possono essere trascurati. Geometricamente questo modello lo si può vedere come in *figura 15*, la quale rappresenta la mappatura bidimensionale nel piano dell'immagine di un oggetto tridimensionale, utilizzando una *pinhole camera*. La descrizione dell'immagine sarà utile per capire il comportamento di una fotocamera di questo tipo:

- Il centro  $O$  del sistema di riferimento tridimensionale è la coordinata in cui si trova il foro della fotocamera, il quale essendo ideale viene approssimato a punto. In letteratura viene definito come *centro ottico*.

---

<sup>34</sup> È il piano che è ortogonale all'asse ottico della fotocamera.

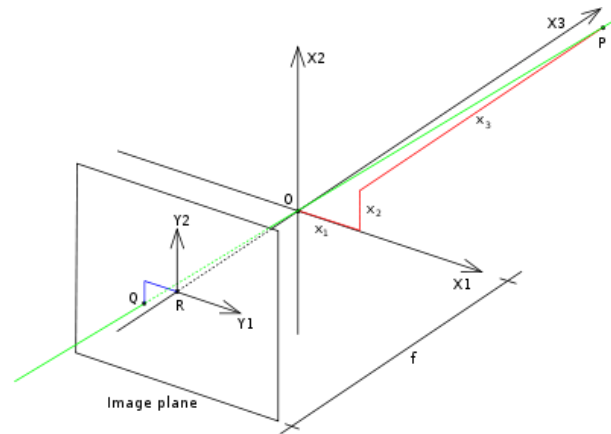


Figura 15: Schema Pinhole Camera Model. (fonte[1])

- L'asse  $X_3$  punta in direzione del corpo celeste o del soggetto in generale, quindi nella direzione dello spazio 3D. È definito come *asse ottico*.
- Il piano  $X_1, X_2$  è il lato anteriore della fotocamera e viene chiamato come *piano principale*.
- Parallelo al piano  $X_1, X_2$  a una distanza  $-f$  dal centro  $O$  del sistema di riferimento principale, si trova il *piano d'immagine* o *image plane*. La distanza è  $f > 0$  e quindi il piano interseca l'asse  $X_3$  nella sua parte negativa.
- La  $f$  è definita come *lunghezza focale* o *distanza focale*, che per le fotocamere che al posto dell'obiettivo hanno un foro, le *pinhole camera*, viene definita come la distanza che c'è tra il foro (la parte ottica della fotocamera) e il materiale sensibile alla luce FPA. Più la *distanza focale* è grande più si è in grado di fare ingrandimenti a soggetti molto distanti, come accade nella telescopio.
- L'origine del sistema di coordinate dell'*image plane* è  $R$ , che è anche il punto in cui l'asse  $X_3$  interseca il piano. Questo punto è definito come *punto principale*.
- Il punto  $P = P(x_1, x_2, x_3)$  in coordinate 3D è la posizione in cui si trova il corpo celeste rispetto al *foro ottico* della fotocamera.
- La linea verde è la *congiungente* o *linea di proiezione* del punto  $P$  che passa attraverso  $O$  e va a intersecare il *piano dell'immagine*, interno alla fotocamera, nel punto  $Q = Q(y_1, y_2)$ .

- L'*image plane* ha un suo sistema di riferimento d'origine  $R$  con assi  $Y_1, Y_2$  paralleli a  $X_1, X_2$ .

La relazione delle coordinate dell'immagine 3D a quella 2D la si può analizzare meglio guardando la *figura 15* vista dall'alto e con senso opposto al verso di  $X_2$  come mostrato in *figura 16*:

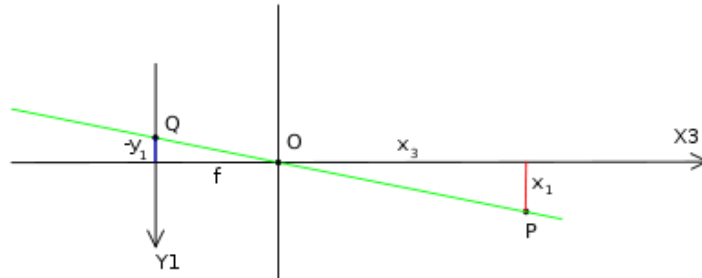


Figura 16: Schema Pinhole Camera Model visto dall'alto. (fonte[l])

Il legame tra  $Q = Q(y_1, y_2)$  e  $P = P(x_1, x_2, x_3)$  come si può vedere è dato dal fatto che la *linea di proiezione* che l'interpola (quella verde) fa da ipotenusa a i due triangoli che hanno per vertice  $Q$  e  $P$  rispettivamente. Nel triangolo di sinistra i cateti sono  $f$  (*distanza focale*) e  $-y_1$  mentre in quello di destra sono  $-x_1$  e  $x_3$ . Essendo due triangoli simili è possibile usare la relazione:

$$\frac{-y_1}{f} = \frac{x_1}{x_3} \quad (20)$$

ora facendo lo stesso ma guardando l'immagine nella direzione negativa di  $X_1$ , si trova una relazione simile alla (20):

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \quad (21)$$

da qui è possibile ricavare  $y_1$  e  $y_2$  come:

$$y_1 = -\frac{fx_1}{x_3} \quad (22)$$

$$y_2 = -\frac{fx_2}{x_3} \quad (23)$$

le quali sono le coordinate del punto  $Q = Q(y_1, y_2)$  nel *piano dell'immagine*.

Come già accennato all'inizio di questo Paragrafo una macchina fotografica *pinhole*, cattura l'immagine ruotata di  $180^\circ$  e la sua dimensione dipende dalla *distanza focale* dal *piano dell'immagine*. Per superare il problema della rotazione e quindi avere una mappatura di coordinate da 3D a 2D che sia utile, è possibile utilizzare due metodi: ruotare il sistema

di coordinate di  $180^\circ$ , cambiando sia la direzione di  $Y_1$  che quella di  $Y_2$  oppure spostare il *piano dell'immagine* a  $f$  invece di  $-f$  mettendolo quindi nella parte positiva di  $X_3$  ottenendo così un cambio di segno nelle coordinate  $y_1$  e  $y_2$ . Questo a livello teorico è possibile e semplifica il *metodo pinhole*, ma nella realtà no, perché non si può avere l'*image plane* davanti al *punto focale*. In questa tesi verrà trattato il *metodo pinhole* con il *piano dell'immagine* davanti alla parte ottica della fotocamera come appresentato in *figura 17*:

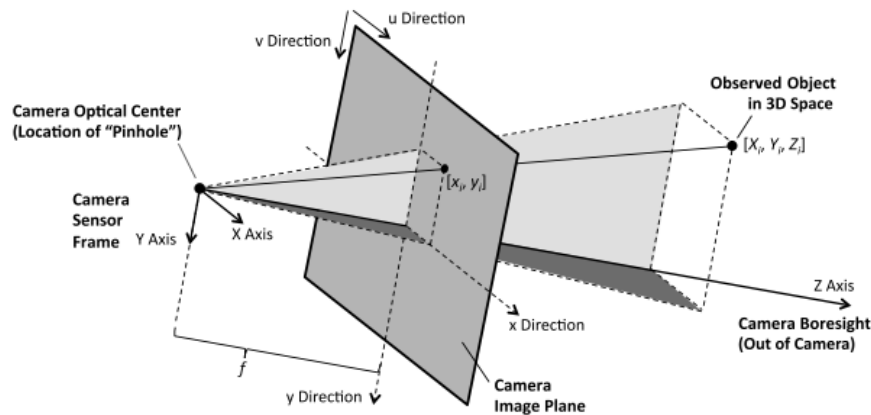


Figura 17: Geometria del Pinhole Camera Model. (fonte[m])

Le coordinate del *piano d'immagine* di un generico punto verranno da ora così definite:

$$[x_i, y_i]$$

Mentre le coordinate dell'oggetto in analisi 3D, il cui sistema di riferimento ha origine nel *piano focale*, sono definite come segue:

$$[X_i, Y_i, Z_i]$$

dalla (22) e (23) è possibile riscrivere la relazione tra i due sistemi di riferimento come:

$$x_i = \frac{fX_i}{Z_i} \quad (24)$$

$$y_i = \frac{fY_i}{Z_i} \quad (25)$$

#### 4.2.2 ELLISSOIDE DI RIFERIMENTO TRIASSIALE

L'ellissoide di riferimento è una superficie che viene definita matematicamente e grazie alla sua forma può approssimare quella reale

della Terra o di altri corpi celesti. Spesso viene usata in situazioni cui si devono utilizzare come coordinate latitudine, longitudine e altitudine. Per studiare corpi celesti minori, come lune o asteroidi viene utilizzato un *ellissoide triassiale* o *ellissoide scaleno*, che a differenza di quello usato per studiare i Pianeti, ha i tre assi  $a$ ,  $b$  e  $c$  con dimensioni differenti tra loro.

### 4.3 RELAZIONE TRA ELLISSOIDE TRIASSIALE E LA SUA PROIEZIONE NEL PIANO DELL'IMMAGINE

L'immagine catturata di un ellissoide di riferimento, se la visuale è "completa", nel piano dell'immagine 2D risulta essere un'ellisse. Ovviamente a seconda della distanza tra la fotocamera e l'oggetto e l'orientamento tra i due, l'immagine proiettata non rispecchia sempre la forma simile a quella reale e a volte può essere fuorviante avvalersi di questa immagine per studiare la posizione della fotocamera. Quindi non in tutti i casi è di buon utilizzo la tecnica basata sull'*ellipse fitting*, ma bisogna valutare caso per caso per via della non linearità del problema. I problemi trattati in questa tesi faranno uso di coni e sezioni coniche. La sezione della conica è descritta tramite l'Eq. (13):

$$Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Fy_i + G = 0$$

il risultato è un'ellisse se e solo se vale la relazione:

$$4AC > B^2$$

Sostituendo in (13), le coordinate ottenute tramite il *pinhole camera method*, quindi (24) e (25), è possibile relazionare la sezione 2D con la superficie conica reale su cui si trovano i punti 3D, tramite l'Eq. (26) (quanto segue è scritto come le coordinate del sistema di riferimento che ha per origine il punto focale):

$$A \left( \frac{fX_i}{Z_i} \right)^2 + B \left( \frac{fX_i}{Z_i} \right) \left( \frac{fY_i}{Z_i} \right) + C \left( \frac{fY_i}{Z_i} \right)^2 + D \left( \frac{fX_i}{Z_i} \right) + F \left( \frac{fY_i}{Z_i} \right) + G = 0 \quad (26)$$

rielaborando il tutto si ottiene:

$$Af^2X_i^2 + Bf^2X_iY_i + Cf^2Y_i^2 + DfX_iZ_i + FfY_iZ_i + GZ_i^2 = 0 \quad (26)$$

Come fatto in precedenza con l'equazione dell'ellisse 2D, verrà trasformata in forma matriciale:

$$\begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix} \begin{bmatrix} Af^2 & Bf^2/2 & Df/2 \\ Bf^2/2 & Cf^2 & Ff/2 \\ Df/2 & Ff/2 & G \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = 0 \quad (27)$$

e definendo:

$$s_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (28)$$

$$C = \begin{bmatrix} Af^2 & Bf^2/2 & Df/2 \\ Bf^2/2 & Cf^2 & Ff/2 \\ Df/2 & Ff/2 & G \end{bmatrix} \quad (29)$$

quindi riscrivendo tutto in maniera più semplice, si ottiene l'equazione della superficie conica:

$$s_i^T C s_i = 0. \quad (30)$$

Infine è possibile dire che l'ellisse che viene a formarsi nel *piano dell'immagine* è data dall'intersezione del piano stesso con il cono d'Eq. (30) com'è possibile vedere in *figura 18*:

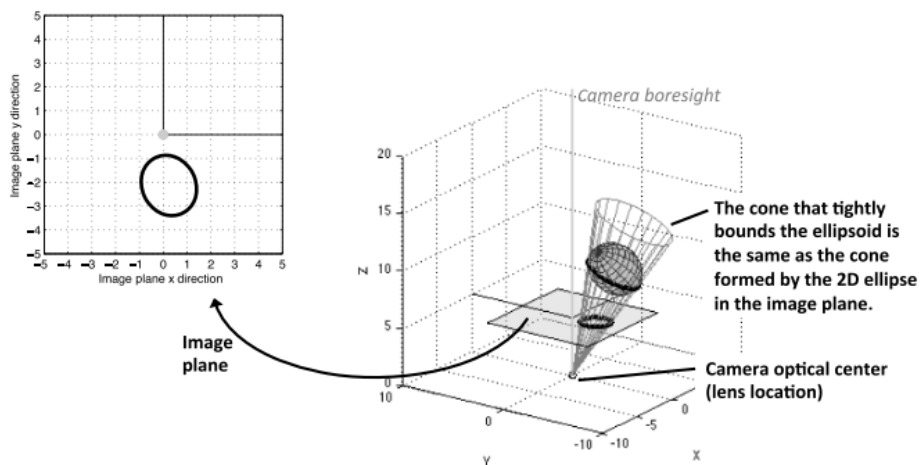


Figura 18: Il cono 3D che delimita l'ellissoide triassiale è lo stesso che proietta nel piano dell'immagine l'ellisse 2D. (fonte[m])

Se si suppone che il corpo celeste può essere identificato come un ellissoide triassiale, si avrà che ogni  $i$  – *esimo* punto che si trova sulla sua superficie, *horizon points*<sup>35</sup>, può essere descritto rispetto al sistema di riferimento del pianeta come:

$$p_i^T \begin{bmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{bmatrix} p_i = p_i^T A_P p_i = 1 \quad (31)$$

dove  $a, b$  e  $c$  sono la metà dei semiassi principali dell'ellissoide, mentre  $p_i$  è un punto generico sulla superficie del Pianeta. Si trasformano ora le coordinate di ognuno di questi punti tramite una matrice di rotazione rispetto al sistema di riferimento che ha origine nel punto nel *punto focale*

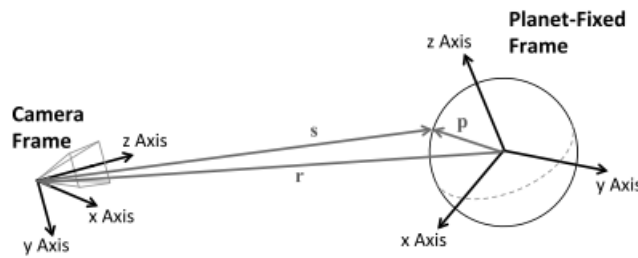


Figura 19: Rappresenta la geometria e come i sistemi di riferimento vedono i punti sulla superficie del pianeta. (fonte[m])

(vedi figura 19):

$$p_i^T T_C^P A_P T_P^C p_i = p_i^T A p_i = 1. \quad (32)$$

La matrice di rotazione dev'essere costruita in base al comportamento della sonda e in base all'effemeridi del corpo celeste. Per semplificare l'annotazione da ora in poi si supporrà, che le coordinate siano intese come calcolate rispetto al *camera frame*, verranno di seguito riscritti il vettore posizione e la matrice come:

$$p_i = T_P^C p_i$$

$$A = T_C^P A_P T_P^C$$

<sup>35</sup>Sono i punti che sono tangenti alla superficie dell'ellissoide triassiale, quando è delimitato dal cono 3D. Vedi figura 20



Come si può notare in *figura 19* il vettore  $s_i$  è quello di posizione che parte dal *punto focale* fino a intersecare la superficie del corpo celeste, per costruzione vettoriale è possibile scriverlo come:

$$s_i = p_i - r \quad (33)$$

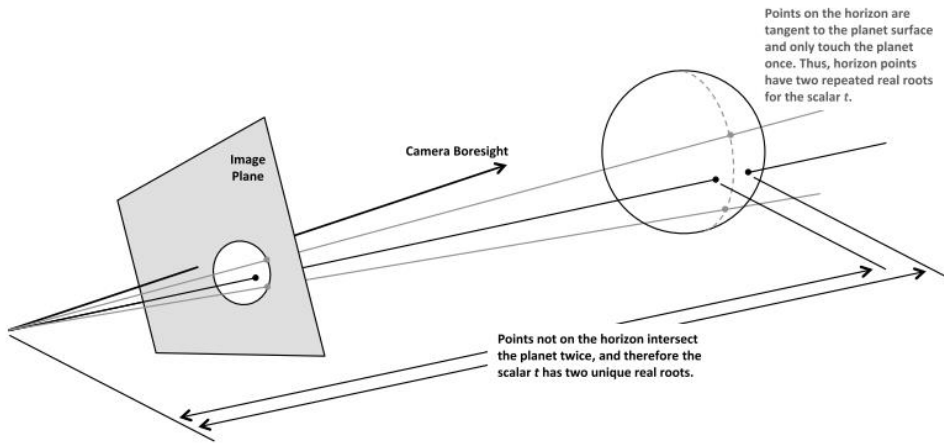
dove  $r$  è il vettore che parte dall'origine del *camera frame* e indica la posizione dell'origine del sistema di riferimento dell'ellissoide triassiale. Se si suppone che il versore LOS lungo  $s_i$  sia dato dalla definizione:

$$e_i = \frac{s_i}{\|s_i\|}$$

il vettore posizione può essere annotato come:

$$s_i = p_i - r = te_i \quad (34)$$

dove  $t$  è la distanza lungo la direzione  $e_i$  prima che il raggio intersechi la superficie del corpo celeste come è possibile osservare in *figura 20*:



*Figura 20: Immagine esplicativa riguardo il significa degli horizon points. (fonte[m])*

riscrivendo le coordinate del punto  $p_i$  in funzione della direzione  $e_i$  risulta:

$$p_i = te_i + r \quad (35)$$

e sostituendo il tutto in (32) si ottiene:

$$(te_i + r)_i^T A (te_i + r)_i = 1 \quad (36)$$

usando la simmetria della matrice  $A$ , è possibile riscrivere l'equazione come una funzione di secondo grado con variabile  $t$ :

$$(e_i^T A e_i)t^2 + 2(e_i^T A r)t + (r^T A r - 1) = 0 \quad (37)$$

Come per tutte le equazioni di secondo grado si possono verificare tre casi:

$$\Delta = 4(e_i^T A r)^2 - 4(e_i^T A e_i)(r^T A r - 1) \quad (38)$$

se

- $\Delta < 0$ , non ci sono soluzioni reali, quindi il LOS non interseca la superficie.

- $\Delta > 0$ , esistono due soluzioni reali e distinte, questo significa che il raggio è secante rispetto al corpo celeste e non identifica un punto nel bordo, quindi non è d'interesse per i fini di questa tesi.
- $\Delta = 0$  esistono due soluzioni reali e coincidenti che identificano un horizon point, questo significa che il raggio è tangente alla superficie.

Dalla (38) essendo un'equazione scalare di secondo grado in  $e_i$ , è possibile esplicitarla e imporla uguale a zero per trovare i punti interesse, cioè quelli che appartengono alla superficie dell'ellissoide triassiale di riferimento, raccogliendo si ottiene:

$$e_i^T [Arr^T A - (r^T Ar - 1)A] e_i = 0 \quad (39)$$

definendo ora la matrice simmetrica  $M_{3 \times 3}$  come:

$$M = Arr^T A - (r^T Ar - 1)A. \quad (40)$$

Sostituendo ora questa matrice nell'Eq. (39) e moltiplicando per  $t^2$ , si ottiene:

$$s_i^T M s_i = 0. \quad (41)$$

L'equazione trovata descrive un cono i cui punti che si trovano sulla superficie sono rappresentati da  $s_i$  ed è possibile osservare che è uguale all'Eq. (30) a meno dei parametri  $A$  e  $r$  che dipendono dalla forma del corpo celeste, dall'effemeridi e dal comportamento della sonda. Eguagliando le due equazioni (30) e (41):

$$s_i^T C s_i = s_i^T M s_i \quad (42)$$

è possibile dedurre che innanzitutto descrivono lo stesso cono in maniera differente, la prima tramite i coefficienti dell'Eq. (26) e la seconda usando i parametri  $A$  e  $r$ . Detto questo si può concludere dicendo che il cono d'equazione (30) che genera l'ellisse nel *piano dell'immagine* per via dell'intersezione del piano stesso con la superficie conica è lo stesso che delimita l'ellissoide triassiale come in *figura 18*, questo significa che la proiezione dell'ellissoide di riferimento scaleno (3D) nell'*image plane* è l'ellisse che si ottiene dalla superficie conica e ha per centroide:

$$x_i = \frac{fX_i}{Z_i}$$

$$y_i = \frac{fY_i}{Z_i}$$

queste coordinate verranno poi convertite dal calcolatore in pixel.

## 4.4 METODI DI REGRESSIONE AD ELLISSE

In questa tesi sono state prese delle immagini raffiguranti MIMAS ed ENCELADO i due satelliti naturali di Saturno, i quali hanno *proiettato* un'ellisse 2D nel FPA al momento dello scatto della foto da parte della camera presente sulla sonda Cassini. In tutte queste immagini non è possibile a vedere per intero la forma del corpo celeste, in realtà nella maggior parte dell'immagini campione utilizzate, se non si effettua un'operazione di identificazione del limbo (*edge detection*), non si è in grado di notare nulla, perché appaiono buie. L'obbiettivo che questa tesi si è imposta, come già ripetuto in precedenza, è quello di fittare il limbo del corpo individuato tramite il metodo Canny, in cui si è ampiamente parlato nel Capitolo 3.3 e ottenere un'ellisse che approssimi la forma reale del corpo e dopodiché calcolarne il suo centroide. Una delle tecniche su cui si basano la maggior parte dei metodi di regressione ad ellisse è quella dei minimi quadrati di cui verrà discusso nei Paragrafi successivi.

### 4.4.1 MINIMI QUADRATI FORMULAZIONE GENERALE

È una tecnica teorica d'ottimizzazione che permette di trovare una funzione, detta *curva di regressione*<sup>36</sup>, che si avvicini il più possibile all'insieme di punti che giacciono sul piano. La funzione trovata dev'essere quella che minimizzi la somma dei quadrati della distanza tra i punti dati (o ottenuti tramite *edge detection*) e quelli della curva di regressione.

La formulazione generale del problema è la seguente: dati i punti  $\bar{x}_i = (x_i, y_i)$ , lo scopo è trovare una funzione  $f$  tale che approssimi la successione dei punti prima definiti. Questa funzione è possibile determinarla minimizzando la distanza euclidea tra le due successioni  $y_i$  e  $f(x_i)$ , ovvero la quantità:

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (43)$$

da cui il nome dei minimi quadrati.

---

<sup>36</sup> Questa curva, usualmente una conica, è anche detta *curva ottima*.

Esistono due tipi di soluzioni del sistema, quella lineare e quella non lineare, quest'ultima la si può trovare solo tramite soluzioni di tipo numerico (metodo iterativo) e non analitico.

#### 4.4.1.1 MINIMI QUADRATI: SOLUZIONE LINEARE

Risolvere il problema dei minimi quadrati lineare equivale a trovare una soluzione di un sistema di tipo lineare sovradimensionato, cioè con un numero maggiore d'equazioni rispetto alle incognite. Inoltre può essere visto come individuare l'unico minimo presente di una parabola, cioè che la soluzione è unica.

La soluzione lineare è possibile trovarla risolvendo un sistema di tipo lineare, data una funzione  $f(x)$  lineare rispetto i  $k$  parametri  $p_i$ ,  $k$  è molto più piccolo rispetto gli  $n$  punti dati, è possibile scrivere la funzione come:

$$f(x) = p_1 f_1(x) + p_2 f_2(x) + \dots + p_k f_k(x) \quad (44)$$

definendo:

$$A = \begin{bmatrix} f_1(x_1) & \dots & f_k(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_n) & \dots & f_k(x_n) \end{bmatrix}$$

$$p = \begin{bmatrix} p_1 \\ \vdots \\ p_k \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

è possibile riscrivere la (44) in forma matriciale come:

$$Ap = y \quad (45)$$

Il problema di minimizzare  $S$  si riconduce a minimizzare la norma del residuo:

$$\|r\| = \|Ap - y\| \quad (46)$$

elevando al quadrato si ottiene:

$$\begin{aligned} \|r\|^2 &= \|Ap - y\|^2 = ([Ap]_1 - y_1)^2 + \dots + ([Ap]_n - y_n)^2 \\ &= \sum_{i=1}^n (y_i - f(x_i))^2 = S \end{aligned} \quad (47)$$

dove con  $[Ap]_i$  s'intende la  $i$ -esima componente del vettore  $Ap$ . Per minimizzare  $\|r\|$  si può derivare la funzione  $S$  rispetto ogni  $p_m$  e imporla uguale a zero:

$$\frac{dS}{dp_m} = \sum_{i=1}^n 2 \sum_{j=1}^k (a_{ij}p_j - y_i) a_{im} = 0. \quad (48)$$

È possibile riscrivere queste equazioni nel sistema:

$$(Ap - y)^T A = 0 \quad (49)$$

da qui si può ricavare il vettore  $p$  che minimizza l'equazione (49):

$$p = (A^T A)^{-1} A^T y \quad (50)$$

NB: questo è possibile se la matrice  $A$  ha rango completo e quindi  $A^T A$  è invertibile.

#### 4.4.1.2 MINIMI QUADRATI: SOLUZIONE NON LINEARE

La risoluzione di un problema dei minimi quadrati non lineare è di una maggiore difficoltà rispetto a quello lineare, questo perché non si ha un solo minimo, ma sono presenti tanti minimi locali, quindi tante soluzioni, è perciò importante avere ottime condizioni iniziali per poter identificare il minimo cercato.

Per trovare la soluzione è possibile usare il metodo Gauss-Newton il quale ci dice che, dato un vettore  $\mathbf{u} = (u_1, \dots, u_n)^T$  di  $n$  incognite e dato un sistema non lineare di  $m$  equazioni, si ottiene:

$$f(\mathbf{u}) = 0 \quad (51)$$

Se  $m > n$  il sistema è sovradeterminato ed è possibile minimizzare la distanza euclidea:

$$\sum_{i=1}^m f_i(\mathbf{u})^2 = \min \quad (52)$$

Per risolvere quest'equazione è necessario utilizzare una sequenza di soluzioni lineari dei minimi quadrati, le quali verranno iterate per restituire una soluzione di tipo numerico del sistema non lineare.

Si approssima la soluzione come  $\mathbf{u} = \tilde{\mathbf{u}} + h$ , e sviluppando:

$$f(\mathbf{u}) = (f_1(\mathbf{u}), \dots, f_m(\mathbf{u}))^T \quad (53)$$

in serie di Taylor attorno  $\tilde{\mathbf{u}}$  si ottiene:

$$f(\tilde{\mathbf{u}} + h) \cong f(\tilde{\mathbf{u}}) + J(\tilde{\mathbf{u}})h \cong 0 \quad (54)$$

dove  $J$  è lo Jacobiano. Da qui è possibile risolvere l'equazione con il metodo dei minimi quadrati rispetto al vettore di correzione  $h$ :

$$J(\tilde{\mathbf{u}})h \cong -f(\tilde{\mathbf{u}}) \quad (55)$$

Il metodo di Gauss-Newton consiste quindi nella risoluzione rispetto  $h$  dell'equazione e aggiornare la correzione nell'equazione  $\mathbf{u} = \tilde{\mathbf{u}} + h$ .

#### 4.4.2 DETERMINAZIONE DELL'ELLISSE CHE MINIMIZZA LA DISTANZA ALGEBRICA (MDA)

Definendo la matrice  $A_{2 \times 2}$  simmetrica positiva<sup>37</sup>:

$$A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

e i vettori:

$$\bar{x} = (x, y)$$

$$b = (b_1, b_2)$$

dove  $\bar{x}$  è il vettore delle coordinate di un generico punto  $P$ .

Partendo dall'Eq. (13) dell'ellisse, si ridefiniscono i coefficienti con le seguenti notazioni:

$$a_{11} = A; a_{21} + a_{12} = B; a_{22} = C; b_1 = D; b_2 = F; c = G$$

ora è possibile riscrivere in forma matriciale l'equazione come:

$$\bar{x}^T A \bar{x} + b^T \bar{x} + c = 0 \quad (56)$$

Introducendo la nuova coordinata  $\tilde{X} = (\tilde{x}_1, \tilde{x}_2)$  con:

$$\bar{x} = Q \tilde{X} + \bar{t}$$

si ha una rotazione e traslazione della conica. La nuova equazione diventa quindi:

$$\tilde{X}^T (Q^T A Q) \tilde{X} + (2\bar{t}A + b^T) Q \tilde{X} + \bar{t}^T A \bar{t} + b^T \bar{t} + c = 0 \quad (57)$$

dove per una maggior semplicità di lettura verranno definiti i nuovi coefficienti come:

$$\bar{A} = (Q^T A Q)$$

$$\bar{b} = (2\bar{t}A + b^T) Q$$

$$\bar{c} = \bar{t}^T A \bar{t} + b^T \bar{t} + c$$

Sostituendo all'Eq. (57) si ottiene:

$$\tilde{X}^T \bar{A} \tilde{X} + \bar{b}^T \tilde{X} + \bar{c} = 0 \quad (58)$$

---

<sup>37</sup> Si definisce simmetrica positiva la matrice che per ogni vettore  $\bar{x} = \begin{pmatrix} x_1 \\ x_i \\ x_n \end{pmatrix}$  il prodotto

$\bar{x}^T A \bar{x}$  restituisce uno scalare maggiore di zero

la matrice  $Q$  dev'essere scelta in maniera tale che la matrice  $\bar{A}$  abbia sulla diagonale gli autovalori<sup>38</sup>  $\lambda_1$  e  $\lambda_2$ , inoltre si imponga un  $\bar{c}$  tale da ottenere  $\bar{b} = 0$ . Il risultato che ne deriva è un'ellisse:

$$\lambda_1 \tilde{x}_1^2 + \lambda_2 \tilde{x}_2^2 + \bar{c} = 0 \quad (59)$$

se e solo se i coefficienti dell'Eq. (59) assumono i seguenti valori:

$$\lambda_1 > 0$$

$$\lambda_2 > 0$$

$$\bar{c} < 0$$

Come fatto in precedenza si possono calcolare anche qui le caratteristiche principali dell'ellisse che si ottenuta tramite le seguenti relazioni:

$$\bar{z} = (t_1, t_2) \quad (60)$$

$$\boxed{a} = \sqrt{\frac{-\bar{c}}{\lambda_1}} \quad (61)$$

$$\boxed{b} = \sqrt{\frac{-\bar{c}}{\lambda_2}} \quad (62)$$

Sapendo che  $Q^T Q = I$ , si deduce che  $A$  e  $\bar{A}$  hanno gli stessi autovalori  $\lambda_1$  e  $\lambda_2$ , questo implica che ogni funzione di questi due autovalori è invariante in traslazione e rotazione.

Definendo:

$$\det A = a_{11}a_{22} - a_{21}a_{12} = \lambda_1\lambda_2 \quad (63)$$

$$\text{traccia } A = a_{11} + a_{22} = \lambda_1 + \lambda_2 = \text{tr } A \quad (64)$$

e i parametri:

$$k = \frac{\boxed{a}}{\boxed{b}} \quad (65)$$

$$\boxed{a}^2 = -\frac{\bar{c}}{\lambda_1} \quad (66)$$

$$\boxed{b}^2 = -\frac{\bar{c}}{\lambda_2} \quad (67)$$

Si può ottenere:

---

<sup>38</sup> Presa una funzione lineare  $F: V \rightarrow V$  con  $V$  spazio vettoriale sul campo Reale, diremo che  $\lambda$  (numero Complesso) è un autovalore di  $F$  se esiste un vettore non nullo  $\bar{v}$  tale che  $F(\bar{v}) = \lambda\bar{v}$ . Diremo quindi che l'autovettore  $\bar{v}$  che è associato all'autovalore  $\lambda$  avrà come sua immagine  $F$  ma con modulo differente. Si può dire che il vettore  $\bar{v}$ , autovettore, manterrà la propria direzione e cambierà solo in modulo.

$$k^2 + \frac{1}{k^2} = \frac{\lambda_2}{\lambda_1} + \frac{\lambda_1}{\lambda_2} = \frac{\lambda_1^2 + \lambda_2^2}{\lambda_1 \lambda_2} = \frac{(\text{tr } A)^2 - 2 \det A}{\det A} = \frac{a_{11}^2 + a_{22}^2 + 2a_{12}^2}{a_{11}a_{22} - a_{12}^2}. \quad (68)$$

riordinando il membro di destra e ridefinendolo per semplicità come:

$$\mu = \frac{(\text{tr } A)^2}{2 \det A} - 1 \quad (69)$$

si può riscrivere la (68) come:

$$k^2 = \mu \mp \sqrt{\mu^2 - 1} \quad (70)$$

Dati dei punti, ad esempio ottenuti tramite *edge detection*, è possibile calcolare il vettore dei coefficienti  $\mathbf{u}$  che individuano l'ellisse di fitting, sostituendoli nell'Eq. (56), ottenendo infine un sistema lineare:

$$B\mathbf{u} = 0 \quad (71)$$

dove  $\mathbf{u}$  uguale a:

$$\mathbf{u} = (a_{11}, a_{12}, a_{22}, a_{21}, b_1, b_2, c)^T. \quad (72)$$

Si può concludere dicendo che la determinazione dell'ellisse che minimizza la distanza algebrica, significa risolvere un sistema di tipo lineare, in cui tramite l'applicazione di alcuni vincoli, ad esempio  $\|\mathbf{u}\| = 1$  per trovare  $\|B\mathbf{u}\| = \min$  è possibile ricavare l'equazione dell'ellisse. Questo metodo non è di difficile applicazione ma risulta in molti casi un metodo molto approssimativo, quindi non restituisce risultati precisi.

#### 4.4.3 DETERMINAZIONE DELL'ELLISSE DI APPROSSIMAZIONE IN FORMA PARAMETRICA (MDG)<sup>39</sup>

Trova l'ellisse d'approssimazione in forma parametrica, significa risolvere un problema di tipo non lineare e di una maggiore difficoltà rispetto a quello visto precedentemente. Questo metodo infatti necessita di condizioni iniziali, le quali possono essere ottenute tramite la determinazione dell'ellisse che minimizza la distanza algebrica. Una delle principali difficoltà che si riscontra nella risoluzione di questo problema è quello d'imporre le corrette condizioni iniziali, perché se queste sono errate o non precise il risultato rischia di divergere.

---

<sup>39</sup> Questo metodo lo si può anche definire come quello che minimizza la distanza geometrica dei punti.



Supponendo d'imporre le giuste condizioni iniziali, è possibile risolvere tramite iterazioni questo tipo di problema e infine ottenere l'equazione parametrica dell'ellisse che fitta i punti ottenuti con il metodo Canny.

Si consideri l'equazione:

$$\bar{x} = \bar{z} + Q(\alpha)\bar{x}' \quad (73)$$

dove:

$$\begin{aligned} \bar{x} &= \begin{pmatrix} x_i \\ y_i \end{pmatrix} \\ \bar{x}' &= \begin{pmatrix} a \cos \varphi \\ b \sin \varphi \end{pmatrix} \\ Q(\alpha) &= \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}. \end{aligned}$$

Minimizzare la distanza della somma dei minimi quadrati degli  $m$  punti ottenuti è equivalente a risolvere il problema non lineare dei minimi quadrati:

$$g_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - Q(\alpha) \begin{pmatrix} a \cos \varphi_i \\ b \sin \varphi_i \end{pmatrix} \approx 0 \quad (74)$$

con  $i = 1, \dots, m$ . Questo è come avere un sistema non lineare di  $2m$  equazioni in  $m$  incognite  $\varphi_1, \dots, \varphi_m$  più  $\alpha, a, b, z_1$  e  $z_2$ . Per calcolare lo Jacobiano, serve conoscere le derivate parziali della funzione  $g_i$ :

$$\begin{aligned} \frac{\partial g_i}{\partial \varphi_j} &= -\delta_{ij} Q(\alpha) \begin{pmatrix} -a \sin \varphi_i \\ b \cos \varphi_i \end{pmatrix} \\ \frac{\partial g_i}{\partial \alpha} &= -\dot{Q}(\alpha) \begin{pmatrix} a \cos \varphi_i \\ b \sin \varphi_i \end{pmatrix} \\ \frac{\partial g_i}{\partial a} &= -Q(\alpha) \begin{pmatrix} \cos \varphi_i \\ 0 \end{pmatrix} \\ \frac{\partial g_i}{\partial b} &= -Q(\alpha) \begin{pmatrix} 0 \\ \sin \varphi_i \end{pmatrix} \\ \frac{\partial g_i}{\partial z_1} &= \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\ \frac{\partial g_i}{\partial z_2} &= \begin{pmatrix} 0 \\ -1 \end{pmatrix} \end{aligned}$$

dove  $\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases}$ .

Lo Jacobiano diventa:

$$J = \begin{pmatrix} -Q(\alpha) \begin{pmatrix} -a \sin \varphi_1 \\ b \cos \varphi_1 \end{pmatrix} & -\dot{Q}(\alpha) \begin{pmatrix} a \cos \varphi_1 \\ b \sin \varphi_1 \end{pmatrix} & -Q(\alpha) \begin{pmatrix} \cos \varphi_1 \\ 0 \end{pmatrix} & -Q(\alpha) \begin{pmatrix} 0 \\ \sin \varphi_1 \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -Q(\alpha) \begin{pmatrix} -a \sin \varphi_m \\ b \cos \varphi_m \end{pmatrix} & -\dot{Q}(\alpha) \begin{pmatrix} a \cos \varphi_m \\ b \sin \varphi_m \end{pmatrix} & -Q(\alpha) \begin{pmatrix} \cos \varphi_m \\ 0 \end{pmatrix} & -Q(\alpha) \begin{pmatrix} 0 \\ \sin \varphi_m \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \end{pmatrix} \quad (75)$$

Sapendo che la matrice  $Q$  di rotazione è ortogonale e definendo la matrice  $U = -diag(Q, \dots, Q)$  sempre ortogonale è possibile riscrivere la matrice Jacobiana come:

$$U^T J = \begin{pmatrix} \begin{pmatrix} -a \sin \varphi_1 \\ b \cos \varphi_1 \end{pmatrix} & & \begin{pmatrix} a \cos \varphi_1 \\ b \sin \varphi_1 \end{pmatrix} & \begin{pmatrix} \cos \varphi_1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ \sin \varphi_1 \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} -a \sin \varphi_m \\ b \cos \varphi_m \end{pmatrix} & & \begin{pmatrix} a \cos \varphi_m \\ b \sin \varphi_m \end{pmatrix} & \begin{pmatrix} \cos \varphi_m \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ \sin \varphi_m \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \end{pmatrix} \quad (76)$$

Definendo:

$$S = diag(\sin \varphi_i)$$

$$C = diag(\cos \varphi_i)$$

come due matrici  $m \times m$  diagonali, e ponendo:

$$A(i, 1:5) = [-b \sin \varphi_i \quad \cos \varphi_i \quad 0 \quad \cos \alpha \quad \sin \alpha]$$

$$B(i, 1:5) = [a \cos \varphi_i \quad 0 \quad \sin \varphi_i \quad -\sin \alpha \quad \cos \alpha]$$

è possibile riscrivere lo Jacobiano in una forma più leggibile:

$$J = \begin{pmatrix} -aS & A \\ -bC & B \end{pmatrix} \quad (77)$$

Imponendo ora le giuste condizioni iniziali dei valori e iterando il procedimento è possibile ottenere con precisione un'ellisse che approssimi al meglio il corpo in analisi. Questa tipologia di studio ha un'accuratezza maggiore rispetto alla risoluzione algebrica del problema, imponendo  $\|u\| = 1$ , come si può osservare in *figura 21*:

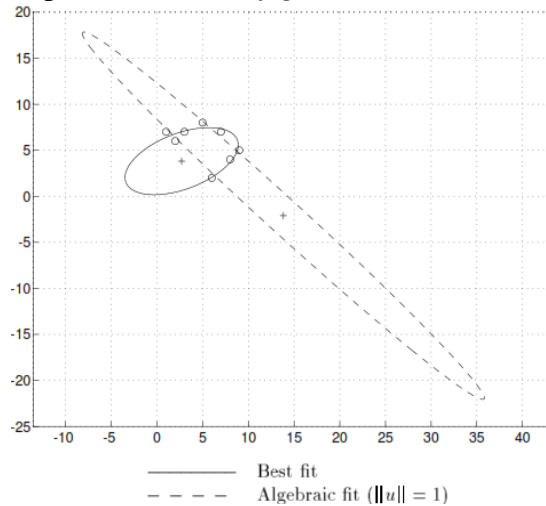


Figura 21:Mostra la differenza di precisione tra un fitting di tipo algebrico e uno di tipo parametrico

## 4.5 ALGELLIPSE<sup>40</sup> (MDA)

<sup>40</sup> Queste funzioni sono state prese da [2]

La prima funzione d'approssimazione utilizzata nel software è *algellipse*, la quale si basa sul minimizzare la distanza algebrica nel senso dei minimi quadrati, imponendo la norma del vettore dei coefficienti uguale a uno:  $\|\mathbf{u}\| = 1$ .

Questo è come imporre:

$$\lambda_1 + \lambda_2 = 1$$

Con questo vincolo il vettore non è invariante sotto trasformazione euclidea, questo quindi comporta che fissando i punti da approssimare, l'ellisse che si ottiene se questi vengo ruotati è differente rispetto a quella che si otterrebbe con i punti nella condizione iniziale, ed è per questo stato creato il metodo *Bookstein*.

#### 4.6 BOOKSTEIN<sup>41</sup> (MDA)

Questo metodo cerca sempre di minimizzare la distanza algebrica dai minimi quadrati, ma utilizza un vincolo differente rispetto ad *algellipse*. Definendo i vettori dei coefficienti in questo modo:

$$\mathbf{v} = (b_1, b_2, c)^T$$

$$\mathbf{w} = (a_{11}, \sqrt{a_{12}}, a_{22})^T$$

il vincolo imposto è:

$$\|\mathbf{w}\| = 1$$

che è come imporre:

$$\lambda_1^2 + \lambda_2^2 = 1$$

i quali rappresentano gli autovalori della matrice.

Il metodo risulta invariante sotto trasformazione euclidea.

#### 4.7 ELLIPSE DIRECT FIT<sup>42</sup> (MDA)

---

<sup>41</sup> Queste funzioni sono state prese da [2]

<sup>42</sup> <https://it.mathworks.com/matlabcentral/fileexchange/22684-ellipse-fit--direct-method->

È un'evoluzione di un precedente metodo d'*ellipse fitting* il quale era lento e richiedeva una grande capacità di calcolo, questo nuovo metodo invece si basa sul minimizzare la distanza algebrica, incorporando tramite la normalizzazione dei fattori il vincolo legato all'ellitticità:  $\Delta = 4AC - B^2 = 1$ . Questo metodo offre diversi vantaggi, per esempio anche i dati errati relativi alle specifiche dell'ellisse, fanno comunque in modo di restituire sempre e solo un'ellisse, inoltre è possibile risolvere il sistema utilizzando una generalizzazione di un "sistema di autovalori". Questo metodo è lineare, veloce e non iterativo.

#### 4.8 ELLIPSE FIT BY TAUBIN<sup>43</sup> (MDA)

Venne pubblicato nel 1991 da *Taubin*, è un metodo abbastanza veloce e non è di tipo iterativo, anche questo restituisce la norma vettore dei coefficienti della matrice uguale a uno. Questo metodo è di tipo lineare, quindi è veloce e non iterativo, si basa sul trovare i coefficienti dell'equazione che descrive l'ellisse che minimizza la distanza algebrica dai punti dati. Anche se è un metodo con un approccio di tipo algebrico, ha comunque una buona precisione e robustezza.

Nel caso che i coefficienti ottenuti siano relativi ad una parabola, questo metodo non approssima un'ellisse, come in quello visto in precedenza, ma ad una parabola. Nel caso in cui venga identificata un'ellisse il metodo utilizza *Ellipse Direct Fit*, ma con una qualità differente.

#### 4.9 FIT\_ELLIPSE<sup>44</sup> (MDG)

Questo metodo si basa sulla risoluzione non lineare del metodo dei minimi quadrati, quindi come soluzione restituirà i parametri fondamentali dell'ellisse d'approssimazione. Se il coefficiente  $B$  dell'equazione (13) è

---

<sup>43</sup> <https://it.mathworks.com/matlabcentral/fileexchange/22683-ellipse-fit--taubin-method->

<sup>44</sup> [https://it.mathworks.com/matlabcentral/fileexchange/3215-fit-ellipse?s\\_tid=srchtitle](https://it.mathworks.com/matlabcentral/fileexchange/3215-fit-ellipse?s_tid=srchtitle)

uguale a zero si ha un'ellisse con inclinazione non nulla, dopo la stima che viene effettuata da questa funzione, l'orientamento che si ottiene viene eliminato tramite una matrice di rotazione, i restanti parametri che descrivono l'ellisse vengono poi restituiti come output. Nel caso in cui i dati ottenuti sono relativi a un'iperbole o una parabola il vettore dei parametri risulterà nullo.

#### 4.10 FITELLIPSE<sup>45</sup> (MDA+MDG: è un sistema “ibrido”)

Questo metodo è il più completo di quelli visto fino ad ora, il primo step è quello di trovare le condizioni iniziali tramite la risoluzione lineare del metodo dei minimi quadrati, ad esempio imponendo il vincolo di *Bookstein*. Questa parte è di semplice risoluzione e ha un basso costo in termini di calcolo.

Determinate le condizioni iniziali la funzione procede con la risoluzione del problema non lineare dei minimi quadrati, quindi punta a trovare la minima distanza geometrica dai punti. Questo però è un metodo come già detto non lineare e più complesso, ed è possibile risolverlo con metodi iterativi.

Il risultato che si ottiene se la soluzione converge è molto preciso, quindi si ottiene un'ellisse in forma parametrica che minimizza la distanza geometrica. In caso contrario se diverge, il risultato non è del tutto da scartare, perché viene disegnata l'ellisse che minimizza la distanza algebrica che è stata usata come condizione iniziale.

---

<sup>45</sup> <https://it.mathworks.com/matlabcentral/fileexchange/15125-fitellipse-m>

## 5. IL PROGETTO

Come detto in precedenza l'obiettivo principale di questa tesi è stato quello della creazione di un software in Matlab, che fosse in grado innanzitutto di rilevare i bordi del corpo celeste da un'immagine, poi successivamente tramite l'ausilio di varie funzioni fittare il limbo, individuato senza la presenza del rumore dell'immagine, a un'ellisse. Fatto questo, oltre a capire quale delle funzioni ha una maggiore robustezza, efficienza e quale approssima al meglio la forma reale del corpo, il software dev'essere in grado d'individuare con precisione il centroide dell'ellisse.

### 5.1 CALIBRAZIONE IMMAGINI DELLA SONDA CASSINI

Il primo step fatto è stato quello di andare nel sito PDS [A] della NASA e scaricare le immagini scattate dalla sonda Cassini relative a ENCELADO e MIMAS, nel formato in cui si trovavano, cioè IMG (Immagine). Le immagini presenti su questo sito sono numerose, ne sono quindi state selezionate cinquantuno come campione per il fitting e il calcolo delle coordinate del centro.

Per poter lavorare con Matlab, ma in generale per poter visualizzare il contenuto delle foto, si è dovuto convertire (o calibrare) le immagini della sonda Cassini da IMG a PNG. Questo è stato possibile scaricando un programma chiamato *img2png* dal sito [B], dove inizialmente era in una cartella zippata *img2png.zip*. Una volta estratto il contenuto ci si trova davanti a una cartella con all'interno vari tools e l'applicativo, dopodiché si portano le immagini in formato IMG che si desidera calibrare, all'interno di questa cartella.

Una volta arrivati a questo punto, manca solo di avviare il programma, ciò non è possibile tramite i metodi convenzionali cliccando semplicemente sull'applicazione *img2png*, ma bisogna farlo dal *Prompt dei Comandi*. Una volta aperto si deve digitare il comando *cd* (change directory) seguito dal

percorso della cartella del programma, poi premere invio per accederci all'interno.

Successivamente digitando `img2png *.img` nel *Prompt*, il programma si avvia ed elabora e converte tutte le immagini di formato IMG presenti all'interno della sua cartella, trasformandole in un formato PNG. Per capire se il programma ha funzionato correttamente, all'interno della cartella si dovranno avere due immagini per ciascuna foto, una in formato IMG e una in PNG.

## 5.2 DATABASE

Calibrate le immagini della sonda Cassini, è stato utilizzato il documento [1] il quale contiene una tabella avente numerosi dati relativi a ognuna delle immagini campione qui utilizzate, per confrontare i dati ottenuti dal programma creato in Matlab verificandone il livello di accordo con i dati di letteratura. In questa tabella sono presenti i dati di maggior interesse per l'obbiettivo di questa tesi, le coordinate dei centroidi dell'ellissi di fitting, ricavate con metodi differenti. Tutti questi dati relativi a ognuna delle cinquantuno immagini campione sono poi stati trascritti su una tabella *Excel*, che è stata importata tramite il comando `xlsread` nel software Matlab.

## 5.3 LETTURA IMMAGINE ED EDGE DETECTION

```
%% READ IMAGES %%  
im = imread('N1484577892_1.png');
```

Name	Size	Bytes	Class	Attributes
im	1024x1024	2097152	uint16	

*Figura 22: Sopra si può vedere il comando che `imread` che carica l'immagine, mentre sotto sono elencate le sue caratteristiche grazie alla funzione `who(im)`*

Tramite il comando `imread` si carica un'immagine in Matlab e usando `imshow` è possibile visualizzarla in una finestra grafica. L'immagine viene rappresentata nell'Workspace come una matrice 1024x1024 `uint 16` (oppure `uint 8` a seconda dell'immagine), questo significa che è

un'immagine quadrata con lati di 1024 pixel e ogni elemento assume un valore da 0 a 255 d'intensità di valori di grigio che viene riportato in una determinata posizione della matrice. Inoltre con *uint 16* s'intende che ogni numero intero della matrice, occupa 2 byte, cioè 16 bit. Per esempio la dimensione, il peso e la classe dell'immagine, sono riassunte in *figura 22* grazie alla funzione *who im* (dove *im* è il nome del file).

Fatto questo, l'immagine è caricata nel software, ora utilizzando il comando *edge* (come si può vedere in *figura 23*) è possibile procedere con il rilevamento dei bordi tramite il metodo Canny. L'intervallo d'intensità in cui i pixel vengono identificati come oggetti di contorno, varia da immagine a immagine ed è bene trovare il giusto range, perché nella fase successiva di regressione, si possono ottenere risultati completamente differenti al variare di poco degli estremi dell'intervallo. Il file che si ottiene tramite la funzione *edge*, che è stato definito in questo caso come *edgeim* è sempre una matrice di 1024x1024 elementi, ma ognuno di essi assume come valore 0 o 1; 0 se il pixel non fa parte del limbo del corpo celeste e quindi rimane spento, mentre 1 se è un pixel facente parte del bordo, quindi risulta acceso.

Tramite la funzione *find*, vedi *figura 23*, è possibile creare dei vettori, in questo caso chiamati *idx1* e *idx2*, dove ogni elemento ha come valore un indice che identifica, la posizione di un pixel acceso nella matrice *edgeim*. Questi risultano poi essere le coordinate *x,y* dei pixel accesi dell'immagine in analisi. Questi due vettori che si vengono a formare saranno poi utilizzati da tutte le funzioni di regressione, perché per poter

```

%%      READ IMAGES      %%
im = imread('N1484577892_1.png');

%%      EDGE DETECTION & PROPERTIES      %%

perimeter_coord = bwperim(im,8); %coordinate perimetro
whos im %lista di tutte le variabili, tipo dimensione ecc.. del file im

edgeim = edge(im,'canny',[0.3 .9]);
% imwrite (edgeim, 'im_mod41.png');

struttura_im = regionprops(edgeim,'Centroid','MajorAxisLength','MinorAxisLength');
centroids = cat(1, struttura_im.Centroid);
majorAxisLength = struttura_im.MajorAxisLength;
minorAxisLength = struttura_im.MinorAxisLength;
[idx1,idx2] = find(edgeim==1);

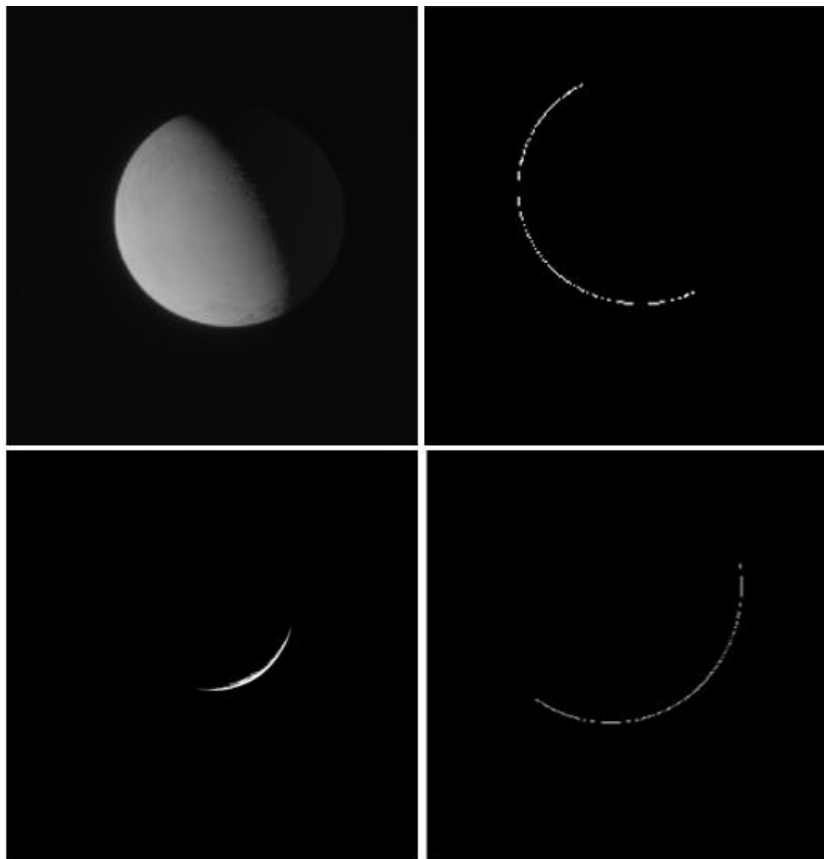
```



*Figura 23: In questa figura è possibile vedere la struttura della funzione edge e i valori del range utilizzati applicando il metodo Canny per quella specifica immagine. Successivamente si può notare la funzione find com'è strutturata.*

approssimare il limbo individuato ad un'ellisse, loro necessitano di conoscere le coordinate dei pixel che compongono il bordo del corpo celeste.

Prendendo due immagini molto differenti tra loro ma relative allo stesso Satellite, Encelado, come quelle in *figura 24*, è possibile vedere il risultato del lavoro dal metodo Canny durante la rilevazione del bordo. In questo caso il range d'intensità utilizzato per queste due immagini è lo stesso e va da [0.3 0.9]:



*Figura 24: In alto è possibile vedere il risultato dell'edge detection in una figura dove è ben visibile il corpo celeste. In basso l'immagine originale è buia, e il risultato del rilevamento dei bordi lo si può vedere a fianco. L'immagine che si osserva in basso a sinistra è stata modificata per permettere al lettore di vedere il limbo del corpo. (La prima è la N1484519176\_1, la seconda N1484577892\_1)*

## 5.4 ERRORE E ERRORE MEDIO

Nel Capitolo successivo, si potrà osservare la tabella con i risultati ottenuti dal software relativi al calcolo del centroide da parte di tutte e sei le funzioni di regressione, sulle cinquantuno immagini campione utilizzate in questa tesi. Questo servirà per individuare quale o quali metodi hanno una stima migliore riguardo alla posizione del centro dell'ellisse. Inoltre verranno analizzate alcune delle immagini elaborate dalle funzioni, per avere un confronto e capire quali di questi metodi è più preciso nel fittare l'ellisse e in base che tipo d'immagine lo è. Per ogni funzione è stato calcolato l'errore commesso (misurato in Pixel) per il rilevamento della posizione del centroide della conica, cioè la distanza euclidea tra le coordinate dei punti ottenute dai vari metodi e quelle usate per il confronto presenti nell'articolo [1]. In questo Capitolo verranno prese in esame solo due immagini (vedi *figura 24*) per mostrare il funzionamento che ognuna delle funzioni di fitting ha, i risultati ottenuti sia per l'approssimazione ad ellisse che per la stima del baricentro, relativo a queste due foto d'Encelado, sono molto simili tra loro, a parte per i metodi *algellipse* e *bookstein* i cui risultati sono stati tra i peggiori.

Si definisce l'*errore* commesso, come la distanza euclidea che si ha tra le coordinate di confronto del centroide (definite come *xconfronto* e *yconfronto*, come si può vedere in *figura 31*) e quelle ottenute dalle funzioni d'*ellipse fitting*. Quest'*errore* può quindi essere calcolato come:

$$err_{fun,x} = x_{fun} - x \quad [Pixel] \quad (78)$$

$$err_{fun,y} = y_{fun} - y \quad [Pixel] \quad (79)$$

$$err_{fun} = \sqrt{err_{fun,x}^2 + err_{fun,y}^2} \quad [Pixel] \quad (80)$$

la cui unità di misura è il Pixel. È stato creato un database contenente tutte le coordinate ottenute tramite i vari metodi, questo è riportato nelle *tabelle* di inizio Capitolo 6, da qui si è poi calcolato l'errore che ogni funzione ha compiuto relativo a ogni immagine (in questo Capitolo come già detto ne verranno prese in esame solo due).

```

% Coordinate di Confronto
xconfronto = database(:,8);
yconfronto = database(:,9);

% Coordinate Algellipse
algex = database(:,11);
algey = database(:,12);

% Coordinate Bookstein
bookx = database(:,14);
booky = database(:,15);

% Coordinate EllipseDirectFit
dirfitx = database(:,17);
dirfity = database(:,18);

% Coordinate EllipseByTaubin
ellfitabux = database(:,20);
ellfitabuy = database(:,21);

% Coordinate Fit_Ellipse
f_ex = database(:,23);
f_ey = database(:,24);

% Coordinate Fitellipse
fitellx = database(:,29);
fitelly = database(:,30);

```

Figura 25: Qui si sono stati definiti i vari vettori delle coordinate dei centroidi di ogni immagine, per ognuna delle sei funzioni

## 5.5 ALGELLIPSE

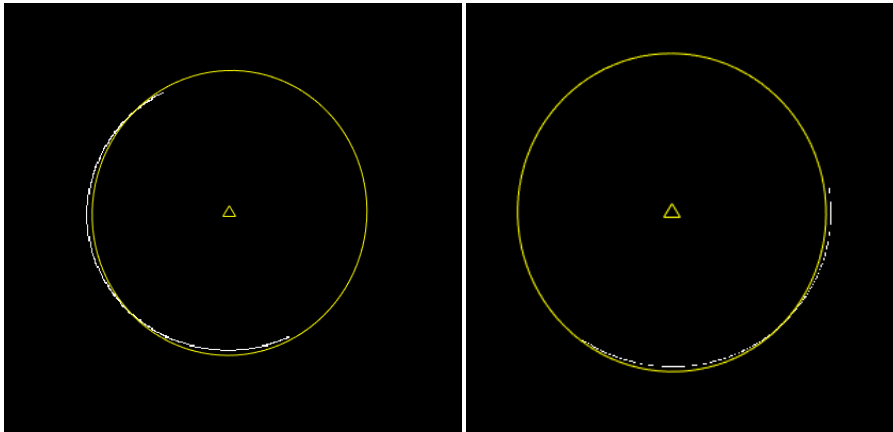
La funzione *algellipse* è una funzione basata sulla ricerca di una curva di regressione, in questo caso un'ellisse, che minimizzi la distanza algebrica dai punti d'input a quelli della curva ottima. Questo significa risolvere un problema lineare dei minimi quadrati, per questa funzione il vincolo è quello che il modulo del vettore dei coefficienti dell'equazione (13) sia uguale a uno. Per poter funzionare questa tecnica, ha bisogno di un'ulteriore funzione chiamata *ellipse\_params*, la quale calcola i parametri dell'ellisse dai coefficienti dell'equazione delle coniche, in cui si è minimizzata la distanza algebrica, questi poi vengono dati ad *algellipse* che li restituisce come output. Questo metodo richiede in input i vettori delle coordinate *idx1* e *idx2* trovati in precedenza dalla funzione *find* (vedi Paragrafo 5.3) e come output restituisce il semiasse maggiore e minore, il centroide e l'angolo d'orientamento dell'ellisse.

Si può osservare che come risultato si ha una forma parametrica dell'ellisse, ma queste sue caratteristiche sono state calcolate dai coefficienti dell'equazione che descrive l'ellisse che ne minimizza la distanza algebrica.

### 5.5.1 FITTING

Le due immagini che sono state sottoposte al metodo Canny e successivamente all'approssimazione tramite curva di regressione sono quelle presenti in *figura 24*. Il risultato ottenuto nella prima immagine è discreto, l'ellisse è simile alla forma reale del corpo celeste anche se non segue esattamente il percorso individuato dal limbo, quindi risulta leggermente più schiacciata ai lati e un po' allungata.

Mentre nella seconda immagine, l'approssimazione è meno precisa, l'ellisse di regressione interseca il limbo d'Encelado e risulta traslata verso il basso. Vedi *figura 26*



*Figura 26: Due foto d'Encelado a cui è stato applicato il metodo Canny d'edge detection e successivamente è stato fatto un fitting tramite la funzione algellipse. Fanno riferimento alle immagini in figura 24*

### 5.5.2 CENTROIDI

Il risultato finale che si è ottenuto da questa funzione sono le coordinate dei centroidi, di cui è stato fatto un confronto, come già detto, tra quelli restituiti come output e quelli presenti in [1]. Nella *tabella* di inizio Capitolo 6 sono riportati i valori delle coordinate  $x$  e  $y$  del centro dell'ellisse.

Per la prima immagine d'Encelado, è possibile notare che le coordinate sono simili, infatti usando le relazioni (78), (79) e (80) l'errore ottenuto è:

$$err_{alg,x_1} = 514.2317 - 512.1 = 2.1137 \quad [Pixel]$$

$$err_{alg,y_1} = 510.1808 - 509.6 = 0.5808 \quad [Pixel]$$

$$err_{alg,1} = \sqrt{2.1137^2 + 0.5808^2} \cong 2.192 \quad [Pixel]$$

Mentre per la seconda immagine l'errore risulta molto più basso:

$$err_{alg,x_2} = 512.35 - 512.4 = -0.05 \quad [Pixel]$$

$$err_{alg,y_2} = 647.95 - 647.95 = 0 \quad [Pixel]$$

$$err_{alg,2} = \sqrt{0.05^2 + 0^2} \cong 0.05 \quad [Pixel]$$

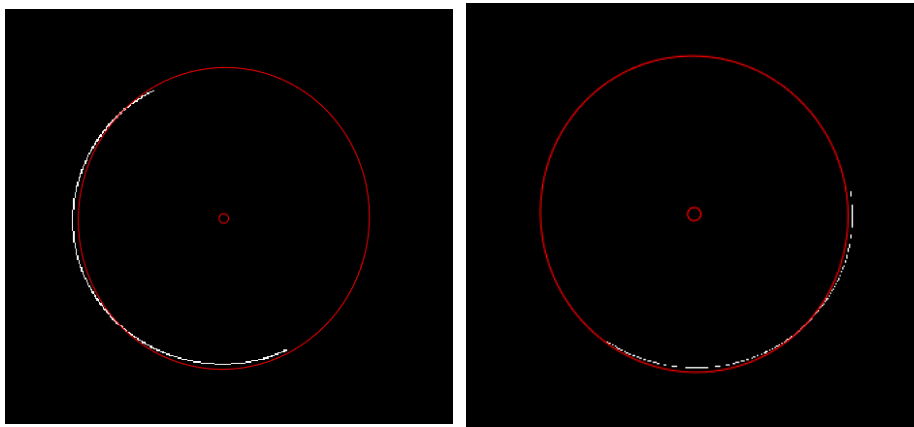
## 5.6 BOOKSTEIN

La funzione *Bookstein* si basa anch'essa sul principio dei minimi quadrati con soluzione lineare, ma ha un vincolo che è differente rispetto a quello usato in *algellipse*. Questo vincolo fa in modo che l'ellisse sia invariante sotto trasformazione euclidea.

In input richiede i vettori *idx1* e *idx2* (vedi Paragrafo 5.3). Come il metodo *algellipse*, necessita della funzione *ellipse\_params*, la quale restituisce a *Bookstein* i parametri fondamentali dell'ellisse, centroide, semiassi e l'angolo d'orientamento, questi successivamente diventano l'output della funzione. Anch'essa è stata presa dall'articolo [2].

### 5.6.1 FITTING

Come fatto con la precedente funzione, verrà analizzata la qualità di fitting sulle due immagini d'Encelado, com'è possibile vedere in *figura 27*:



*Figura 27: Due foto d'Encelado a cui è stato applicato il metodo Canny d'edge detection e successivamente è stato fatto un fitting tramite la funzione bookstein. Fanno riferimento alle immagini in figura 24.*

La prima immagine, approssima con una discreta qualità la forma del satellite saturniano, anche se si può notare che è leggermente schiacciata

ai lati e quindi non segue completamente i punti individuati con l'*edge detection* (molto simile al risultato ottenuto d'*algellipse*).

La seconda immagine, è stata fittata con una discreta precisione, ma come per *algellipse* la qualità è minore rispetto la prima, l'ellisse d'approssimazione non parte dal limbo del corpo e inoltre risulta traslata verso il basso.

### 5.6.2 CENTROIDI

Il confronto dei centroidi, come per *algellipse*, è possibile trovarlo nella *tabella* di inizio Capitolo 6. L'*errore* che si ottiene da quei dati per la prima immagine è:

$$err_{book,x_1} = 514.0500 - 512.1 = 1.95 \quad [Pixel]$$

$$err_{book,y_1} = 510.0941 - 509.6 = 0.4941 \quad [Pixel]$$

$$err_{book,1} = \sqrt{1.95^2 + 0.4941^2} \cong 2.0116 \quad [Pixel]$$

Mentre analizzando l'errore relativo alla seconda immagine è:

$$err_{book,x_2} = 510.7085 - 512.4 = -1.6915 \quad [Pixel]$$

$$err_{book,y_2} = 649.7686 - 647.95 = 1.8186 \quad [Pixel]$$

$$err_{book,2} = \sqrt{1.6915^2 + 1.8186^2} \cong 2.4836 \quad [Pixel]$$

## 5.7 ELLIPSE DIRECT FIT

Questa funziona si basa sui minimi quadrati, cercando una soluzione di tipo lineare senza utilizzare le iterazioni. Ha la particolarità che se i punti approssimerebbero meglio un'iperbole, la funzione restituisce lo stesso un'ellisse, anche se la forma non sarà tra le migliori.

*Ellipse direct fit* calcola l'equazione dell'ellisse che minimizza la distanza algebrica tra i punti, quindi questa funzione restituisce i coefficienti della conica, successivamente tramite le formule presenti nel Capitolo 4.1 è possibile ricavare il centroide, semiassi e l'angolo d'orientamento.

### 5.7.1 FITTING

Il confronto tra le solite due immagini è quello mostrato in *figura 28*:

Come si può notare la prima immagine è stata fittata con ottima qualità, infatti l'ellisse segue molto bene l'andamento della parte visibile del corpo celeste mantenendo le giuste proporzioni. Mentre la seconda immagine non segue esattamente il bordo individuato dall'*edge detection*, ma tuttavia il risultato può essere considerato di buona qualità.

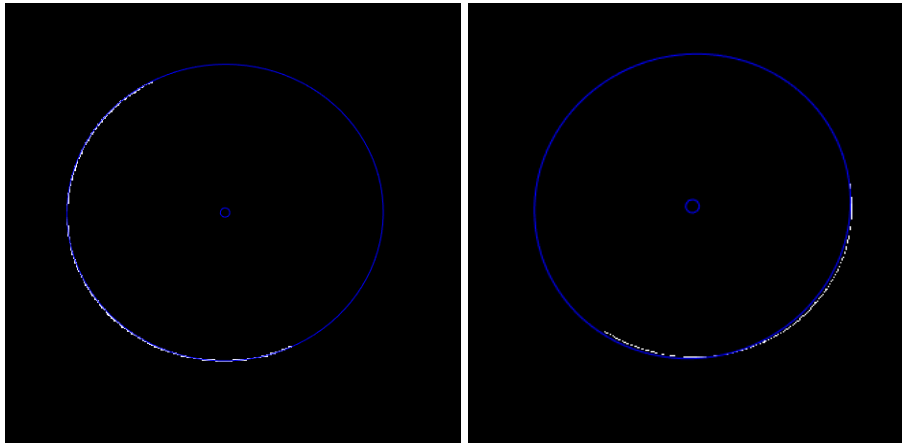


Figura 28: Due foto d'Encelado a cui è stato applicato il metodo Canny d'*edge detection* e successivamente è stato fatto un fitting tramite la funzione *Ellipse Direct Fit* Fanno riferimento alle immagini in figura 24

### 5.7.2 CENTROIDI

Il risultato del calcolo del centroide da parte della funzione *ellipse direct fit* e relativo a queste due immagini è nella *tabella* di inizio Capitolo 6. Grazie a quei dati è possibile trovare l'*errore* commesso da questo metodo.

Per la prima immagine risulta:

$$err_{e.direct,x_1} = 514.0377 - 512.1 = 1.9377 \quad [Pixel]$$

$$err_{e.direct,y_1} = 510.0888 - 509.6 = 0.4888 \quad [Pixel]$$

$$err_{e.direct,1} = \sqrt{1.9377^2 + 0.4888^2} \cong 1.998 \quad [Pixel]$$

Quello relativo alla seconda immagine invece è:

$$err_{e.direct,x_2} = 510.7099 - 512.4 = 1.6901 \quad [Pixel]$$

$$err_{e.direct,y_2} = 649.7712 - 647.95 = 1.8212 \quad [Pixel]$$

$$err_{e.direct,2} = \sqrt{1.6901^2 + 1.8212^2} \cong 2.4845 \quad [Pixel]$$

## 5.8 ELLIPSE FIT by TAUBIN

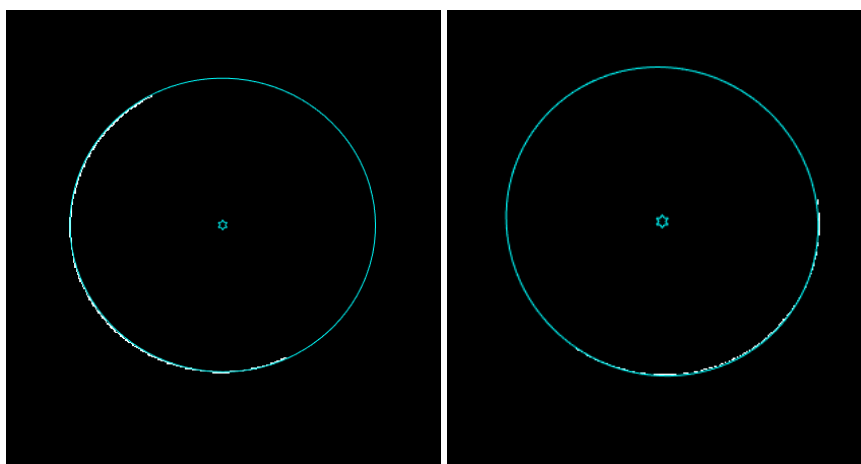
Questo metodo è stato pubblicato da G. Taubin in [3], anch'esso richiede in input un insieme di coordinate, le quali nel caso di questa tesi, sono le coordinate dei pixel del bordo del corpo celeste ottenute tramite *l'edge detection* e raggruppate dalla funzione *find*.

Questo metodo restituisce il vettore dei coefficienti dell'equazione algebrica, da dove è poi possibile, tramite le formule dell'ellisse viste nel Capitolo 4.1, calcolare i parametri necessari al fitting.

È un metodo non iterativo, molto accurato e robusto, sostanzialmente è molto simile ad Ellipse Direct Fit, perché anch'essa se i coefficienti appartengono ad un'ellisse utilizza il metodo Direct Fit. Detto questo però, sarà possibile notare che la precisione sia in termini di forma dell'ellisse che riguardo ai centroidi è diversa, Ellipse fit By Taubin risulta essere molto più precisa.

### 5.8.1 FITTING

Le due immagini a cui è stato sottoposto un'*ellipse fitting* dal metodo *Ellipse fit by Taubin*, dopo un accurato *edge detection* sono quelle in *figura 29*:



*Figura 29: Due foto d'Encelado a cui è stato applicato il metodo Canny d'edge detection e successivamente è stato fatto un fitting tramite la funzione Ellipse Fit by Taubin. Fanno riferimento alle immagini in figura 24*

La prima immagine ha un'ottima qualità d'approssimazione, segue esattamente il limbo individuato dal rilevamento dei bordi. La seconda ellisse di regressione risulta anch'essa molto precisa.



### 5.8.2 CENTRODI

Nella *tabella* di inizio Capitolo 6 si trovano le coordinate del centroide relativo alle immagini ottenute dai veri metodi e comparate con quelle nell'articolo [1]. Da questi risultati è possibile calcolare l'*errore*, quello relativo alla prima immagine è:

$$err_{taubin,x_1} = 514.1884 - 512.1 = 2.0884 \quad [Pixel]$$

$$err_{taubin,y_1} = 510.1619 - 509.6 = 0.5619 \quad [Pixel]$$

$$err_{taubin,1} = \sqrt{2.0884^2 + 0.5619^2} \cong 2.1626 \quad [Pixel]$$

Si può commentare dicendo che fino a qui l'errore che le varie funzioni di regressione hanno commesso sulla prima foto è molto simile. Le coordinate relative al secondo centroide invece calcolate da *ellipse fit by Taubin* è leggermente più elevato rispetto agli altri:

$$err_{taubin,x_2} = 510.2080 - 512.4 = -2.192 \quad [Pixel]$$

$$err_{taubin,y_2} = 650.534 - 647.95 = 2.584 \quad [Pixel]$$

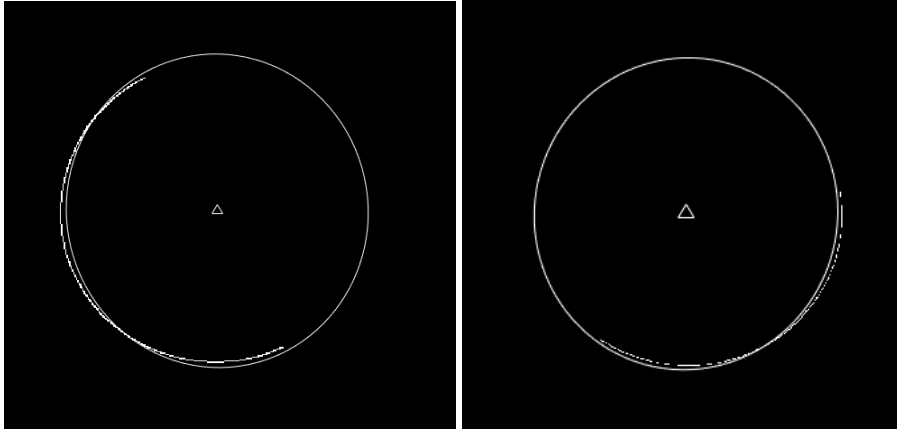
$$err_{taubin,2} = \sqrt{2.192^2 + 2.584^2} \cong 3.3884 \quad [Pixel]$$

### 5.9 FIT\_ELLIPSE

La funzione *fit\_ellipse* come già detto in precedenza utilizza il metodo d'approssimazione dei minimi quadrati con soluzione non lineare. Necessita in ingresso almeno cinque punti (dieci coordinate) e come output restituisce una *struct* contenente i parametri dell'ellisse che minimizza la distanza geometrica tra i punti, il risultato finale è quindi una forma parametrica della conica.

### 5.9.1 REGRESSIONE AD ELLISSE

Usando sempre le due foto d'Encelado viste nei Paragrafi precedenti, è possibile osservare l'operato di *fit\_ellipse* in *figura 30*:



*Figura 30: Due foto d'Encelado a cui è stato applicato il metodo Canny e successivamente è stato fatto una regressione tramite la funzione *fit\_ellipse*. Fanno riferimento alle immagini in *figura 24**

In nessuna delle due immagini si può dire di avere ottenuto una approssimazione accurata, perché in entrambi i casi l'ellisse non segue in maniera pulita i punti che compongono il limbo del satellite naturale.

### 5.9.2 CENTROIDE

Le coordinate relative al centroide date come output da questa funzione relative al centro dell'ellisse, sono riportate nella *tabella* di inizio Capitolo 6. Grazie a quei risultati è stato possibile calcolare l'*errore* commesso dalla funzione sul centro dell'ellisse nelle due immagini.

Nella prima è:

$$err_{f\_ell,x_1} = 514.1725 - 512.1 = 2.0725 \quad [Pixel]$$

$$err_{f\_ell,y_1} = 510.1540 - 509.6 = 0.554 \quad [Pixel]$$

$$err_{f\_ell,1} = \sqrt{2.0725^2 + 0.554^2} \cong 2.1452 \quad [Pixel]$$

Mentre quello relativo alla seconda immagine:

$$err_{f\_ell,x_2} = 510.2353 - 512.4 = -2.1647 \quad [Pixel]$$

$$err_{f\_ell,y_2} = 650.4955 - 647.95 = 2.5455 \quad [Pixel]$$

$$err_{f\_ell,2} = \sqrt{2.1647^2 + 2.5455^2} \cong 3.3415 \quad [Pixel]$$

Anche qui lo scarto ottenuto per la prima immagine è coerente rispetto le precedenti funzioni e anche quello relativo alla seconda è simile a quello ottenuto con *ellipse fit by Taubin*

## 5.10 FIT ELLIPSE

Il nome è molto simile alla funzione vista precedentemente, differisce solo di un underscore, ma i risultati generali ottenuti sono di una qualità superiore, sia per la forma dell'ellisse, sia per l'accuratezza nel calcolo del centroide.

Il primo step è il calcolo dei minimi quadrati lineare, di default usa il vincolo Bookstein, questo restituisce le condizioni iniziali da cui il metodo iterativo partirà nel calcolo dei parametri. È un processo non lineare il quale richiede in input *idx1* e *idx2*, e restituisce i parametri dell'ellisse, quindi la forma parametrica dell'ellisse che minimizza la distanza geometrica. Questo metodo ibrido, è composto da varie funzioni al suo interno che servono appunto per la risoluzione iterativa ed è possibile dire che la qualità di fitting e del calcolo del centroide è ottima.

### 5.10.1 FITTING

Dalle due immagini d'Encelado fino qui utilizzate nei precedenti Paragrafi, si può osservare come il metodo *FitEllipse*, abbia una precisione

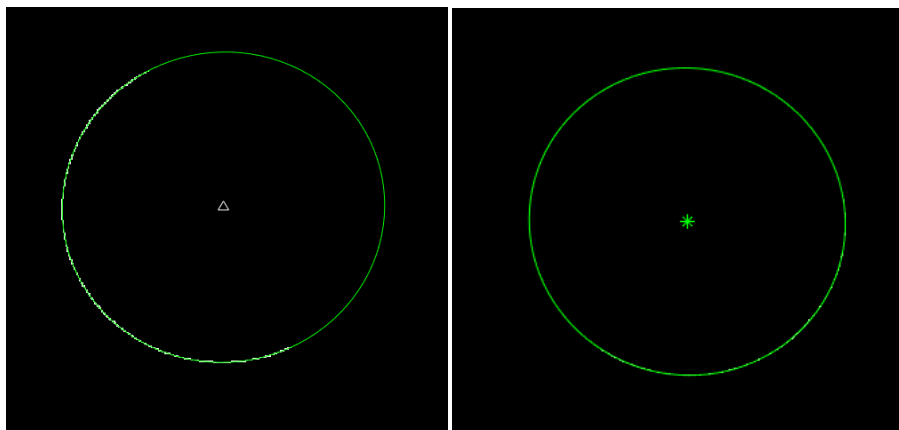


Figura 31: Due foto d'Encelado a cui è stato applicato il metodo Canny d'edge detection e successivamente è stato fatto un fitting tramite la funzione *Fit Ellipse*. Fanno riferimento alle immagini in figura 24

in assoluto migliore nella costruzione dell'ellisse rispetto a tutte le precedenti funzioni. Infatti la curva segue perfettamente la forma che viene

suggerita dal limbo del corpo celeste, per entrambe le figure, inoltre la proporzione e l'orientamento risultano di un'ottima qualità, soprattutto la seconda immagine risulta quasi impeccabile (*figura 31*).

### 5.10.2 CENTROIDE

Dalla *tabella* di inizio Capitolo 6, è possibile rilevare i dati del centroide ottenuti con il metodo *fitellipse* e confrontati con quelli presenti nell'articolo [1]. Da qui si è calcolato l'*errore* per entrambe le immagini, per la prima si è ottenuto:

$$err_{fitellipse,x_1} = 514.2084 - 512.1 = 2.1084 \quad [Pixel]$$

$$err_{fitellipse,y_1} = 510.1725 - 509.6 = 0.5725 \quad [Pixel]$$

$$err_{fitellipse,1} = \sqrt{2.1084^2 + 0.5725^2} \cong 2.1847 \quad [Pixel]$$

Mentre quello relativo alla seconda:

$$err_{fitellipse,x_2} = 510.2390 - 512.4 = -2.1610 \quad [Pixel]$$

$$err_{fitellipse,y_2} = 650.4919 - 647.95 = 2.5419 \quad [Pixel]$$

$$err_{fitellipse,2} = \sqrt{2.1610^2 + 2.5419^2} \cong 3.3363 \quad [Pixel]$$

Anche con quest'ultima funzione, è possibile notare che l'errore commesso nel calcolo del centroide della prima immagine è coerente con gli altri risultati, mentre quello relativo alla seconda immagine, è leggermente inferiore rispetto a *ellipse fit by Taubin e Fit\_ellipse*, ma superiore ai primi tre metodi.

## 5.11 CONCLUSIONI CAPITOLO

In queste due immagini è possibile vedere come lavorano le funzioni d'approssimazione, dove *fitellipse* (ibrido) è la funzione che ha ottenuto la migliore curva di regressione per entrambe le immagini. Mentre *algellipse* (MDA) ha ottenute l'errore minore nel calcolo del centroide. I risultati ottenuti dal calcolo dell'errore relativo ai centri dell'ellissi per queste due immagini, in questo caso, potrebbe fuorviare e far pensare che i metodi algebrici siano quelli più precisi, ma come verrà mostrato nel Capitolo successivo, la realtà sarà ben diversa.

## 6. RISULTATI OTTENUTI DALLE FUNZIONI

### 6.1 TABELLA CENTROIDI

IMMAGINE	X CONF.	Y CONF.	X ALGEL.	Y ALGEL.	X BOOK.	Y BOOK.	X DIRECT	Y DIRECT	X TAUBIN	Y TAUBIN	X FIT_ELL	Y FIT_ELL	X FIEELIS	Y FIEELIS	X FITELLIS	Y FITELLIS
N1466584402_1	515.22	523.55	515.474	521.5154	515.4877	521.5136	515.4876	521.5139	515.4866	521.5105	515.4877	521.5138	515.4908	521.5088		
N1466584454_1	515.19	519.29	515.3931	517.3678	515.3931	517.3588	515.4014	517.3592	515.3966	517.3593	515.4004	517.3593	515.4207	517.362		
N1466584531_1	512.13	517.68	512.2455	515.9014	512.256	515.8906	512.2616	515.8958	512.2559	515.8923	512.262	515.8951	512.2598	515.905		
N1466584557_1	511.73	514.87	512.2281	512.87	512.2281	512.8607	512.2402	512.8682	512.2287	512.8649	512.2392	512.8677	512.2564	512.8895		
N1466584912_1	513.84	518.5	514.18	516.6836	514.1848	516.6802	514.1886	516.6853	514.1846	516.6831	514.1884	516.6849	514.1814	516.7214		
N1467495142_2	689.57	408.5	690.9498	404.8392	690.585	402.8646	690.6329	404.5733	690.6422	411.1748	690.6671	410.5563	690.6287	412.1876		
N1467553222_2	437.78	361.69	439.1631	335.0922	439.041	335.4342	439.0321	336.3609	439.0038	358.2249	439.0068	358.1283	439.0035	358.3609		
N1476211039_2	586.64	410.65	590.4351	409.5115	590.4287	409.4996	590.4197	409.4875	590.4306	409.498	590.4215	409.4904	590.3928	409.4508		
N1476383960_2	724.67	294.64	724.6178	291.9942	724.6123	291.98	724.6194	291.9851	724.6151	291.9829	724.6187	291.9845	724.61	292.0347		
N1476468560_2	556.74	332.59	556.712	330	556.7339	329.8783	556.7419	329.884	556.7369	329.876	556.7417	329.8821	556.7799	329.8999		
N1476788542_2	448.56	412.49	448.3951	409.4477	448.4156	409.4261	448.4183	409.4371	448.4169	409.4293	448.4183	409.4351	448.4185	409.4817		
N1477935150_2	450.37	544.48	451.2444	544.2986	447.1548	538.6724	448.314	540.3546	450.6748	543.8469	450.4125	543.4481	450.6492	543.873		
N1478818535_2	328.81	392.56	327.6006	388.0984	327.7732	388.0033	327.7954	388.0347	327.7723	387.9928	327.7944	388.0254	327.7452	388.0703		
N1479140857_2	214.22	577.36	213.3887	573.3837	213.5409	573.3837	213.5092	573.441	213.5381	573.3827	213.5175	573.4289	213.4072	573.5484		
N1479200258_2	281.3	730.05	281.7379	726.8393	281.1216	725.3517	281.3515	726.36	281.7284	727.8031	281.6796	727.5416	281.6662	727.4441		
N1480051843_1	436.57	543.08	435.9392	539.8732	435.9395	539.8503	435.9602	539.8586	435.944	539.8539	435.9582	539.858	435.9767	539.913		
N1480649447_2	552.72	389.43	552.406	386.5225	552.4181	386.5166	552.422	386.5198	552.4185	386.5168	552.4222	386.5192	552.4169	386.5643		
N1481035670_2	584.39	532.38	584.2087	529.4708	584.2124	529.4578	584.2166	529.4619	584.2122	529.4574	584.2164	529.4614	584.2499	529.4916		
N1481236191_2	127.46	702.58	127.4257	700.6408	127.4189	700.6526	127.4207	700.6502	127.4169	700.6534	127.4202	700.6508	127.4039	700.6106		
N1481316652_2	437.64	485.01	437.6582	483.3226	437.5655	483.1774	437.585	483.2111	437.6307	483.3083	437.6246	483.2939	437.63	483.294		
N1481355712_2	464.34	399.29	465.1235	397.0967	465.0657	397.238	465.0695	397.3029	465.0437	397.668	465.0496	397.6311	465.043	397.6757		
N1481515793_2	527.69	514.05	528.9148	512.9783	528.8652	512.9431	528.8709	512.9634	528.8656	513.3323	528.8678	513.2962	528.8648	513.3306		
N1481705814_2	498.19	634.35	500.1012	633.0544	500.0694	633.1067	500.064	633.1216	500.0366	633.1912	500.0401	633.1829	500.0359	633.1906		
N1484509816_1	563.91	505.55	564.1942	505.3242	564.0391	505.267	564.0331	505.2692	564.2615	505.3186	564.2399	505.3145	564.2478	505.3177		

NI1484509882_1	563.41	505.72	563.7805	505.5899	563.6226	505.5321	563.6165	505.5341	563.8421	505.5844	563.822	505.5805	563.8464	505.5856
NI1484509915_1	562.53	506.12	563.277	505.9281	563.1305	505.8753	563.12	505.8765	563.3428	505.9221	563.3218	505.9184	563.3409	505.9226
NI1484510353_1	559.8	505.09	560.4137	504.927	560.2085	504.8539	560.1987	504.8561	560.494	504.9202	560.4675	504.9154	560.4692	504.9187
NI1484510386_1	561.29	505.19	561.7445	505.0343	561.5645	504.9686	561.5591	504.9711	561.809	505.0284	561.7871	505.0239	561.7967	505.0283
NI1484517052_1	545.27	504.73	547.0554	504.8747	546.7437	504.781	546.6966	504.7786	547.2258	504.8691	547.1843	504.8629	547.264	504.8799
NI1484517118_1	545.3	505.29	547.0545	505.7103	546.7087	505.6071	546.6565	505.6044	547.243	505.7045	547.1944	505.6974	547.3079	505.719
NI1484517151_1	544.37	505.43	545.0387	505.4638	544.7235	505.3705	544.6783	505.3703	545.2089	505.457	545.1635	505.4507	545.2619	505.4687
NI1484573247_1	521.99	516.81	5.29E+02	5.08E+02	527.5843	510.1858	527.5171	510.2699	526.7016	511.6786	526.7478	511.5936	526.6736	511.7352
NI1501627117_1	476.02	515.36	477.9207	515.9799	477.6406	515.9368	477.5238	515.944	478.4355	515.9024	478.3562	515.9066	478.693	515.8961
NI1521712680_1	613.76	443.1	616.7115	443.527	598.3841	431.8822	603.6096	435.1351	614.7326	441.9053	613.8782	441.4225	614.166	441.5049
NI1648319659_1	643.09	547.8	644.263	546.223	664.0971	545.4382	644.1165	545.7002	644.2236	546.9399	644.2141	546.873	644.2252	546.9389
NI1466514680_1	473.85	513.67	477.3489	512.6677	477.344	512.6487	477.3354	512.6549	477.3426	512.6484	477.3372	512.654	477.3119	512.6801
NI1484519176_1	512.09	509.57	514.2137	510.1808	514.05	510.0941	514.0377	510.0888	514.1884	510.1619	514.1725	510.154	514.2084	510.1725
NI1484577892_1	512.35	647.95	511.9521	647.95	510.7085	649.7686	510.7099	649.7712	510.208	650.534	510.2353	650.4955	510.239	650.4919
NI1487264883_2	419.36	628.73	419.129	624.7508	419.1239	624.7647	419.1245	624.7681	419.1314	624.8022	419.1307	624.7989	419.1295	624.8023
NI1525360749_1	567.21	448.42	568.7243	448.6487	568.5641	448.4496	568.5747	448.4644	568.6667	448.5848	568.6522	448.5662	568.6642	448.5814
NI1569849851_1	509.37	469.57	509.1576	465.2594	509.2231	465.035	509.2202	465.5916	509.3007	465.8698	509.295	465.8514	509.2965	465.8775
NI1572040488_1	546.47	435.94	549.8187	434.852	549.8143	434.8545	549.8139	434.8547	549.8124	434.8554	549.8133	434.855	549.8053	434.8576
NI1646171175_1	528.01	100.3	528.8108	96.5815	558.3276	96.6908	549.711	97.3777	529.711	99.0109	531.6367	98.8526	525.432	99.2019
NI1646174483_1	530.08	100.01	553.9474	96.4034	559.7027	96.204	551.3949	97.2791	532.8468	99.6533	534.57	99.4734	531.2137	99.7571
NI1646174617_1	529.79	99.4	555.0311	95.9099	560.8068	95.9099	552.2738	97.0041	533.3273	99.4454	535.0162	99.2344	530.3668	99.7601
NI1646174657_1	531.96	99.81	557.177	95.9854	562.8178	95.9083	554.6146	96.828	537.2161	98.7672	538.6802	98.6228	535.2661	98.946
NI1646174775_1	529.24	101.24	553.7914	96.8875	561.0923	96.875	551.0435	97.8083	524.8816	100.2144	527.9463	99.9867	561.0923	96.875
NI1646174965_1	528.81	100.8	556.6686	96.0321	565.0225	96.2062	552.8578	97.026	511.7233	99.8981	518.0348	99.545	565.0225	96.2062
NI1663946254_1	521.91	517.86	522.5042	517.23	522.5038	517.2305	522.5038	517.2035	522.5038	517.2305	522.5038	517.2305	522.5051	517.2306
NI1669795989_1	496.67	472.88	509.9663	467.8345	503.305	470.0648	503.3316	470.0597	501.8519	470.5255	501.9177	470.505	501.5268	470.6114
NI1675108551_1	492.06	225.85	490.9029	224.9847	489.8036	225.3316	489.9061	225.3117	488.2979	225.6907	488.3832	225.6703	488.2005	225.6983

## 6.2 ERRORE MEDIO (Misurato in PIXEL)

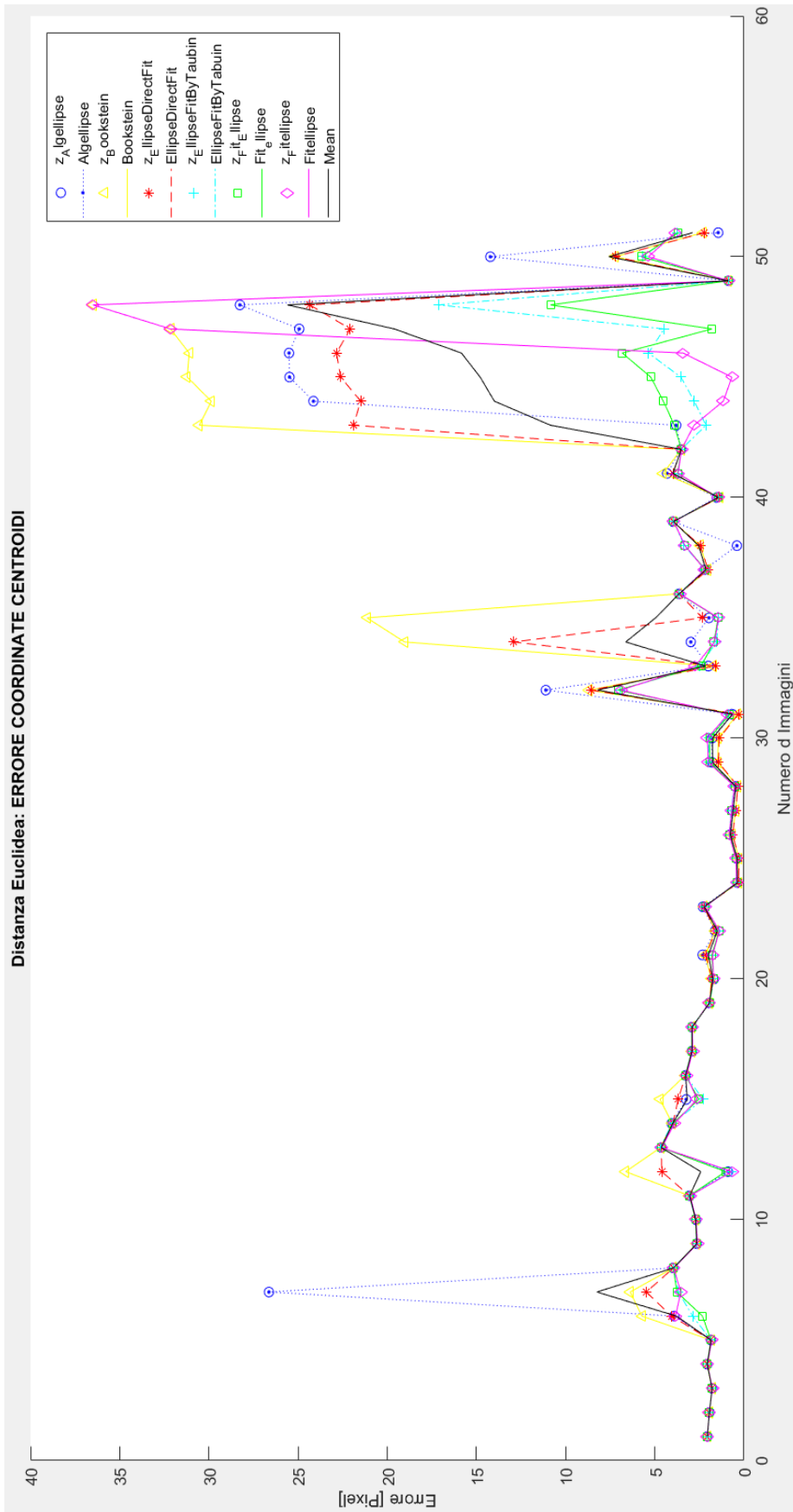
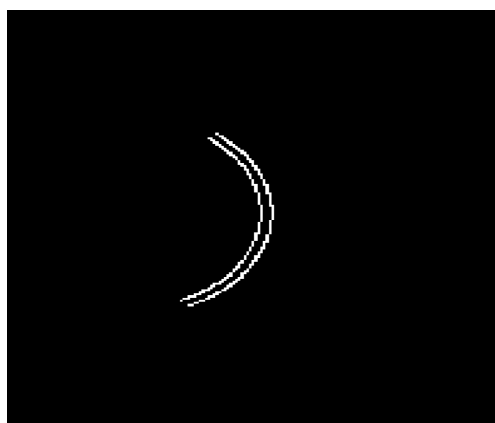


Figura 32: Grafico errore medio

Da come è possibile osservare nel grafico in *figura 32*, l'errore più grande di stima è stato commesso dal metodo *fitellipse* (ibrido) e *bookstein* (MDA), con uguale valore: 36.5027 [*Pixel*]. Questo è relativo all'immagine in *figura 33* (dopo l'applicazione del metodo Canny, il limbo del corpo si presenta in questa maniera, anche variando il range d'intensità luminosa). Anche le altre funzioni hanno commesso degli errori abbastanza elevati con un media (escludendo le prime due citate) di 20.141 [*Pixel*], relativi all'immagine sopracitata.

Gli altri picchi d'errore presenti nel grafico, parte destra, commessi dalle funzioni *bookstein* (MDA), *algellipse* (MDA) e *ellipse direct fit* (MDA), sono tutti riguardanti immagini, che dopo il rilevamento dei bordi, si presentano come quella in *figura 33*. **NB:** Il problema risiede nell'identificazione del limbo, non tanto nella fase di regressione ad ellisse, questo aspetto andrà approfondito in futuro.



*Figura 33: la maggior parte delle immagini con questo tipo di figura a doppia mezza luna sul lato destro, sono quelle che hanno determinato i picchi nel grafico dell'errore medio. (N1646174965\_1)*

Su ognuna delle figure è stato calcolato l'*errore medio* commesso dalle sei funzioni, come è possibile osservare nel grafico con i diversi colori, la curva in nero interpola i punti le cui coordinate sono rappresentate dalla media dell'errore commesso relativo a ogni immagine (vedi *figura 34*).

Si ritiene utile anche il calcolo della media dell'errore compiuto dai vari metodi su una singola immagine, nel caso le funzioni di regressione ad ellisse implementate siano numerose per avere una certa ridondanza. Per questo si è ritenuto importante riportare l'esempio in *figura 34*.



N1466584402_1	2,054596834	N1481316652_2	1,742546013	N1487264883_2	3,955633546
N1466584454_1	1,940449201	N1481355712_2	1,992576547	N1525360749_1	1,438338565
N1466584531_1	1,787810889	N1481515793_2	1,499988814	N1569849851_1	3,991310691
N1466584557_1	2,063081259	N1481705814_2	2,225075667	N1572040488_1	3,514690898
N1466584912_1	1,842766247	N1484509816_1	0,3694606	N1646171175_1	10,84559575
N1467495142_2	3,792535932	N1484509882_1	0,383123985	N1646174483_1	13,99448435
N1467553222_2	8,246385381	N1484509915_1	0,757693214	N1646174617_1	14,79187901
N1476211039_2	3,955493605	N1484510353_1	0,610708208	N1646174657_1	15,8431383
N1476383960_2	2,646996025	N1484510386_1	0,463257662	N1646174775_1	19,59746604
N1476468560_2	2,702925263	N1484517052_1	1,76208973	N1646174965_1	25,59495897
N1476788542_2	3,050684549	N1484517118_1	1,769699559	N1663946254_1	0,868920771
N1477935150_2	2,424020708	N1484517151_1	0,644649454	N1669795989_1	7,579220766
N1478818535_2	4,646366869	N1484573247_1	8,212974218	N1675108551_1	2,881555477
N1479140857_2	4,000847848	N1501627117_1	2,15563493		
N1479200258_2	3,18138499	N1521712680_1	6,630963019		
N1480051843_1	3,270911312	N1648319659_1	4,957828857		
N1480649447_2	2,919210531	N1466514680_1	3,630305068		
N1481035670_2	2,918206757	N1484519176_1	2,133299395		
N1481236191_2	1,937405501	N1484577892_1	2,544745685		

Figura 34: Tabella contenente la media dell'errore relativo a ogni immagine

Mentre nella tabella sotto (vedi figura 35), sono messi a confronto gli errori medi che ogni funzione ha commesso nel calcolo del centroide relativo a tutte e cinquantuno le immagini campione qui utilizzate:

<u>FUNZIONE</u>	<u>ERR. MED.</u>	<u>TIPOLOGIA</u>	<u>ERR. MAX.</u>
<b>ALGELLIPSE</b>	5,4433	MDA	28,26
<b>BOOKSTEIN</b>	6,9094	MDA	36,5
<b>ELLIPSE DIRECT FIT</b>	5,1855	MDA	24,34
<b>TAUBIN</b>	2,8657	MDA	17,11
<b>FIT_ELLIPSE</b>	2,8202	MDG	10,85
<b>FITELLIPSE</b>	3,6895	IBRIDO	36,5

Figura 35: Errore medio commesso da ogni funzione, per il calcolo del centroide delle 51 immagini campione (in pixel), tipologia e errore massimo compiuto.

Da qui è facile osservare che gli errori massimi commessi sono quelli compiuti dalla funzione ibrida e da Bookstein (MDA), ma come già detto

sono relativi a un'immagine che ha un pessimo limbo di partenza per la creazione della curva di regressione. Anche i massimi delle altre fuizioni sono relativi a quell'immagine. che minimizzano la distanza algebrica, quindi lineari.

I metodi che hanno l'errore medio più elevato sono quelli che minimizzano la distanza algebrica della curva di regressione dai punti del limbo individuati tramite il rilevamento dei bordi. Mentre quelli che hanno dato i risultati più accurati e quindi che si possono ritenere molto affidabili sono *Taubin*, *fit\_ellipse* e *fitellipse*. Il primo è un metodo algebrico, il secondo è un metodo parametrico (o geometrico), mentre il terzo come già spiegato è un metodo *ibrido*, cioè usa una il metodo Bookstein per trovare le condizioni iniziali, poi risolve il problema dei minimi quadrati non lineare. La media dell'errore commesso su ogni immagine da parte d'ognuno di questi metodi è stata calcolata per poter identificare quali dei metodi ha una maggiore precisione. Di conseguenza, come detto precedentemente, nel caso in cui si debba usare un software del genere per l'OPNAV, la stima dell'errore medio permetterebbe di dare importanze differenti ai vari metodi a seconda dei risultati, e quindi ottenere una maggiore affidabilità. Questa differenziazione può essere fatta utilizzando un certo tipo di pesatura.

### 6.3 IMMAGINI ELABORATE TRAMITE ELLIPSE FITTING

Si osservino ora alcune delle immagini di Mimas ed Encelado, in cui sono stati applicati i sei metodi d'approssimazione visti fin ora:

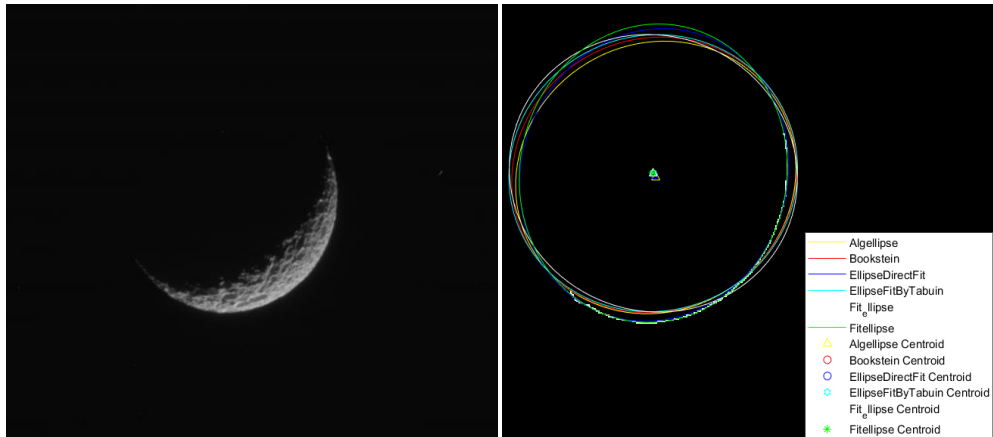


Figura 36: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni.

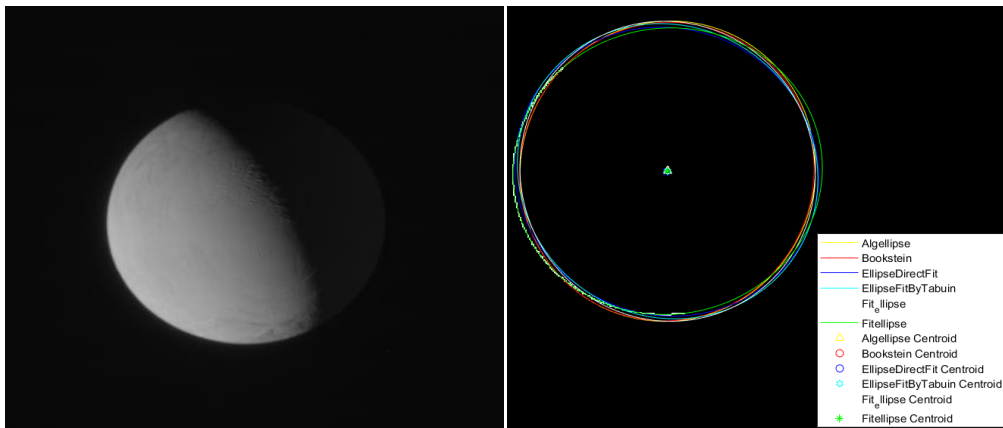


Figura 37: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Encelado, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni. (N1484519176\_1)

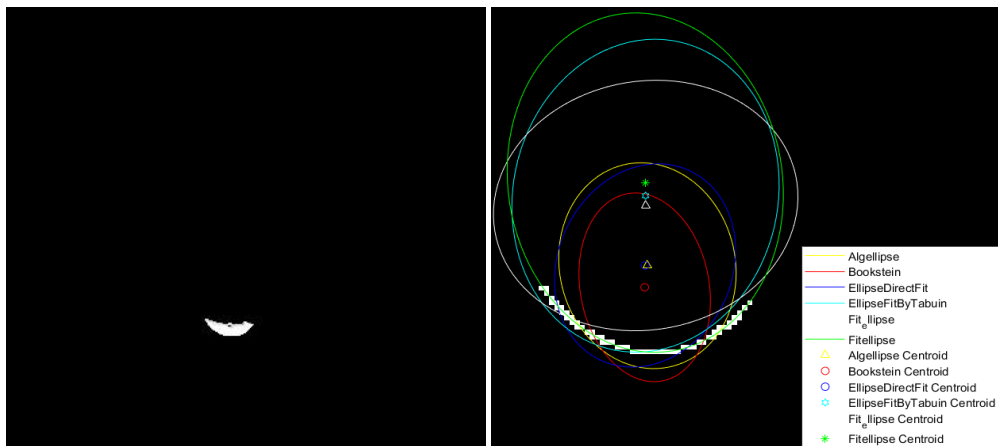


Figura 38: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni. (N1467495142\_2). (Questa immagine è stata modificata per permettere al lettore di vederne il limbo del corpo, altrimenti

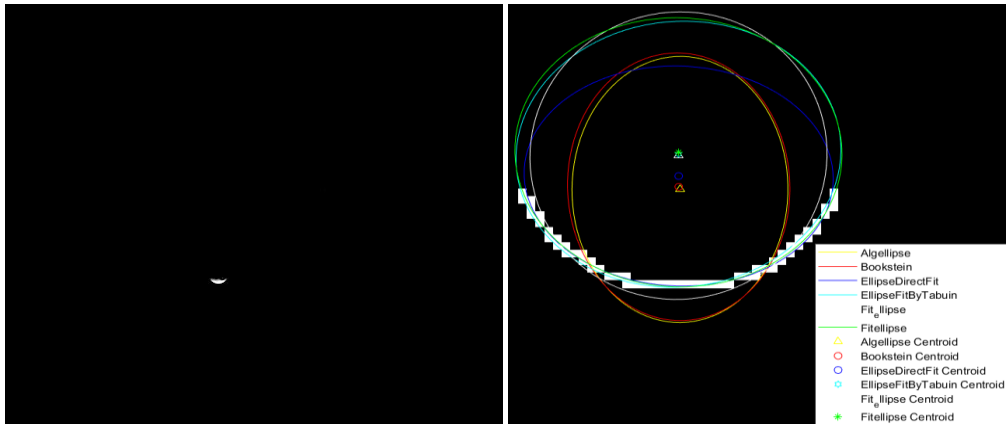


Figura 39: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni. (N1467553222\_2). (Questa immagine è stata modificata per permettere al lettore di vederne il limbo del corpo, altrimenti apparirebbe buia).

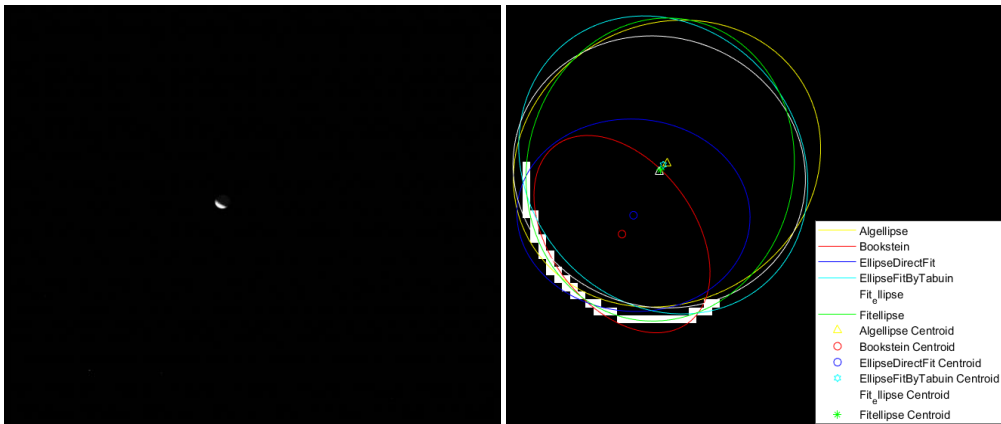


Figura 40: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni. (N1477935150\_2). (Questa immagine è stata modificata per permettere al lettore di vederne il limbo del corpo, altrimenti apparirebbe buia).

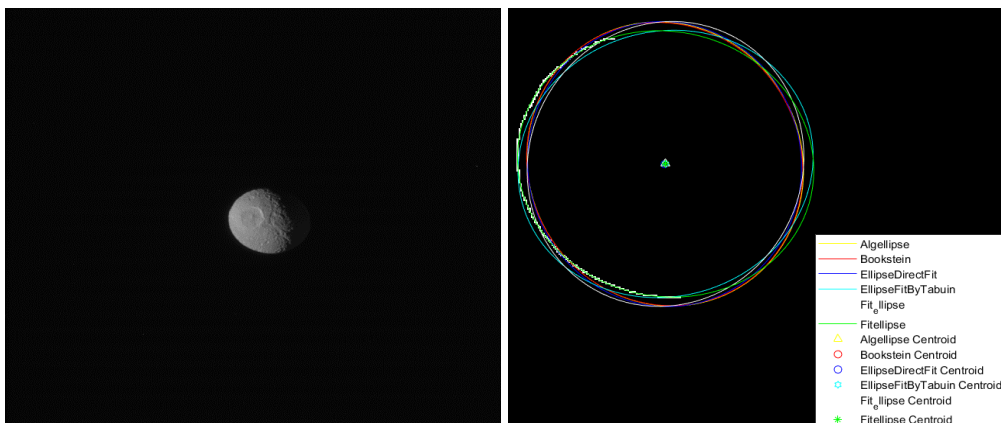


Figura 41: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni.

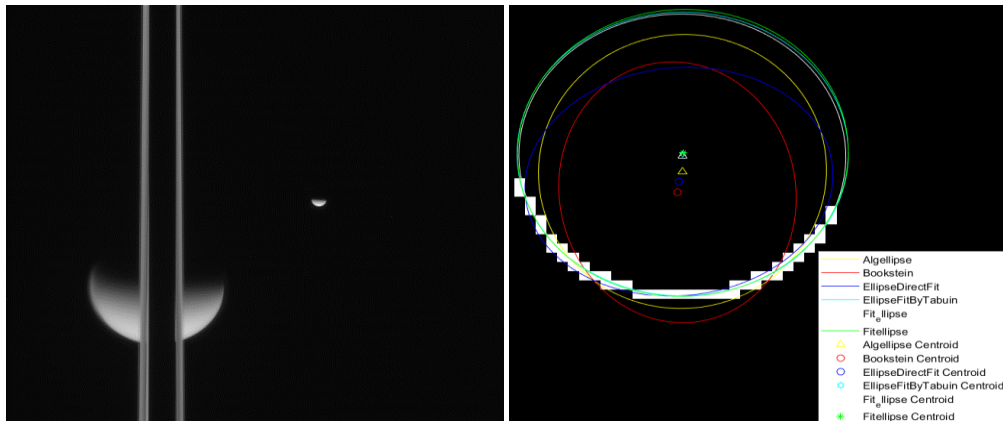


Figura 42: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Saturno con i suoi anelli a sinistra e Mimas nella parte destra, mentre a destra si possono osservare le ellissi approssimanti fornite dalle varie funzioni.

Si può osservare che per le *figure 36,37 e 41*, l'ellisse d'approssimazione è di ottima qualità da parte di tutte le funzioni e anche per quanto riguarda i centroidi, la precisione avuta da queste tecniche è molto buona, quindi, l'errore medio totale commesso è basso: 2,13 [Pixel] per la *figura 37*; 1,76 [Pixel] per quella in *figura 41*; mentre per la *figura 36* è relativamente alto, ed è di circa 8,21 [Pixel].

Questo risultato probabilmente è dovuto al fatto che le immagini ottenute, sia quelle originali dalla sonda Cassini che quelle dal metodo Canny, contengono una figura ben definita e chiara, con una grande quantità di punti e quindi informazioni per le funzioni di approssimazione.

Per le altre immagini sopra riportate, le funzioni *algellipse (MDA)*, *bookstein (MDA)* ed *ellipse direct fit (MDA)* approssimano in maniera non soddisfacente il corpo celeste, fornendo un'ellisse con proporzioni e orientamento visibilmente errati. Mentre i metodi che creano l'ellisse d'approssimazione più conforme alla realtà, con proporzioni e orientamento giusti, sono *fitellipse (MDA+MDG)*, *ellipse fit by Taubin (MDA)* e *fit\_ellipse (MDG)*.

Si può dire che nonostante l'aspetto dell'ellisse approssimante vari sensibilmente, la variazione del centroide è invece limitata.

La funzione *fit\_ellipse (MDG)* è abbastanza precisa, ma incostante, perché in alcune immagini l'ellisse approssimante non segue la forma che il limbo suggerisce, il risultato spesso ottenuto è quindi una curva che è traslata o ruotata rispetto al corpo celeste.

Sono state riportate solo alcune delle immagini utilizzate per testare il software Matlab, per mettere in mostra la qualità principali di ogni funzione, i risultati ottenuti anche nelle altre sono comunque molto simili a quelli appena visti. Però ci sono alcuni tipi di immagini, in cui anche se si spazia su tutto il range di valori d'intensità luminosa, quindi le quantità  $T_1$  e  $T_2$  di cui è stato discusso nel metodo Canny (Capitolo 3.3), il rilevamento dei bordi che si ottiene è di scarsa qualità. Di conseguenza questo comporta come risultato una costruzione dell'ellisse che se si ha fortuna è discreta, ma nella maggior parte dei casi non ha una buona qualità. Si può comunque osservare nelle immagini qui sotto (vedi *figura 43,44*) che le funzioni che meglio fittano sono *fitellipse*, quindi la funzione ibrida e *ellipse fitting by Taubin* che si basa sul trovare una soluzione lineare con il metodo dei minimi quadrati, le quali riesco a seguire (non sempre) la forma del limbo individuata dal rilevamento dei bordi.

NB: l'immagine ottenuta dall'*edge detection* non è coerente con la reale forma del corpo.

Alcuni esempi di quanto detto sopra sono i seguenti:

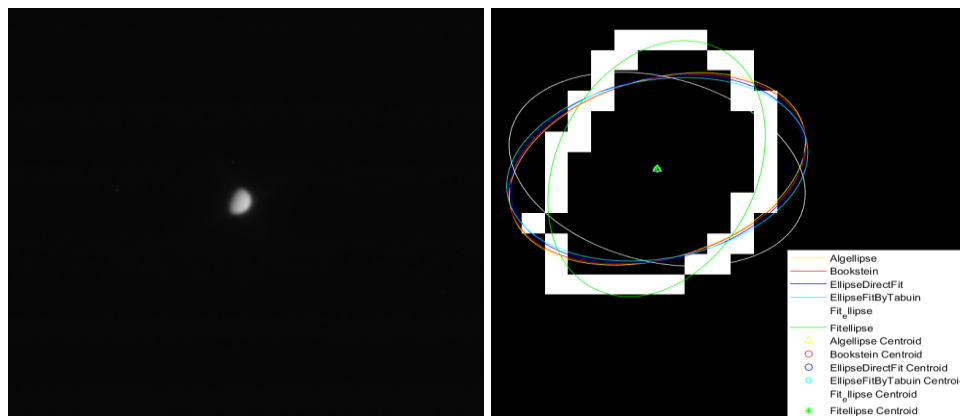


Figura 43: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Encelado, mentre a destra si può osservare il fitting fatto dalle varie funzioni. (N1466514680\_1)

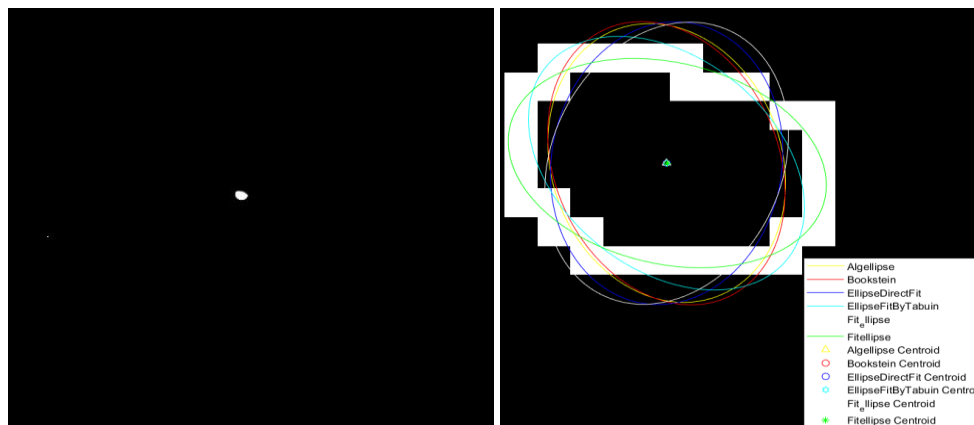


Figura 44: A sinistra si può vedere l'immagine scattata dalla sonda Cassini che ritrae Mimas, mentre a destra si può osservare il fitting fatto dalle varie funzioni. (N1466584454\_1). (Questa immagine è stata modificata per permettere al lettore di vederne il limbo del corpo, altrimenti apparirebbe buia).

Per questo tipo d'immagini, è risultato che i due metodi migliori per l'approssimazione del limbo ed il calcolo del centroide sono *ellipse fit by Taubin* e *fitellipse*, quindi un metodo di tipo algebrico e il metodo ibrido. Mentre per quanto riguarda l'errore medio totale per il calcolo del centroide si hanno ottimi risultati, in tutte le immagini simili si è riscontrato un errore inferiore ai 5 [Pixel]. Questo probabilmente è dovuto al fatto che è visibile nella foto originale più di tre quarti del corpo celeste e si ha che l'edge detection anche se non è di grande qualità risulta "ricco" di punti, e questo permette alle funzioni di basarsi su maggiori informazioni.

## 7. CONCLUSIONI

Analizzando tutte e cinquantuno le foto che sono state utilizzate per testare la qualità dell'algoritmo per la determinazione dei centroidi d'immagini di corpi celesti, di cui una buona parte è stata riportata nei Capitoli precedenti, si è dedotto che il metodo che meglio approssima l'oggetto di studio con un'ellisse proporzionata, ben orientata e congrua alla forma del corpo celeste è quello che risolve il problema dei minimi quadrati non lineare, usando una soluzione lineare come condizione iniziale, questo è stato precedentemente chiamato come metodo Ibrido o *fitellipse*. Tale metodo è suddiviso in varie fasi: inizialmente trova una soluzione lineare dei minimi quadrati, ovvero l'equazione dell'ellisse che minimizza la distanza algebrica dei punti. Questa viene poi utilizzata come stima iniziale per la risoluzione non lineare dei minimi quadrati, per il calcolo dell'ellisse che minimizza la distanza geometrica dei punti. Successivamente questa funzione procede con delle iterazioni finché la soluzione non converge. La convergenza o meno dei risultati dipende dalla qualità delle condizioni iniziali ottenute dal metodo algebrico. Questo approccio si è rivelato il più robusto e il più costante nei risultati.

Osservando l'errore medio che ogni funzione ha commesso nel calcolo dei centroidi delle cinquantuno immagini campione utilizzate, si può osservare dalla tabella in *figura 35*, che il metodo più preciso si basa in questo caso sulla risoluzione non lineare del problema dei minimi quadrati (*fit\_ellipse*), con un errore medio di 2,8202 [Pixel]. Mentre il secondo metodo con errore più basso è basato sulla ricerca dell'equazione dell'ellisse che minimizza la distanza algebrica, cioè soluzione lineare del problema, con un risultato di 2,8657 [Pixel]. Il metodo ibrido come visto non è quello che commette l'errore medio minimo ma risulta tuttavia vicino al valore più basso, 3,6895 [Pixel], con uno scarto di 0,8693 [Pixel] rispetto al primo citato.

Quindi il risultato migliore per il calcolo del centroide è stato ottenuto con un metodo di risoluzione non lineare del problema ai minimi quadrati (*fit\_ellipse*), che ovviamente anch'esso utilizza delle condizioni iniziali per la sua risoluzione, ma ottenute in maniera differente rispetto a quello



ibrido. Tuttavia vedendo la differenza d'errore medio ottenuto tra quest'ultima funzione citata e quella ibrida e osservando la qualità dell'ellisse d'approssimazione si può concludere questo confronto dicendo che il metodo che ha restituito i migliori risultati ed è stato il più costante è quello che utilizza come condizione iniziale una soluzione del metodo lineare, di default usa quella con vincolo di Bookstein (vedi Capitolo 4.6), per la risoluzione del metodo dei minimi quadrati non lineare, questo tipo di problema è iterativo, più complesso ma come si è visto più preciso.

Un altro aspetto che si è notato durante il test di questo algoritmo è che solitamente nelle immagini in cui si ha un una figura chiara, ben definita e con una buona parte del limbo visibile i metodi risultano nella media precisi, sia nel calcolo dell'ellisse approssimante, che nel calcolo del centroide. Questo vale per entrambe le tipologie di algoritmi, sia quelli che minimizzano la distanza algebrica sia quelli che minimizzano la distanza geometrica (vedi esempi in *figure 36,37 e 41*). Mentre per le figure in cui i bordi del corpo celeste non sono ben chiari o risultano pochi punti presenti dal rilevamento del contorno dell'immagine, i metodi più accurati sono generalmente quelli con una risoluzione non lineare del problema dei minimi quadrati.

Ovviamente data la vastità della qualità e della tipologia delle immagini che ritraggono corpi celesti, la scelta che si effettua del metodo anche se ponderata, potrebbe risultare non idonea, questo perché non c'è nessuna legge che assicura la qualità della regressione. Quindi anche se il metodo ibrido, in media, risulta quello con maggiore precisione, non è detto che anche se si ha un limbo chiaro, nitido e lungo l'ellisse d'approssimazione sia di buona qualità.

Nel caso in cui si debba sviluppare un algoritmo al fine di creare sistema di OPNAV, il consiglio è di non utilizzare un solo metodo, questo perché alcuni hanno una maggiore precisione in certi casi sul calcolo del centroide, mentre altri nell'approssimazione del limbo. Tutto questo perché il problema dell'elaborazione di immagini di questo tipo non risulta lineare, e quindi con una certa ridondanza ci si può assicurare una precisione maggiore.

## RINGRAZIAMENTI

Desidero ringraziare in primo luogo il *Professor Dott. Ing. Dario Modenini* per avermi supportato durante lo svolgimento di questa tesi con gentilezza e disponibilità.

Un ringraziamento va anche al *Professo Dott. Ing. Marco Zannoni* per i consigli preziosi che mi ha fornito

Desidero inoltre ringraziare tutti i professori che in questi tre anni mi hanno aiutato ad arricchire il mio bagaglio culturale con interesse e dedizione.

Infine, desidero ringraziare mia madre, la mia morosa, mia sorella e mio padre che mi hanno sempre aiutato lungo questo percorso soprattutto nei momenti più impegnativi, ed è per questo che dedico a loro la mia tesi.

## BIBLIOGRAFIA

- [1]. *Astrometric reduction of Cassini ISS images of the Saturnian*. **R. Tajeddine, N. J. Cooper, V. Lainey, S. Charnoz, D. Murray**. 1, s.l. : Astronomy & Astrophysics, 2013.
- [2]. *Fitting of Circles and Ellipses Least Square Solution*. **W. Gander, Gene H. Golub, Rolf Strebel**. 4, s.l. : BIT Numerical Mathematics, 1994, Vol. 34.
- [3]. *Estimation of Planar Curves, Surface and Non-Planar Space Curves Defined by Implicit Equation with applications to edge and range image segmentation*. **Taubin, G.** 1991, Vol. 13.

## FONTE IMMAGINI

- [a] [http://roveri.faculty.polimi.it/wp-content/uploads/4\\_Elaborazione\\_delle\\_immagini.pdf](http://roveri.faculty.polimi.it/wp-content/uploads/4_Elaborazione_delle_immagini.pdf)
- [b] [https://commons.wikimedia.org/wiki/File:EM\\_Spectrum\\_Properties\\_it.svg](https://commons.wikimedia.org/wiki/File:EM_Spectrum_Properties_it.svg)
- [c] <https://pds-imaging.jpl.nasa.gov/volumes/iss.html>
- [d] [http://www.dmi.unict.it/~battiato/mm1112/Parte%206\\_1%20-%20Filtraggio%20nel%20Dominio%20della%20Frequenza.pdf](http://www.dmi.unict.it/~battiato/mm1112/Parte%206_1%20-%20Filtraggio%20nel%20Dominio%20della%20Frequenza.pdf)
- [e] [https://www.researchgate.net/profile/Roberto\\_Guarnieri/publication/278126258\\_Filtraggio\\_delle\\_immagini\\_nel\\_dominio\\_della\\_frequenza\\_e\\_loro\\_applicazione\\_in\\_ambiente\\_Matlab/links/557c199608aec87640db3591/Filtraggio-delle-immagini-nel-dominio-della-frequenza-e-loro-applicazione-in-ambiente-Matlab.pdf](https://www.researchgate.net/profile/Roberto_Guarnieri/publication/278126258_Filtraggio_delle_immagini_nel_dominio_della_frequenza_e_loro_applicazione_in_ambiente_Matlab/links/557c199608aec87640db3591/Filtraggio-delle-immagini-nel-dominio-della-frequenza-e-loro-applicazione-in-ambiente-Matlab.pdf)
- [f] <https://sites.google.com/site/rcartvision/derivate-di-immagini>
- [g] <http://bias.csr.unibo.it/fei/Dispense/3%20-%20FEI%20-%20Estrazione%20dei%20Bordi%20e%20Segmentazione.pdf>
- [h] <https://sites.google.com/site/rcartvision/derivate-di-immagini>
- [i] <https://it.wikipedia.org/wiki/Ellisse>
- [l] [https://en.wikipedia.org/wiki/Pinhole\\_camera\\_model](https://en.wikipedia.org/wiki/Pinhole_camera_model)
- [m] JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS, Vol. 38, No. 2, February 2015, Optical Navigation Using Planet's Centroid, and Apparent Diameter in Image, John A. Christian, \* West Virginia University, Morgantown, West Virginia 26506, DOI: 10.2514/1.G000872

## **SITOGRAFIA**

[A] <https://pds-imaging.jpl.nasa.gov/volumes/iss.html>

[B] <http://bjj.mmedia.is/utls/img2png/>