Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Calcolo di ologrammi con Octave e Matlab

Relatore:
Prof. Marco Cuffiani

Presentata da:
Alessandro Fuschi

Anno Accademico 2016/2017

Sommario

Questo scritto ha lo scopo di illustrare i principi dell'olografia digitale. Nella prima parte descriverò le basi dell'olografia per poi analizzare i processi numerici, svolti al computer, che permettono di simulare i fenomeni della diffrazione della luce incidente su di un ostacolo. Nella seconda parte esporrò un programma illustrativo, il quale ripercorre i passaggi fondamentali della *Computer Generated Holography*.
Per la scrittura dei programmi utilizzerò **OCTAVE**, un'applicazione software open-source per l'analisi numerica in gran parte compatibile con MATLAB. Octave ha un insieme di funzionalità fornite per il calcolo matriciale ed è scritto in conformità dello standard POSIX nel linguaggio C++ usando le librerie Standard Template Library.

Indice

Introduzione

L'olografia è un processo di registrazione e ricostruzione dell'immagine ottica di un oggetto, quindi della luce stessa che diffonde. La peculiarità di questa tecnica è quella di saper elaborare punto per punto, sia l'ampiezza sia la fase dell'onda elettromagnetica (la luce), a differenza di una fotografia che contiene solo l'informazione della distribuzione dell'intensità. Il principio su cui si basa è che l'informazione ottica dell'immagine di un oggetto può essere ricavata dalla figura di interferenza prodotta dalla sovrapposizione di due onde coerenti, quindi generate da un laser: una diffusa dall'oggetto, detta onda oggetto, mentre l'altra indisturbata, detta onda di riferimento. L'ologramma è proprio la figura d'interferenza registrata su di una superfice piana, anche se contiene l'informazione tridimensionale da cui potrò ricostruire l'immagine.
La tecnica olografica fu scoperta dallo scienziato ungherese Dennis Gabor nel 1947, il quale realizzò i primi ologrammi con l'ausilio della luce verde dello spettro di una lampada a vapori di mercurio[1]; tuttavia le immagini erano molto disturbate a causa della scarsa coerenza della luce e bisognò aspettare la costruzione dei primi laser agli inizi degli anni 60. Inoltre nella tecnica sviluppata da Gabor l'onda oggetto e l'onda di riferimento si propagavano sullo stesso asse e, così facendo, in fase di ricostruzione l'immagine reale e virtuale dell'oggetto si formano una dietro l'altra, disturbandosi.
Un importante sviluppo dell'olografia fu portato da Leith ed Upatnieks nel 1964, i quali introdussero un'onda di riferimento inclinata rispetto al piano di registrazione; con questo stratagemma durante la fase di ricostruzione le immagini virtuale e reale si separano spazialmente in modo da poter osservare più nitidamente l'immagine dell'oggetto [2].
Nel 1967 Goodman e Lawrence ipotizzarono la cattura elettronica dell'ologramma e la successiva ricostruzione numerica dell'immagine al computer [3], il processo all'epoca richiedeva un enorme sforzo computazionale ed impiegava svariate ore per essere svolto. Negli anni '90 grazie al grande sviluppo dei computer si riuscì a velocizzare il processo ideato da Goodman e Lawrence, portando i tempi di elaborazione nell'ordine dei secondi.
Le applicazioni dell'olografia odierna si possono suddividere in due branche principali, la *digital holography* (DH) e la *computer generated holograms* (CGH). La prima utilizza i CCD per registrare l'ologramma per poi ricostruire numericamente l'immagine e trova le sue applicazioni nella microscopia digitale, nell'analisi delle particelle, nell'interferometria e nella misura delle deformazioni; mentre la CGH porta tutto il processo olografico nel computer dove l'oggetto diventa un modello numerico ed è alla base delle televisioni 3D[4].

Capitolo 1

Olografia

Nel primo capitolo illustrerò le nozioni generali dell'olografia, sopratutto il processo di registrazione e ricostruzione; questi rimangono teoricamente uguali fra l'olografia analogica, la DH e la CGH.

Durante tutto lo scritto utilizzerò le approssimazioni per cui le onde prese in considerazione saranno perfettamente coerenti e monocromatiche (senzaciò dovrei argomentare quali deformazioni porterebbero all'immagine finale), inoltre le onde coerenti saranno sempre onde piane.

1.1 Registrazione e visualizzazione di ologrammi

La ricostruzione dell'immagine di un oggetto è sempre composta da due passaggi fondamentali, la prima è la registrazione dell'ologramma per poi -in seconda fase- andare a ricostruire l'immagine.
In primis dovrò registrare le intensità delle finissime frange di interferenza date dalla sovrapposizione dell'onda oggetto con l'onda di riferimento grazie ad un dispositivo in grado di farlo (è proprio la scelta del dispositivo che diversifica l'olografia analogica da quella digitale).
Se chiamo il fronte dell'onda diffusa dall'oggetto $O(x, y)$ mentre quella indisturbata la chiamo onda di riferimento $R(x, y)$, il dispositivo dovrà registrare la distribuzione dell'intensità delle due onde sovrapposte, data da:

$$\begin{aligned} H(x,y) &= |R(x,y) + O(x,y)|^2 = \\ &= |R(x,y)|^2 + |O(x,y)|^2 + R^*(x,y)O(x,y) + R(x,y)O(x,y)^*. \end{aligned} \tag{1.1}$$

Per ricostruire l'immagine punto per punto bisogna far incidere sull'ologramma, quindi sulla figura d'interferenza registrata nella prima fase, un'onda coerente più simile possibile a quella generata dal laser iniziale (se non il laser iniziale), sempre per evitare deformazioni dell'immagine.

Quando quest'onda, che chiamerò onda di ricostruzione $R_c(x,y)$, andrà a colpire l'ologramma si avranno queste componenti:

$$\begin{aligned}\psi_{finale}(x,y) =& R_c(x,y)\,|R(x,y)|^2 + R_c(x,y)\,|O(x,y)|^2 + R_c(x,y)R(x,y)^*O(x,y) + \\ &+ R_c(x,y)R(x,y)O(x,y)^*.\end{aligned} \tag{1.2}$$

Il primo ed il secondo termine sono detti l'ordine zero e formano un alone chiaro che va a disturbare l'immagine.
Il terzo termine è proporzionale all'ampiezza dell onda diffusa dall'oggetto $O(x,y)$, e costituisce *l'immagine virtuale* che si va a formare prima del dispositivo di registrazione. Infine il quarto termine proporzionale ad $O(x,y)^*$ costruisce l'immagine reale, si forma alla stessa distanza dell'immagine virtuale ma solo dalla parte opposta del dispositivo di registrazione.
Il processo di registrazione e visualizzazione dell'ologramma appena descritto fu presentato da Denis Gabor, per il quale vinse anche in nobel; è stato il primo ad essere attuato anche se presenta limitazioni importanti nella qualità del'immagine ricostruita, infatti sià l'ordine zero sià le immagini virtuale e reale si trovano sullo stesso asse, disturbandosi a vicenda. Per ovviare a questo problema si pensò di inclinare l'onda di riferimento di un angolo θ rispetto al piano di registrazione, così facendo avremo che le immagini virtuale e reale e lo zero-order si separeranno durante la ricostruzione. Questo procedimento si chiama *"olografia fuori asse"* ed è di fondamentale importanza per la moderna olografia.

1.1.1 Olografia fuori asse

Chiamando sempre l'onda di riferimento $R(x,y)$ andrò a moltiplicargli un esponenziale complesso per introdurre l'inclinazione dell'angolo θ.

$$R(x,y) = R_0 e^{ik_0 sen\theta(x+y)}. \tag{1.3}$$

con R_0=ampiezza dell'onda
k_0=vettore d'onda

L'ologramma avrà un equazione del tipo:

$$\begin{aligned}H(x,y) &= |O(x,y) + R(x,y)|^2 = \left|O(x,y) + R_0 e^{ik_0 sen\theta(x+y)}\right|^2 \\ &= |O(x,y)|^2 + \left|R_0 e^{ik_0 sen\theta(x+y)}\right|^2 + O(x,y)^* R_0 e^{ik_0 sen\theta(x+y)} + O(x,y)\left(R_0 e^{ik_0 sen\theta(x+y)}\right)^* \\ &= t_1 + t_2 + t_3 + t_4.\end{aligned} \tag{1.4}$$

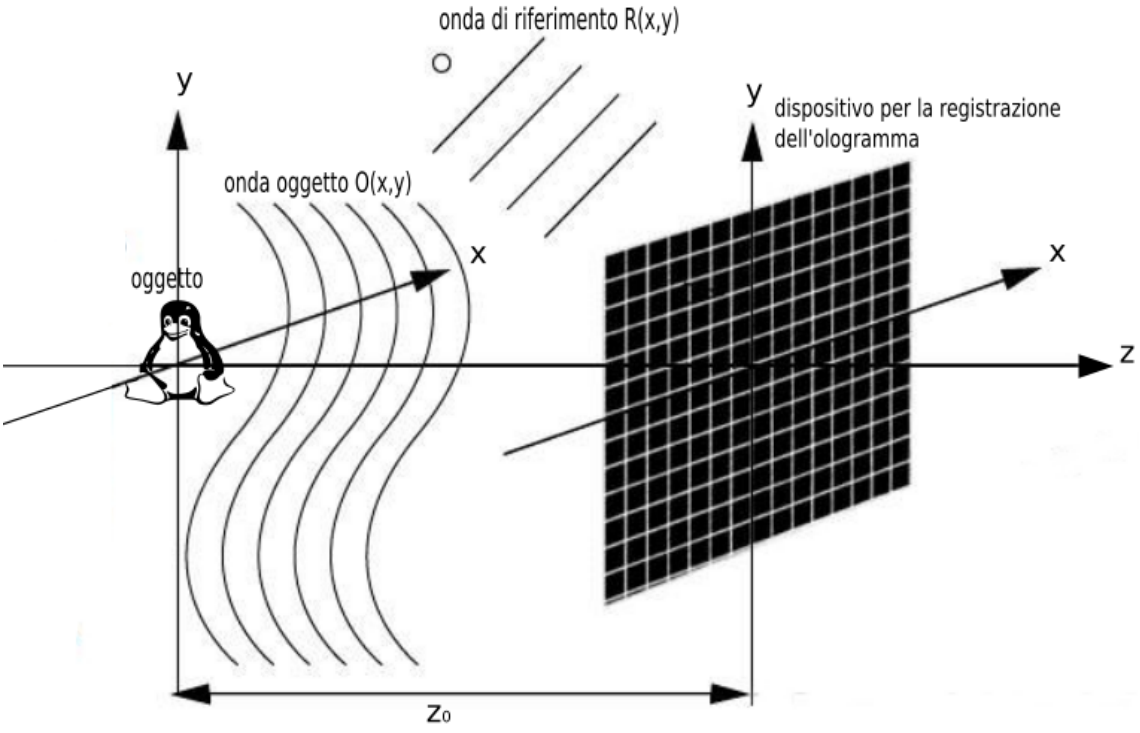

Figura 1.1: *Apparato per l'olografia fuori asse in cui l'oggetto è in asse con il dispositivo di registrazione mentre l'onda di riferimento è inclinata.*

Considerando un'onda di ricostruzione $R_c(x, y)$ che si propaga lungo l'asse z, perpendicolarmente al piano di registrazione, avrò che $R_c(x, y) = R_{0c}$ l'ampiezza dell'onda.
Così facendo otterò i seguenti termini:

$$\psi^{(1)}_{finale} = R_{oc}t_1 = R_{oc}\left|O(x,y)\right|^2, \tag{1.5a}$$

$$\psi^{(2)}_{finale} = R_{oc}t_2 = R_{oc}\left|R_0 e^{ik_0 sen\theta(x+y)}\right|^2 = R_{0c}R_0^2, \tag{1.5b}$$

$$\psi^{(3)}_{finale} = R_{oc}t_3 = R_{oc}O(x,y)^* R_0 e^{ik_0 sen\theta(x+y)}, \tag{1.5c}$$

$$\psi^{(4)}_{finale} = R_{oc}t_4 = R_{oc}O(x,y)\left(R_0 e^{ik_0 sen\theta(x+y)}\right)^*. \tag{1.5d}$$

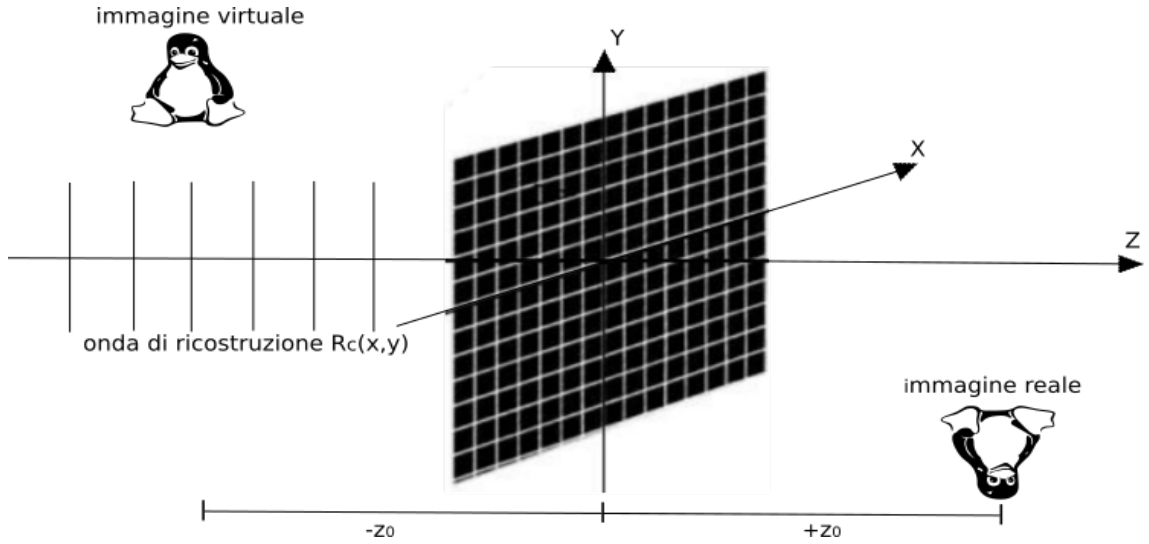

Figura 1.2:*Ricostruzione delle immagini reale e virtuale nell'apparato fuori asse illuminando l'ologramma precedentemente registrato con un onda piana perpendicolare all'ologramma stesso.*

Utilizzando questo procedimento, come illustrato nelle **Figure 1.1** e **1.2**, riusciamo a dividere le immagini virtuale e reale e lo zero order, il quale è composto dai primi due termini, migliorando la qualità delle immagini ricostruite.
Lo zero-order continua a propagarsi perpendicolarmente al piano di registrazione, i termini $\psi^{(3)}_{finale}$ e $\psi^{(4)}_{finale}$ ottengono una deviazione di $sen\theta$ e $-sen\theta$ rispettivamente.
Nell'immagine precedente ho preso come origine del sistema di riferimento il centro del dispositivo di registrazione, quindi l'immagine virtuale che ho posizionato in $-z_0$ si va a formare dov'era l'oggetto iniziale, solo spostata sul piano x,y[5].

In questo scritto utlizzerò sempre l'apparato fuori asse per illustrare i principi dell'olografia anche se esisterebbero svariati altri apparati per la produzione di ologrammi, ognuno con i suoi pro e contro.
Infine dovrei discutere sul minimo angolo di incidenza dell'onda di riferimento θ per cui le immagini virtuale e reale e lo zero-order si dividano; questo argomento lo riprenderò dopo quando avrò presentato un formulismo matematico in grado di descrivere completamente gli argomenti trattati.

1.2 Differenze tra olografia analogica, DH e CGH

I passaggi teorici rimangono sempre quelli descritti nella sezione precedente (prima si registra l'ologramma ed in seconda fase si ricostruisce l'immagine iniziale), anche se tra le diverse tecniche -analogico, DH, CGH- si utilizzano dei metodi diversi per i passaggi del processo olografico che portano a delle applicazioni diverse.

1.2.1 Olografia analogica

Per l'olografia analogica si utilizzano dei vetrini cosparsi di un gel fotosensibile, i quali riescono modificare la loro trasmittanza se esposti alla luce (nel nostro caso di un laser siccome abbiamo bisogno di una luce coerente) per un determinato intervallo temporale, che dipende dal gel usato; in pratica si imprimono su questo gel le frange di interferenza dell'ologramma, similmente a come si imprime una fotografia sulla pellicola. Consecutivamente ci sono vari passaggi chimici che permettono di bloccare la trasmittanza del vetrino anche se esposto ad altra luce.
Infine si illumina il vetrino con il laser in modo da formare le immagini virtuale e reale.
Utilizzando questo procedimento siamo riusciti a produrre degli ologrammi analogici in laboratorio, i quali conservando sià l'informazione dell'ampiezza sià l'informazione della fase dell'onda proveniente dall'oggetto restituiscono un immagine tridimensionale.

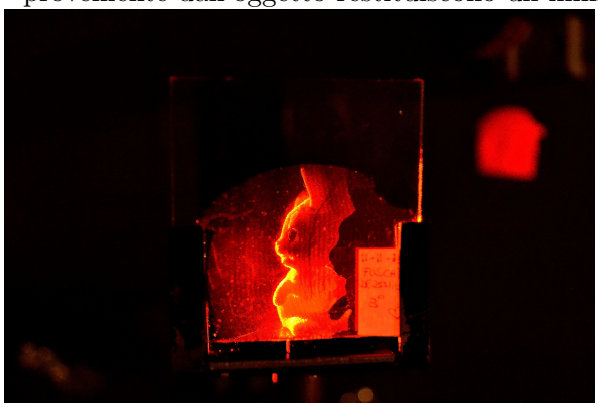

Figura 1.3.1

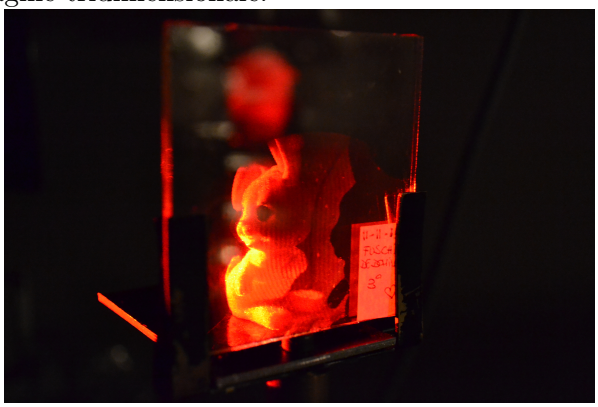

Figura 1.3.2

Le due foto mostrano un ologramma analogico prodotto da me insieme ad una mia collega in laboratorio in cui abbiamo ricostruito l'immagine di un pupazzo in plastica.

Dalle foto **1.3.1** e **1.3.2** si riesce a vedere l'effetto di parallasse che si produce nell'immagine riscostruita, infatti spostando il punto di osservazione si riescono a vedere particolari diversi dell'oggetto[6].

1.2.2 Olografia digitale (DH)

In questo metodo per registrare l'ologramma si utlizzano i CCD, *Charge-Coupled Device*; sono dei circuiti integrati formati da un griglia di semiconduttori in grado di accumulare carica elettrica proporzionale all'intensità dell'onda elettromagnetica che li colpisce.
Questi semiconduttori hanno un grandezza caratteristica di qualche decina di *nm* e grazie alle loro piccole dimensioni riescono ad immagazinare tutta l'informazione dell'onda elettromagnetica analizzata.
La capacità o meno di sapere immagazzinare tutta l'informazione di un'onda da parte dei CCD verrà analizzato successivamente, è un risultato a cui si puà arrivare solo dopo aver enunciato il teorema del campionamento di Nyquist-Shannon.

L'idea di fondo della olografia digitale è quella di sostituire il dispositivo di registrazione analogico con i CCD, i quali andranno a formare una matrice di valori discreti su cui campioneremo i valori dell'intensità dell'ologramma.
Successivamente alla registrazione dell'ologramma sul CCD si simula al computer la diffrazione prodotta dall'onda di ricostruzione sull'ologramma, ottenendo le immagini virtuale e reale del'oggetto considerato.

1.2.3 Computer generated holography (CGH)

Tutto il processo viene svolto attraverso metodi numerici al computer e l'oggetto viene descritto con dei modelli numerici. I passi principali di questa tecnica sono di simulare la diffrazione della luce diffusa dall'oggetto fino al piano di registrazione dovre andrò a sommargli l'onda di riferimento, anch'essa ricreata al computer (anche in questo caso dovrò rispettare le condizioni del teorema del campionamento, sennò le simulazioni della diffrazione della luce daranno dei risultati errati). Dopo aver salvato l'ologramma in una matrice di valori discreti si simula ancora una volta la diffrazione prodotto dall'onda di ricostruzione sull'ologramma per ricreare l'oggetto iniziale.

Capitolo 2

Teoria scalare di diffrazione

L'argomento che tratterò in questo capitolo è di fondamentale importanza, in quanto presenterò una descrizione più formale degli argomenti visti fino ad ora ed analizzerò il fenomeno della diffrazione della luce da un oggetto.
In pratica fornirò una descrizione matematica di come si propaga la luce del laser dopo aver incontrato un ostacolo; nel nostro caso applicherò questa teoria sia per conoscere la funzione d'onda dell'oggetto $O(x, y)$ sul piano di registrazione dell'ologramma sia poi per ricostruire le immagini virtuale e reale.
Più precisamente, il concetto che tratterò è come un'onda coerente si propaghi dopo aver colpito un oggetto ad una precisa distanza dall'oggetto stesso.

2.1 Equazione d'onda scalare

Durante tutta l'esposizione tratteremo la luce come un'onda poichè i fenomeni che noi andremo ad osservare saranno l'intereferenza e la diffrazione di onde elettromagnetiche, proprietà intrisiche di qualsiasi onda.

Inizieremo dalle equazioni di Maxwell nel caso dello spazio privo di cariche e di correnti elettriche:

$$\vec{\nabla} \cdot \vec{E} = 0 \qquad \vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t},$$

$$\vec{\nabla} \cdot \vec{B} = 0 \qquad \vec{\nabla} \times \vec{B} = \frac{1}{c^2}\frac{\partial \vec{E}}{\partial t}.$$

Utilizzando c nelle equazioni di Maxwell ho sottointeso che sto descrivendo l'elettromagnetismo in un spazio lineare, omogeneo ed isotropo, che è quello in cui ci troveremo per i fenomeni osservati.

Andando ad applicare ancora un rotore alla relazione $\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ e conoscendo le identità vettoriali (1) $\vec{\nabla} \times (\vec{\nabla} \times \vec{X}) = \vec{\nabla}(\vec{\nabla} \cdot \vec{X}) - \vec{\nabla}^2 \vec{X}$, (2) $\vec{\nabla} \times (\vec{\nabla} \vec{X}) = 0$, con $\vec{X}$ un qualsiasi vettore; posso ricavare l'equazione d'onda in $\vec{E}$:

$$\vec{\nabla}^2 \vec{E} - \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2} = 0. \tag{2.1}$$

Riscrivo il campo elettrico nelle sue componenti cartesiane $\vec{E} = E_x \widehat{x} + E_y \widehat{y} + E_z \widehat{z}$, con $\widehat{x}, \widehat{y}, \widehat{z}$ i versori degli assi. Sostituendo nella 2.1 ottengo:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) (E_x \widehat{x} + E_y \widehat{y} + E_z \widehat{z}) = \frac{1}{c^2} \frac{\partial^2}{\partial t^2} (E_x \widehat{x} + E_y \widehat{y} + E_z \widehat{z}) . \tag{2.2}$$

Confrontando solo la componente $\widehat{x}$ in entrambi i membri, ottengo:

$$\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 E_x}{\partial t^2}. \tag{2.3}$$

Nello stesso modo posso ricavare le relazioni di $\vec{E_y}$ e $\vec{E_z}$. Infine andrò a scrivere i set di equazioni trovate per $\vec{E_x}\, \vec{E_y}\, \vec{E_z}$ in una forma più compatta:

$$\nabla^2 \psi = \frac{1}{c^2} \frac{\partial^2 \psi}{\partial t^2} \qquad \psi = E_x, E_y, E_z. \tag{2.4}$$

ψ rappresenta una delle componenti del campo elettrico.
L'equazione 2.4 viene chiamata *equazione d'onda scalare* e successivamente andremo a descrivere il campo elettrico in tal modo.

L'equazione delle onde piane è una soluzione della 2.4, infatti per una oscillazione armonica di frequenza ω_0 la soluzione piana in coordinate cartesiana è:
$\psi(x, y, z, t) = A_\psi e^{i(\omega_0 t - \vec{k_0} \cdot \vec{R})}$
con A_ψ l'ampiezza dell'onda, $\vec{k_0} = k_{0x} \widehat{x} + k_{0y} \widehat{y} + k_{0z} \widehat{z}$ il vettore d'onda, $\vec{R} = x \widehat{x} + y \widehat{y} + z \widehat{z}$ il vettore posizione.
Il modulo di $\vec{k_0}$ è detto numero d'onda ed ha valore $\left| \vec{k_0} \right| = \frac{2\pi}{\lambda}$ con λ la lunghezza d'onda.
Se l'onda piana si propaga lungo l'asse z si ottiene l'equazione:

$$\psi(x, y, z, t) = A_\psi e^{i(\omega_0 t - |k_0| z)}. \tag{2.5}$$

2.2 Teoria scalare di diffrazione

In questa sezione analizzerò come si distribuisce il campo dopo aver incontrato un ostacolo sul suo percorso ad una precisa distanza.

Assumiamo che un'onda piana, che si propaga lungo z, incontri un ostacolo in due dimensioni in cui ogni punto può far passare la luce completamente, parzialmente o bloccarla del tutto; possiamo descrive questa proprietà con una funzione di trasmittanza della luce $t(x, y)$. Ponendo l'ostacolo sul piano xy in z=0 (come in figura) avremo che l'onda piana appena prima di esso avrà equazione $\psi(x, y, z = 0^-, t) = A_\psi e^{i\omega_0 t}$, mentre appena dopo l'ostacolo:

$$\psi(x, y, z = 0^+, t) = A_\psi t(x, y) e^{i\omega_0 t} = \psi_{p0}(x, y) e^{i\omega_0 t}, \tag{2.6}$$

$\psi_{p0} = A_\psi t(x, y)$ viene chiamata *ampiezza complessa*

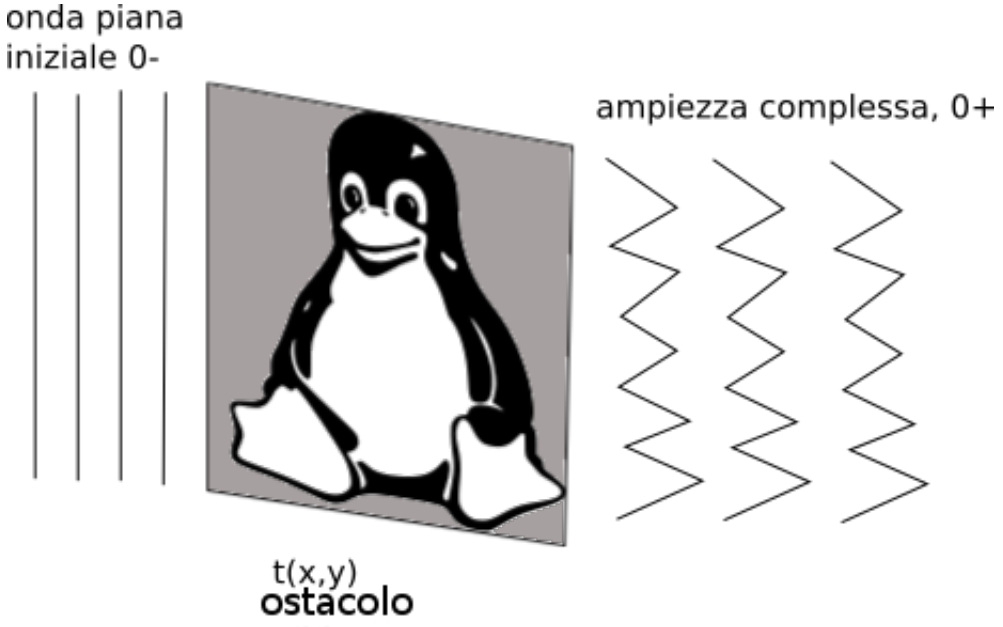

Figura 2.1: *Ampiezza complessa; cioè l'onda coerente dopo l'incontro con l'ostacolo che in questo caso è un'immagine bidimensionale, descritta da una funzione di trasmittanza della luce.*
Nella DH questo concetto viene utilizzato quando si simula la diffrazione della onda di ricostruzione sull'ologramma, mentre nella CGH andrò ad applicarlo anche per simulare la diffrazione dell'oggetto oltre che quella sull'ologramma.

Per trovare la distribuzione del campo ad una distanza z_0 dall'ostacolo andiamo ad analizzare l'equazione:

$$\psi(x, y, z_0, t) = \psi_p(x, y; z_0) e^{i\omega_0 t}, \tag{2.7}$$

dove $\psi(x, y; z_0)$ è l'equazione incognita che può essere calcolata partendo dalla condizione iniziale $\psi(x, y; 0^+)$ conosciuta.

Vado a sostituire le funzioni d'onda scalari 2.4 in $\psi_p(x, y; z_0)$ ed ottengo le *equazioni di Helmholtz*:

$$\frac{\partial^2 \psi_p}{\partial x^2} + \frac{\partial^2 \psi_p}{\partial y^2} + \frac{\partial^2 \psi_p}{\partial z^2} + |k_0|^2 \psi_p = 0. \tag{2.8}$$

Per la soluzione dell'equazione 2.8 dovrò introdurre le trasformate di Fourier, uno strumento matematico molto importante.

Trasformate di Fourier

In questa sezione riporterò solo le nozioni fondamentali delle trasformate ed in particolare illustrerò le proprietà che utilizzeremo in seguito.

La trasformata di Fourier è un operatore che agisce su uno spazio di funzioni a decresenza rapida; in pratica applicando la trasformata ad un'onda è possibile scomporla nelle sue componenti sinusoidali.
Nel nostro caso lo andremo ad applicare alla nostra funzione che si trova nello spazio cartesiano per portarla nello spazio delle frequenze spaziali. Una volta risolta nello spazio delle frequenze applicherò la trasformata inversa per riportare la soluzione nello spazio cartesiano.

La trasformata di Fourier in due dimensioni applicata ad una generica funzione f(x,y) è così definita:

$$\mathcal{F}\{f(x, y)\} = F(k_x, k_y) = \iint_{-\infty}^{\infty} f(x, y) e^{i(k_x x + k_y y)} dx dy, \tag{2.9}$$

mentre le trasformata inversa è:

$$\mathcal{F}^{-1}\{F(k_x, k_y)\} = f(x, y) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} f(x, y) e^{-i(k_x x + k_y y)} dk_x dk_y. \tag{2.10}$$

con k_x, k_y frequenze spaziali di grandezza (radianti/lunghezza).

Le proprietà che andremo ad utilizzare delle trasformate sono le seguenti:

funzione originale	trasformata di fourier
f(x,y)	$F(k_x, k_y)$
$f(x - x_0, y - y_0)$	$F(k_x, k_y)e^{i(k_x x_0 + k_y y_0)}$
$f(ax, by)$	$\frac{1}{\|ab\|} f\left(\frac{k_x}{a}, \frac{k_y}{b}\right)$
$\frac{\partial f(x,y)}{\partial x}$	-$ik_x F(k_x, k_y)$
$f_1 * f_2 = \int\int_{-\infty}^{\infty} f_1(x', y') f_2(x - x', y - y') dx' dy'$	$F_1(k_x, k_y) F_2(k_x, k_y)$
$f_1 \otimes f_2 = \int\int_{-\infty}^{\infty} f_1(x', y') f_2(x + x', y + y') dx' dy'$	$F_1^*(k_x, k_y) F_2(k_x, k_y)$
1	$4\pi\delta(x, y) = \int\int_{-\infty}^{\infty} 1 e^{i(\pm k_x x \pm k_y y)} dk_x dk_y$
$\delta(x, y)$	1
rect(x,y)	$sinc(\frac{k_x}{2\pi}, \frac{k_y}{2\pi})$

Le funzioni rect(x) e sinc(x) sono così definite:

$$rect(x) = \begin{cases} 1 & \text{se } |x| < 1/2, \\ 0 & \text{altrove.} \end{cases}$$

mentre sinc(x):

$$sinc(x) = \frac{sin(\pi x)}{\pi x}.$$

La funzione rect(x) verrà utilizzata per descrivere il campionamento di un segnale analogico da parte di un singolo pixel; invece $sinc(\frac{kx}{2\pi})$, che è la trasformata di fourier di rect(x), ci servirà per analizzare le condizioni portate dalla digitalizzazione del segnale.

Ritornando alla soluzione dell'equazione 2.8, vado ad applicare la trasformata di fourier in due dimensioni in x,y, utlizzando la notazione $\mathcal{F}\{\psi(x, y)\} = \Psi(k_x, k_y)$.

$$\mathcal{F}\left\{\frac{\partial^2 \psi_p}{\partial x^2}\right\} = (-ik_x)^2 \Psi_p(k_x, k_y; z_0), \qquad \mathcal{F}\left\{\frac{\partial^2 \psi_p}{\partial y^2}\right\} = (-ik_y)^2 \Psi_p(k_x, k_y; z_0). \tag{2.11}$$

Sostituendo i termini nella trasformata della 2.8, ottengo:

$$\frac{\partial^2 \Psi_p}{\partial z^2} + k_0^2 \left(1 - \frac{k_x^2}{k_0^2} - \frac{k_y^2}{k_0^2}\right) \Psi_p = 0. \tag{2.12}$$

Questa è un'equazione differenziale ordinaria del secondo ordine nella forma $\frac{\partial^2 f(z)}{\partial z} + cost^2 f(z) = 0$*, con cost=costante.*
La soluzione è data da $f(z_0) = f(0)e^{-i\,cost\,z_0}$*, con f(0) la condizione iniziale.*

Quindi la soluzione dell'equazione 2.12 è:

$$\Psi_p(k_x, k_y; z_0) = \Psi_{p0}(k_x, k_y) e^{\left[-ik_0\sqrt{1-\frac{k_x^2}{k_0^2}-\frac{k_y^2}{k_0^2}}\right]z_0}, \tag{2.13}$$

con $\Psi_{p0}(k_x, k_y) = \mathcal{F}\{\psi_{po}(x, y)\}$ e vado a definire la *funzione di trasferimento* $H(k_x, k_y; z_0)$ come:

$$\mathcal{H}(k_x, k_y; z_0) = \frac{\Psi_p(k_x, k_y; z_0)}{\Psi_{p0}(k_x, k_y)}. \tag{2.14}$$

Alla fine si arriva a calcolare $\psi_p(x, y; z_0)$ applicando la trasformata inversa a $\Psi(k_x, k_y; z_0)$:

$$\begin{aligned}\psi_p(x, y; z_0) &= \mathcal{F}^{-1}\{\Psi_p(k_x, k_y; z_0)\} = \mathcal{F}^{-1}\{\mathcal{F}\{\psi_{p0}(x, y)\}\mathcal{H}(k_x, k_y; z_0)\} \\ &= \mathcal{F}^{-1}\{\Psi_{p0}(k_x, k_y)\mathcal{H}(k_x, k_y; z_0)\} \\ &= \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \Psi_{p0}(k_x, k_y) e^{\left[-ik_0\sqrt{1-\frac{k_x^2}{k_0^2}-\frac{k_y^2}{k_0^2}}\right]z_0} e^{-i(k_x x + k_y y)} dk_x dk_y.\end{aligned} \tag{2.15}$$

L'equazione 2.15 è quello che stavo cercando, infatti determina la distribuzione del campo ad una distanza z_0, per una data distribuzione del campo a z=0.
La 2.15 viene chiamata **equazione con trasformata doppia** siccome per calcolare il risultato bisogna applicare la trasforma due volte, prima a ψ_{p0}, poi la trasformata inversa a $\Psi(k_x, k_y)\mathcal{H}(k_x, k_y; z_0)$.

Cercherò di descrivere il contenuto fisico dell'integrale 2.15, utilizzando la relazione:

$$\psi_{p0}(x, y) = \frac{1}{4\pi}\iint_{-\infty}^{\infty} \Psi_{p0}(k_x, k_y) e^{-i(k_x x + k_y y)} dk_x dk_y. \tag{2.16}$$

La 2.16 è la relazione fra l'ampiezza complessa $\psi_{p0}(x, y)$ ed il suo spettro $\Psi_{p0}(k_x, k_y)$.
Innanzitutto dobbiamo riconoscere che le frequenze radiali k_x, k_y nell'integrale sono le stesse dell'equazione d'onda originale k_{0x}, k_{0y}.
Osservando l'integrale 2.16, si nota che $\Psi_{p0}(k_x, k_y)e^{-i(k_x x + k_y y)}$ ha l'equazione di un'onda piana e sommandola lungo le componenti k_x, k_y si ottiene $\psi_{p0}(x, y)$ per z=0.
Per calcolare la distribuzione del campo ad una distanza z_0 lasciamo propagare le sue componenti, ciò equivale ad aggiungere uno spostamento alla fase dell'onda di $e^{ik_{0z}z_0} = e^{ik_z z_0}$, ottenendo:

$$\psi_p(x, y; z_0) = \frac{1}{4\pi}\iint_{-\infty}^{\infty} \Psi_{p0}(k_x, k_y) e^{-i(k_x x + k_y y + k_z z_0)} dk_x dk_y = \mathcal{F}^{-1}\left\{\Psi_{p0}(k_x, k_y) e^{-ik_z z_0}\right\}.$$

Osservando che:

$$|\vec{k_0}| = \sqrt{k_x^2 + k_y^2 + k_z^2} \quad \longrightarrow \quad k_z = k_{0z} = \pm|k_0|\sqrt{1 - \frac{k_x^2}{k_0^2} - \frac{k_y^2}{k_0^2}}.$$

Prendiamo l'equazione di k_z con il +, poiche sta a significare che l'onda si propaga lungo la direzione positiva dell'asse z. Infine sostituita nell'equazione precedente si riottiene la 2.15.

2.2.1 Diffrazione di Fresnel

La diffrazione di Fresnel è un'approssimazione della 2.15 che può essere applicata ad onde che si propagano con piccoli angoli rispetto all'asse ottico del sistema (nel nostro caso sarà sempre l'asse z), sotto la così detta *approssimazione parassiale*.
Nel concreto l'approssimazione si può utilizzare quando $k_x^2 + k_y^2 \ll k_0^2$ in modo da avere:

$$\sqrt{1 - \frac{k_x^2}{k_0^2} - \frac{k_y^2}{k_0^2}} \simeq 1 - \frac{k_x^2}{2k_0^2} - \frac{k_y^2}{2k_0^2}, \tag{2.17}$$

sostituisco nell'equazione 2.15:

$$\begin{aligned} \psi_p(x, y; z_0) &= \frac{1}{4\pi} \iint_{-\infty}^{\infty} \Psi_{p0}(k_x, k_y) e^{-i\left(k_0 - \frac{k_x^2}{2k_0} - \frac{k_y^2}{2k_0}\right) z_0} \, e^{-i(k_x x + k_y y)} dk_x dk_y \\ &= \mathcal{F}^{-1}\left\{\Psi_{p0}(k_x, k_y)\mathcal{H}_{Fe}\right\}. \end{aligned} \tag{2.18}$$

con $\mathcal{H}_{Fe}(k_x, k_y; z_0) = e^{-i\left(k_0 - \frac{k_x^2}{2k_0} - \frac{k_y^2}{2k_0}\right) z_0}$ *funzione di trasferimento nelle frequenze spaziali di Fresnel*.
La $\mathcal{H}_{Fe}$ è la approssimazione parassiale di di $\mathcal{H}$ e la sua trasformata inversa detta *risposta all'impulso spaziale di Fresnel* ha equazione:

$$h_{Fe}(x, y; z_0) = \mathcal{F}^{-1}\left\{\mathcal{H}_{Fe}(k_x, k_y; z_0)\right\} = \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2 + y^2)}. \tag{2.19}$$

Posso esprimere l'equazione 2.18 come integrale di convoluzione, utilizzando le proprietà delle trasformate di Fourier, ottenendo:

$$\begin{aligned} \psi(x, y; z_0) &= \psi_{p0}(x, y) * h_{Fe}(x, y; z_0) \\ &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} \iint_{-\infty}^{\infty} \psi_{p0}(x', y') e^{\frac{-ik_0}{2z_0}\left[(x - x')^2 + (y - y')^2\right]} dx' dy' \\ &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2 + y^2)} \iint_{-\infty}^{\infty} \psi_{p0}(x', y') e^{\frac{-ik_0}{2z_0}(x'^2 + y'^2)} e^{\frac{-ik_0}{z_0}(xx' + yy')} dx' dy' \\ &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2 + y^2)} \mathcal{F}\left\{\psi_{p0}(x', y') e^{\frac{-ik_0}{2z_0}(x'^2 + y'^2)}\right\}_{k_x = \frac{k_0 x}{z_0}, \, k_y = \frac{k_0 y}{z_0}}. \end{aligned} \tag{2.20}$$

Quando si utilizza l'integrale 2.20 per calcolare la diffrazione in approssimazione parassiale bisogna fare attenzione a riscalare gli assi alla fine, infatti avendo applicato solo la trasformata diretta ci troviamo nello spazio delle frequenze.
Senza modificare gli assi con $k_x = \frac{k_0 x}{z_0}, k_y = \frac{k_0 y}{z_0}$ non si arriva alla giusta soluzione.

I termini (x',y') sono quelli relativi al piano dell'oggetto quindi per z=0, mentre con (x,y) si intende il piano a distanza z_0 dove viene calcolata la diffrazione.
Infine la 2.20 viene chiamata **equazione con singola trasformata** siccome si va ad applicare solo la trasformata diretta per arrivare alla soluzione.

2.2.2 Diffrazione di Fraunhofer

L'equazione 2.20 può essere ulteriormente semplificata se la distanza tra l'oggetto ed il piano di registrazione è abbastanza grande, più precisamente deve rispettare la condizione:

$$\frac{k_0}{2}\left[x'^2 + y'^2\right]_{max} = \frac{\pi}{\lambda}\left[x'^2 + y'^2\right]_{max} \ll z_0. \tag{2.21}$$

Sotto questa condizione il termine $e^{\frac{-ik_0}{2z_0}(x'^2+y'^2)}$ è circa 1 e si può eliminare dall'equazione:

$$\begin{aligned}\psi_p(x,y;z_0) &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \iint_{-\infty}^{\infty} \psi_{p0}(x',y') e^{\frac{-ik_0}{z_0}(xx'+yy')} dx'dy' \\ &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \mathcal{F}\left\{\psi_{p0}(x',y')\right\}_{k_x=\frac{k_0 x}{z_0}, k_y=\frac{k_0 y}{z_0}}.\end{aligned} \tag{2.22}$$

La 2.22 viene chiamata la *equazione di diffrazione di Fraunhofer* ed è un caso limite della diffrazione di Fresnel.

2.2.3 Analisi della risposta all'impulso spaziale di Fresnel

In questa sezione illustrerò come si distribuisce l'onda generata dalla diffrazione di un singolo punto del'oggetto sul piano di registrazione posto ad una distanza z_0.
Il punto che genera l'onda può essere descritto dalla delta di Dirac, così definita:

$$\delta(x - x_0, y - y_0) = \delta(x_0, y_0) = \begin{cases} 1 & \text{se } x = x_0 \, e \, y = y_0, \\ 0 & \text{altrove.} \end{cases}$$

Utilizzando la diffrazione di Fresnel, l'onda oggetto ψ_{p0} generata da un singolo punto sul piano di registrazione, sarà data da:

$$\begin{aligned}\psi_{p0}(x,y;z_0) &= \delta(x_0,y_0) * h_{Fe}(x,y;z_0) = \delta(x_0,y_0) * \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \\ &= \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)}.\end{aligned} \tag{2.23}$$

Prenderò l'onda di riferimento piana con stessa fase iniziale dell'onda oggetto ed ipotizzo che abbia percorso la stessa distanza z_0, quindi $\psi_R = R_0 e^{-ik_0 z_0}$.
In questo modo la distribuzione dell'intensità delle due onde sovrapposte, quindi l'ologramma, è dato da:

$$\begin{aligned} H(x,y) = |\psi_{p0} + \psi_R|^2 &= \left| R_0 e^{-jk_0 z_0} + \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \right|^2 \\ &= R_0^2 + \left(\frac{k_0}{2\pi z_0}\right)^2 + R_0 \frac{-jk_0}{2\pi z_0} e^{\frac{jk_0}{2z_0}(x^2+y^2)} + R_0 \frac{jk_0}{2\pi z_0} e^{\frac{-jk_0}{2z_0}(x^2+y^2)}. \end{aligned} \tag{2.24}$$

Posso semplicare la 2.24 in una funzione reale, siccome ci interessa il campo elettrico e compattare l'equazione nel seguente modo:

$$|H(x,y)| = A + B \sin\left[\frac{jk_0}{2z_0}\left(x^2+y^2\right)\right]. \tag{2.25}$$

con $A = R_0^2 + (\frac{k_0}{2\pi z_0})^2$ e $B = \frac{R_0 k_0}{\pi z_0}$

L'equazione 2.25 viene chiamata *Fresnel Zone Plate (FZP)* e la andrò a disegnare nella figura seguente per due diverse distanze z_0, una il doppio dell'altra:

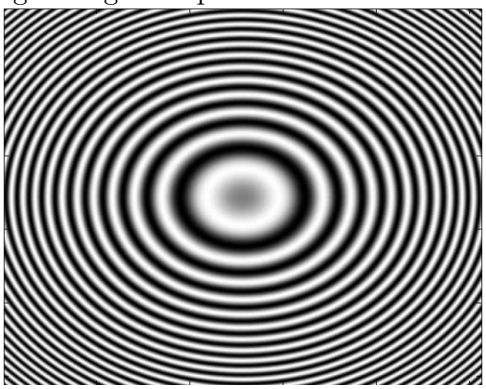

Figura 2.2.1
FZP con $z = z_0$

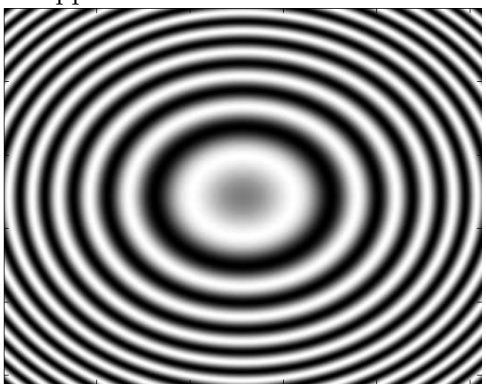

Figura 2.2.2
FZP con $z = 2z_0$

Grafici della FZP generati da un singolo punto sul piano oggetto di cui ne ho simulato la diffrazione per due distanze, uno il doppio dell'altra, mettendo in evidenza le variazioni delle frequenze spaziali.

La frequenza spaziale con cui la fase di FZP varia in funzione di x (stesso ragionamento si potrebbe fare con la y) è data da :

$$f_{local} = \frac{1}{2\pi}\frac{\partial}{\partial x}\left(\frac{k_0}{2z_0}x^2\right) = \frac{k_0 x}{2\pi z_0}. \tag{2.26}$$

La frequenza spaziale con cui si alternano massimi e minimi della FZP, come si può vedere anche dalle **Figure 2.2.1** e **2.2.2**, cresce linearmente al crescere della coordinata spaziale x (equivalentemente y); inoltre all'aumentare di z_0 si vede come le frangie diminuiscano di densità, quindi la FZP porta in sè l'informazione della z_0.

Vado ad analizzare il caso in cui ci siano due punti sorgenti, $\delta(x,y) + \delta(x - x_1, y - y_1)$, sempre posizionati a distanza z_0 dal piano di registrazione; l'onda oggetto ψ_{p0} diventa:

$$\psi_{p0}(x,y;z_0) = [b_0\delta(x,y) + b_1\delta(x - x_1, y - y_1)] * h(x,y;z_0). \tag{2.27}$$

con b_0 e b_1 le ampiezze dei due punti sorgenti.
L'ologramma, in questo caso, avrà equazione:

$$\begin{aligned} H(x,y) &= |\psi_{p0} + \psi_R|^2 = \\ &= \left| Re^{-ik_0z_0} + b_0e^{-ik_0z_0}\frac{ik_0}{2\pi z_0}e^{\frac{ik_0}{2z_0}(x^2+y^2)} + b_1e^{-ik_0z_0}\frac{ik_0}{2\pi z_0}e^{\frac{ik_0}{2z_0}[(x-x_1)^2+(y-y_1)^2)]}\right|^2, \end{aligned} \tag{2.28}$$

come fatto precedentemente porto l'equazione nella forma reale, ed ottengo:

$$\begin{aligned} H(x,y) = C + \frac{R_0b_0k_0}{\pi z_0}sin\left[\frac{k_0}{2_0}(x^2+y^2)\right] + \frac{R_0b_1k_0}{\pi z_0}sin\left\{\frac{k_0}{2_0}[(x-x_1)^2 + (y-y_1)^2]\right\} + \\ + 2b_0b_1\left(\frac{k_0}{2\pi z_0}\right)^2 cos\left\{\frac{k_0}{2z_0}[(x_1^2+y_1^2) + 2xx_1 + 2yy_1]\right\}. \end{aligned} \tag{2.29}$$

con C costante ottenuta similmente alla costante A della 2.15.

Il secondo ed il terzo termine sono la FZP dei due punti generatori, mentre l'ultimo termine è una grata cosinuisoidale dovuta all'interferenza tra le onde sferiche. La grata, data dal coseno, durante il processo di ricostruzione genera del rumore sul piano di ricostruzione [7].

Capitolo 3

Calcolo della diffrazione con le FFT

Nel capitolo precedente ho introdotto la teoria della diffrazione che, grazie agli integrali 2.15 e 2.20, mi permette di calcolare la distribuzione del campo elettrico ad una precisa distanza z_0 dall'ostacolo; adesso spiegherò come questi integrali possano essere svolti al computer ponendo le basi per la *Dital Holography* e la *Computer Generated Holography*.
Per descrivere un segnale luminoso al computer dovrò campionarlo, cioè registrare i valori dell'ampiezza per intervalli discreti e regolari dello spazio o del tempo; in questo modo ottengo un stringa di valori (nel mio caso discreta nello spazio) che approssima la funzione originale. Per un segnale periodico esiste il teorema del campionamento il quale assicura che se gli intervalli di campionamento presi sono abbastanza piccoli (in confronto con le frequenze dell'onda) posso descrivere la funzione originale continua del segnale senza perdere nessuna informazione.
Nella CGH quando simulerò la diffrazione della luce dovrò stare attento a definire degli intervalli adeguatamente piccoli in modo da rispettare le condizioni rese note dal teorema del campionamento.
Invece nella DH si digitalizza il segnale quando viene registrato l'ologramma con l'ausilio del CCD, che schematicamente posso disegnare come:

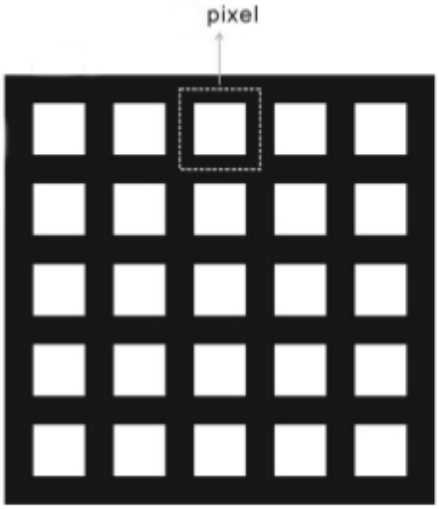

figura 3.1
struttura del CCD

Il CCD è composto da numerosi semiconduttori sensibili alla luce, chiamati pixel, disposti consecutivamente a formare una griglia duo dimensionale.
Ogni pixel è schermato dagli altri in modo che possano funzionare uno indipendentemente dall'altro (nella figura 3.1.1 la parte bianca è la zona attiva del pixel, mentre quella scura è la schermatura che li divide).

Quando il CCD viene investito da un'onda elettromagnetica ogni pixel trasferisce un quantità di elettroni proporzionale all'energia del fotone che lo ha colpito, quindi proporzionale all'intensità dell'onda elettromagnetica in quel punto.
Il teorema del campionamento in questo caso impone che la dimensione dei pixel dovrà essere abbastanza piccola, in modo da poter immagazzinare tutta l'informazione dell'ologramma.

3.1 Digitalizzazione del segnale

Un qualsiasi segnale campionato può essere descritto con (lo prendo in una dimensione per semplificare la trattazione):

$$f[n] = f(n\Delta_x) = \sum_{k=0}^{N-1} f(k\Delta_x)\delta([n-k]\Delta_x), \tag{3.1}$$

con N=numero di campionamenti, n numero intero compreso tra 0 ed N-1 e $\delta([n-k]\Delta_x)$ definita come:

$$\delta([n-k]\Delta_x) = \begin{cases} 1 & \text{se } n = k \\ 0 & \text{se } n \neq k \end{cases}$$

Inoltre è sottoinsteso che il segnale f(x) deve essere nullo al di fuori dell'intervallo $0 \leq x < N\Delta_x$, cosi facendo N, che è il numero di campionamenti presi, sarà un numero finito.

Per sapere se il la grandezza dell'intervallo di campionamento è sufficiente per immagazzinare tutta l'informazione del segnale, dovrò enunciare il *Teorema del campionamento di Nyquist-Shannon*, che afferma:

Dato un intervallo di campionamento Δ_x, quindi una frequenza di campionamento $f_s = \frac{1}{\Delta_x}$ è possibile ricostruire univocamente un segnale analogico f(x), con frequenza di banda f_b limitata, partendo dai suoi campionamenti $f(n\Delta_x)$ se solo se $f_s > 2f_b$.

Con la frequenza di banda f_b si intende la massima frequenza dell'onda in esame.

Il teorema può anche essere riformulato nella seguente maniera:
Dato un intervallo di campionamento Δ_x, la frequenza di campionamento f_s è fissata e l'onda che si vuole campionare dovrà avere una frequenza massima f_b t.c:

$$f_b < \frac{f_s}{2} = f_{NQ}. \tag{3.2}$$

con f_{QN} detta *frequenza di Nyquist.*

Lo spettro di una funzione campionata f[n] è diverso dallo spettro della funzione originale f(x), infatti applicando la trasformata di Fourier al segnale analogico continuo campionato si ottiene:

$$\mathcal{F}\left\{f(x)\sum_{n=-\infty}^{\infty}\delta(x-n\Delta_x)\right\} = 2\pi f_s \sum_{n=-\infty}^{\infty} F(k_x - 2\pi f_s n). \tag{3.3}$$

con $F(k_x) = \mathcal{F}\{f(x)\}$.

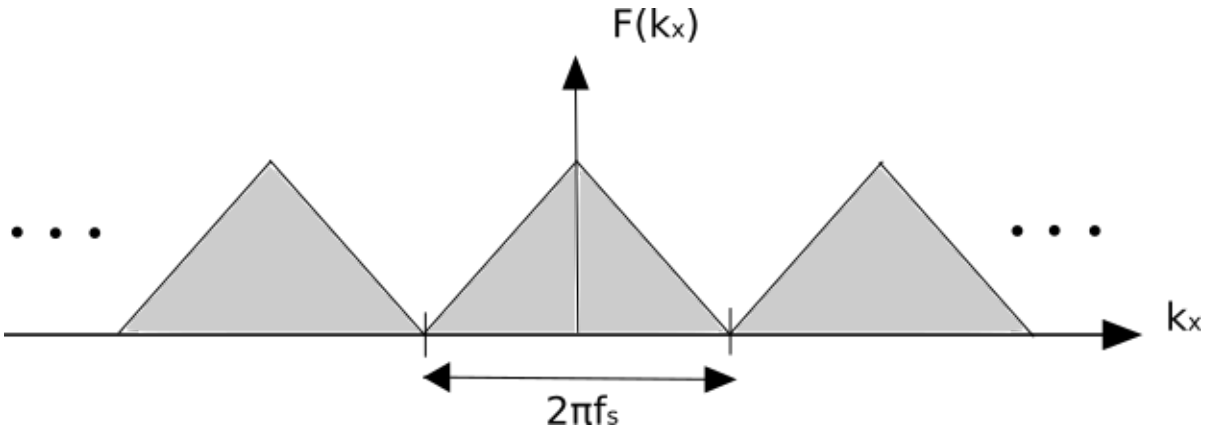

Figura 3.2

Spettro della funzione campionata, la figura mostra il grafico di un segnale campionato in cui $B < 2\pi f_s$, che quindi rispetta il teorema del campionamento

Lo spettro della funzione campionata si ripete ogni $2\pi f_s$ radianti/secondi, quindi possiamo prendere qualsiasi intervallo di lunghezza $2\pi f_s$ per ottenere tutta l'informazione del segnale.

Per rientrare nelle condizioni del teorema del campionamento, la frequenza di banda dell'onda in esame $B = 2\pi f_b$ dovrà essere minore della massima frequenza $2\pi f_s$; in caso contrario si avranno degli effetti di sovrapposizione che renderebbero impossibile un ricostruzione decente del segnale.

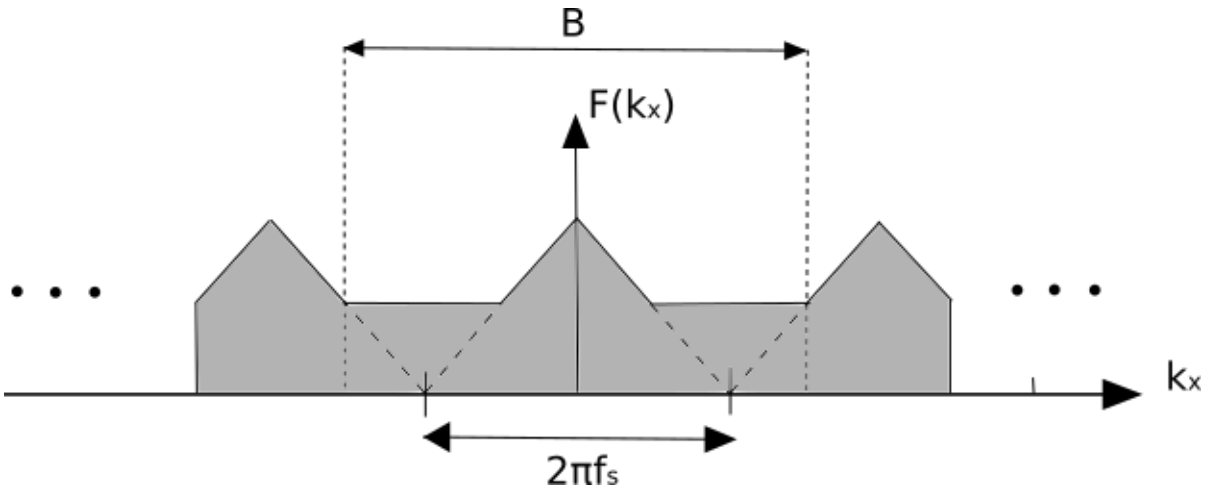

Figura 3.3

Spettro di una funzione campionata che non rispetta il teorema di Nyquist-Shannon, quindi $B > 2\pi f_s$, dovuto ad un sottocampionamento del segnale. La sovrapposisione fra gli spettri produce frequenze non proprie del segnale originale nel processo di conversione generando il fenomeno del così detto ***aliasing****.*

Durante il campionamento di un segnale in cui si discretizza spazialmente di Δ_x la funzione continua, implicitamente si discretizzano anche le frequenze del segnale; quindi avremo degli intervalli di grandezza Δf, detta *frequenza di risoluzione*, sui cui andremo a campionare lo spettro del segnale.
Per trovare la formula di Δf si osserva che il segnale f(x), che è campionato su di un dispositivo di lunghezza $L_x = N\Delta_x$, avrà uno spettro che può essere espresso come:

$$\mathcal{F}\left\{f(x)rect\left(\frac{x}{L_x}\right)\right\} = F(k_x) * sinc\left(\frac{k_x L_x}{2\pi}\right). \tag{3.4}$$

Come si può vedere dalla formula la grandezza dell'intervallo Δf, su cui è campionato lo spettro, è data dalla periodicità della funzione $sinc\left(\frac{k_x L_x}{2\pi}\right)$.
La 'larghezza' di Δk della funzione sinc può essere definita imponendo l'argomento uguale ad 1, in modo da annullare la funzione, quindi:

$$\Delta k = \frac{2\pi}{L_x} = \frac{2\pi}{N\Delta_x} \quad \longrightarrow \quad \Delta f = \frac{1}{N\Delta_x} = \frac{1}{L_x}. \tag{3.5}$$

Quando ci si riferisce al segnale analogico f(x) non si intende l'onda elettromagnetica che si va ad utilizzare, ma bensì l'immagine che si costruisce partendo dall'onda iniziale. Se si prendesse

come f(x) una qualsiasi onda nel visibile si avrebbero delle frequenze elevatissime che di sicuro non rispetterebbero il teorema del campionamento su di un qualsiasi ccd. La frequenza di banda di cui si parla può essere associata alla funzione 2.25, la quale ci dice quando velocemente varia la distrubuzione del campo diffratto, quindi della nostra immagine.

Ricapitolando, nel passaggio da analogico a digitale si avranno le seguenti condizioni:

- Una volta fissato l'intervallo di campionamento Δ_x, la massima frequenza che il segnale potrà avere sarà data dalla *frequenza di Nyquist* $f_{NQ} = \frac{f_s}{2}$.

- La risoluzione della frequenza di campionamento Δf è fissata una volta conosciuti il numero di campionamenti N, e la grandezza dell'intervallo di campionamento Δ_x dalla formula $\Delta f = \frac{1}{N\Delta_x}$.

3.2 Calcolo della diffrazione con le FFT

Innanzitutto si devono introdurre le trasformate di Fourier discrete che permettono di applicare la trasformata di Fourier ad un array di campionamenti di una funzione equidistanziati, con Δ_x l'intervallo di campionamento.
Le trasformate discrete DFT vengono così definite:
trasformata discreta di Fourier

$$F[m] = \sum_{n=0}^{N-1} f[n] e^{-i\frac{2\pi nm}{N}}, \tag{3.6}$$

con n e m numeri interi che vanno da 0 ad N-1 ed N è il numero di campionamenti presi.
Mentre la *trasformata discreta inversa di Fourier* è così definita:

$$f[n] = \frac{1}{N} \sum_{m=0}^{N-1} F[m] e^{i\frac{2\pi mn}{N}}. \tag{3.7}$$

Le traformate discrete hanno le stesse proprietà delle trasformate continue ma hanno dei tempi di elaborazione molto lunghi, quindi nei casi concreti non possono essere utilizzate.
Grazie all'algoritmo scoperto da Cooley e Tukey, diffuso su una pubblicazione nel 1965, si poterono iniziare ad elaborare le trasformate di Fourier discrete con un computer. Questo algoritmo restituisce lo stesso risultato delle traformate discrete riducendo il numero di passaggi aritmetici da un ordine $O(N^2)$ ad $O(N \log_2(N))$ [8].
Da qui in avanti andremo ad applicare questo algoritmo per calcolare le trasformate di Fourier che chiamerò *fft2* per la trasformata diretta e *ifft2* per trasformata inversa, mentre il due sta ad indicare che utilizzerò l'algoritmo in due dimensioni.

3.2.1 Metodo D-FFT

D-FFT, *Double fast Fourier trasform*, è il metodo che si utilizza al computer per andare a risolvere l'equazione 2.15, il quale ci permette di calcolare il campo diffratto $\psi_p(x, y; z_0)$ con l'utilizzo di due trasformate:

$$\psi_p(x, y; z_0) = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\psi_{p0}(x, y)\right\}\mathcal{H}(x, y; z_0)\right\}.$$

con ψ_{p0} l'ampiezza complessa ed $\mathcal{H}(k_x, k_y; z_0)$ la funzione di trasferimento.
Per portarla in digitale, ipotizzo di campionare la funzione con M intervalli di grandezza Δ_x lungo x, mentre in y si avranno N intervalli di grandezza Δ_y, ottendendo:

$$\psi_p[m, n] = \mathit{ifft2}\left\{\mathit{fft2}\left\{\psi_{p0}[m, n]\right\}\mathcal{H}[p.q]\right\}, \tag{3.8}$$

con $\mathcal{H}[p, q]$, la funzione di trasferimento discretizzata, data da:

$$\mathcal{H}[p.q] = e^{-ik_0 z_0\sqrt{1-\frac{(p\Delta k_x)^2}{k_0^2}-\frac{(q\Delta k_y)^2}{k_0^2}}}, \tag{3.9}$$

con Δk_x e Δk_y gli intervalli sulle frequenze corrispondenti agli intervalli di campionamento Δ_x e Δ_y di equazione:

$$\Delta k_x = \frac{2\pi}{M\Delta_x} \qquad \Delta k_y = \frac{2\pi}{N\Delta_y}. \tag{3.10}$$

Infine [m,n] e [p,q] sono rispettivamente gli indici degli intervalli campionati, i primi nello spazo cartesiano mentre i secondo nello spazio delle frequenze. I due set di indici hanno lo stesso range di valori che sono $-M/2 < m, p < M/2 - 1$ e $-N/2 < n, q < N/2 - 1$.
Gli indici si prendono negli intervalli appena detti e non più da zero siccome si è preso come origine del piano x,y il centro dell'oggetto.

Validità metodo D-FFT

Per validità del metodo si intende che il prodotto dello spettro dell'ampiezza complessa $\Psi_{p0}[p, q]$ con la funzione di trasferimento $\mathcal{H}[p.q]$ dovrà rispettare il teorema del campionamento.
Genericamente prendiamo due funzioni a(x) e b(x) a banda limitata k_a e k_b, rispettivamente. La frequenza massima della funzione risultante c(x)=a(x)b(x) sarà dato dal prodotto di convoluzione tra gli spettri delle due funzioni originali; quindi la frequenza di banda di c(x) è $k_c = k_a + k_b$.
Prese due funzione a banda limitata a(x) e b(x) campionate con a[n], b[n] tali che le loro frequenze di Nyquist siano $k_a < 2\pi f_{NQ,a},\ k_b < 2\pi f_{NQ,b}$, per la funzione c(x) dovrò rispettare la condizione $k_c < 2\pi f_{QN,c}$ per non avere effetti di aliasing.
Per arrivare all'equazione 3.8 si sono dovuti campionare i due termini $\psi_{p0}(x, y)$ e $\mathcal{H}(k_x, k_y; z_0)$ ed inoltre anche il prodotto $\Psi_{p0}(k_x, k_y)\mathcal{H}(k_x, k_y; z_0)$ sarà campionato nello spazio delle frequenze.
Per verificare la validità del metodo D-FFT ci si deve accertare che l'ultimo prodotto sia ben campionato senza effetti di sovrapposizione; per fare ciò vado ad analizzare le due componenti separatamente.

Per comprendere le proprietà di Ψ_{p0}, prendo ad esempio due segnali che contengono due soli impulsi; il primo $f_1(x)$ avrà gli i due picchi proprio sul limite della distanza di registrazione L_x mentre il secondo segnale conterrà i due pricchi più vicini, come in figura:

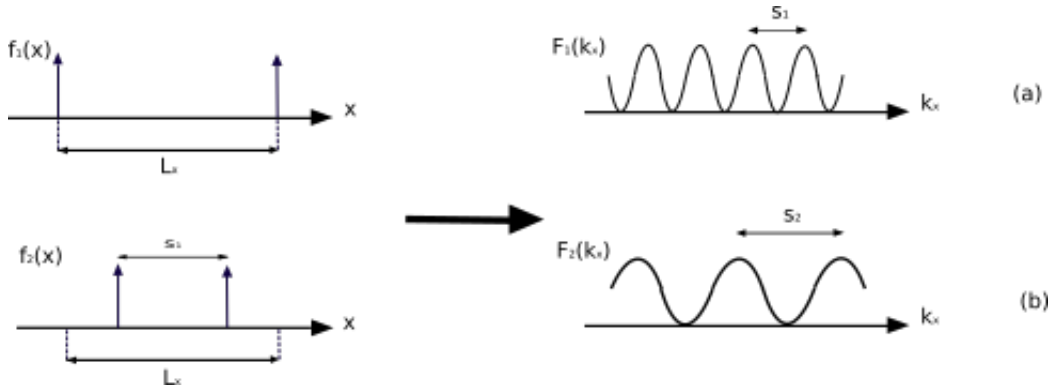

Figura 3.4

Grafico delle due funzioni, la prima con i picchi agli estremi del piano di registrazione mentre la seconda con i picchi più vicini. Come si può vedere dalla figura lo spettro della prima funzione ha una frequenza più alta rispetto al secondo spettro.

Lo spettro della funzione $f_1(x)$ è dato da:

$$F_1(k) = \mathcal{F}\left\{\delta(x + L_x/2) + \delta(x - L/2)\right\} = 2\cos\frac{k_x L_x}{2}. \tag{3.11}$$

Il periodo di $F_1(k)$ nello spazio delle frequenze è dato da:

$$s_1 = \frac{4\pi}{L_x} \quad \textit{radianti/lunghezza}, \tag{3.12}$$

mentre la frequenza, sempre nello spazio delle frequenze, di $F_1(k)$ è:

$$\frac{1}{s_1} = \frac{L_x}{4\pi} \quad \textit{lunghezza/radianti}. \tag{3.13}$$

Invece per l'equazione $f_2(x)$ la distanza tra i due picchi $d_x < L_x$, così anche la frequenza sarà minore rispetto a quella di $f_1(x)$, infatti $\frac{1}{s_2} = \frac{d_x}{4\pi} < \frac{L_x}{4\pi} = \frac{1}{s_1}$. Da questi due esempi si può vedere che in generale la frequenza più alta sarà data da $f_1(x)$.
Adesso vado a considerare il caso in cui $\Delta_x = \Delta_y$ ed M=N ed assumo che ψ_{p0} sia non nulla solo per un certo range di campionamenti M'*M' centrati nell'origine, con $0 < M' < M$.
In accordo con la 3.13 la massima frequenza di ψ_{p0} è data da $M'\Delta_x/4\pi$ nello spazio delle frequenze. mentre per $\mathcal{H}(k_x, k_y; z_0)$ similmente all'equazione 2.26 avrò, sempre nello spazio delle frequenze:

$$\frac{1}{2\pi}\frac{d}{dk_r}\left[-ik_0 z_0\sqrt{1 - \frac{k_r^2}{k_0^2}}\right] = \frac{zk_r}{k_0\sqrt{1 - k_r^2/k_0^2}} \qquad con \quad k_r = \sqrt{k_x^2 + k_y^2}. \tag{3.14}$$

Quindi la frequenza massima di $\mathcal{H}[q,p]$ lungo l'asse orizzontale o verticale è data da $k_r^{max} = \Delta_{kx}\, M/2$ con $\Delta_{kx} = \frac{4\pi}{M\Delta_x}$.
Osservando ancora l'equazione 3.5 la massima frequenza del prodotto tra lo spettro dell'ampiezza complessa e della funzione di trasferimento dovrà essere minore di $\frac{M\Delta_x}{4\pi}$. Sommando le due frequenze appena trovate si arriva alla condizione:

$$\frac{M'\Delta_x}{4\pi} + \frac{z_0 k_r^{max}}{2\pi\sqrt{k_0^2 - (k_r^{max})^2}} \leq \frac{M\Delta_x}{4\pi}, \tag{3.15}$$

da cui possiamo ricavare:

$$z_0 \leq \frac{\sqrt{4\Delta_x^2 - \lambda^2}}{2\lambda}(M - M')\Delta_x. \tag{3.16}$$

Da questa ultima equazione possiamo notare che il metodo D-FFT è preferibile per distanze via via minori, sennò per essere applicato potrebbe richiedere un numero di campionamenti estremamente elevato.

3.2.2 Metodo S-FFT

Riprendiamo l'equazione 2.20

$$\psi_p(x,y;z_0) = \frac{ik_0}{2\pi z_0} e^{-ik_0 z_0} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \mathcal{F}\left\{ \psi_{p0(x,y)} e^{\frac{-ik_0}{2z_0}(x^2+y^2)} \right\}_{k_x = \frac{k_0 x}{z_0},\, k_y = \frac{k_0 y}{z_0}}.$$

Voglio calcolarla nello spazio discretizzato con M intervalli di grandezza Δ_x lungo x, ed N intervalli di grandezza Δ_y lungo y. Sul piano di diffrazione il numero dei campionamenti rimane uguale ma la grandezza degli intervalli si modifica, come si può vedere dalla formula.
I nuovi intervalli di campionamento li chiamerò (Δ_x^d, Δ_y^d) e sono definiti come:

$$\Delta_x^d = \frac{\lambda z}{M\Delta_x} \qquad\qquad \Delta_y^d = \frac{\lambda z_0}{N\Delta_y}. \tag{3.17}$$

Quindi la 2.20 discretizzata risulta:

$$\psi_p[p,q] = \frac{ik_0 e^{-ik_0 z_0}}{2\pi z_0} Q_2[p,q]\, fft2\{\psi_{p0}[m,n] Q_1[m,n]\}, \tag{3.18}$$

con m,n,p,q sono indici con le stesse funzioni spiegate nella sezione precedente, mentre $Q_1[m,n]$ e $Q_2[p,q]$ sono:

$$Q1[m,n] = e^{\left[\frac{-ik_0}{2z_0}\left(m^2\Delta_x^2 + n^2\Delta_y^2\right)\right]} \tag{3.19}$$

$$Q2[p,q] = e^{-\frac{ik_0}{2z_0}\left[(p\Delta_x^d)^2 + (q\Delta_y^d)^2\right]} = e^{\left[-i\pi\lambda z_0\left(\left(\frac{p}{M\Delta_x}\right)^2 + \left(\frac{q}{N\Delta_y}\right)^2\right)\right]} \tag{3.20}$$

Validità del metodo con trasformata singola

Come nella discussione fatta per la validità del metodo D-FFT, quando campiono il prodotto fra $\psi_{p0}(x,y)$ ed $e^{\left[\frac{-ik_0}{2z_0}(x^2+y^2)\right]}$, per ottenere $\psi_{p0}[m,n]Q_1[m,n]$ dovrò soddisfare la seguente condizione per un dato Δ_x:

$$\textit{massima frequenza } \psi_{p0} + \textit{massima frequenza } e^{\left[\frac{-ik_0}{2z_0}(x^2+y^2)\right]} \leq \frac{1}{2\Delta_x}. \tag{3.21}$$

Possiamo assegnare metà del contributo alla banda finale ad ognuno dei due termini in modo arbitrario. Comunque bisogna porre attenzione su $Q_1[m,n]$ poichè la sua frequenza dipende da z, come visto nella equazione 3.5. La frequenza maggiore la troviamo a bordi della zona di registrazione, quindi per $(M/2)\Delta_x$ e questa deve essere minore di $\frac{1}{4\Delta_x}$, perciò:

$$\frac{(M/2)\Delta_x}{\lambda z_0} \leq \frac{1}{4\Delta_x}, \tag{3.22}$$

quindi

$$z_0 \geq \frac{2M\Delta_x^2}{\lambda}. \tag{3.23}$$

Al contrario del metodo D-FFT, il metodo S-FFT funziona sempre meglio per distanze maggiori [9].

3.3 Implementazione dei metodi D-FFT e S-FFT con Octave

Per svolgere i metodi a singola e doppia trasformata, che ci permettono di simulare il fenomeno dello diffrazione della luce, andrò ad utlizzare Octave che contiene gli algoritmi delle trasformate di Fourier veloci.
Negli esempi andrò a simulare la diffrazione di un laser che produce un'onda piana attraverso un **pinhole**, una piccola apertura di forma circolare al centro di uno schermo opaco; le piccole dimensioni del foro, paragonabili alla lunghezza d'onda dell'onda incidente, permetteranno di osservare i fenomeni della diffrazione.
Utilizzerò dei piccoli script che non sono inclusi in Octave per rendere più intuitivo il codice:

- *axis scaling*: ha la funzione di creare un array i cui valori corrispondono agli intervalli del campionamento; riceve due valori, il primo è il numero di intervalli ed il secondo è la grandezza di un singolo intervallo Δ.
  Lo script porrà il valore 0 al centro dell'array per poi assegnare multipli interi di Δ crescenti al crescere dell'ordine dell'array, e decrescenti dalla parte centrale all'inizio. *axis scaling* ha lo scopo di simulare la distanza discretizzata dal centro del CCD, quindi la distanza dall'asse ottico sul piano xy.

- *pinhole*: con questo script andrò a costruire il pinhole, il quale sarà una matrice di valori zero con al centro una circonferenza di valori 1, andando a simulare la funzione di trasmittanza della luce dell'oggeto in questione.
  pinhole riceve 3 argomenti; i primi due sono gli assi scalati mentre l'ultimo parametro è il raggio del pinhole. Disegnando il risultato dello script, in cui il nero sono gli zeri e gli uno sono in bianco, si avrà:

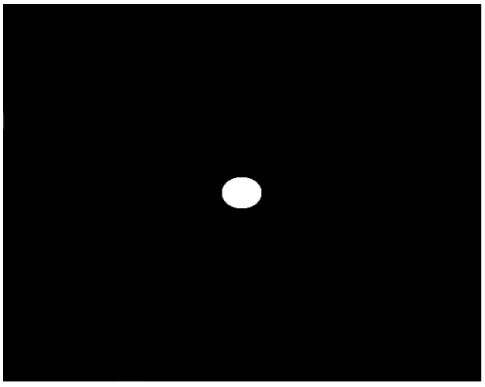

Figura 3.5
funzione di trasmittanza del pinhole, il foro è rappresentato dal cerchietto bianco al centro mentre la parte opaca che non fa passare la luce è la zona nera.

- *Normalization 2D* : serve per normalizzare ad 1 i valori delle matrici.

3.3.1 Implementazione metodo D-FFT

In questa sezione andrò ad implementare l'equazione 3.8, la quale calcola la distibuzione del campo $\psi_p(x, y; z)$ per distanze di propagazione piccole. Prima di illustrare il codice modificherò la 3.8 per renderne la scrittura al computer più semplice.

$$\psi_p[m,n] = \mathit{ifft2}\left\{\mathit{fft2}\left\{\psi_{p0}[m,n]\right\}\mathcal{H}[p.q]\right\}.$$

con $\mathcal{H}[p,q]e^{-ik_0 z_0\sqrt{1-\frac{(p\Delta k_x)^2}{k_0^2}-\frac{(q\Delta k_y)^2}{k_0^2}}}$, ed $\quad \Delta k_x = \frac{2\pi}{M\Delta_x} \quad \Delta k_y = \frac{2\pi}{N\Delta_y}$

Vado a modificare la $\mathcal{H}(p,q)$:

$$\begin{aligned}\mathcal{H}(x,y;z_0) &= e^{-ik_0 z_0\sqrt{1-\frac{(p\Delta k_x)^2}{k_0^2}-\frac{(q\Delta k_y)^2}{k_0^2}}} = e^{-i\frac{2\pi}{\lambda}z_0\sqrt{1-\frac{p^2\lambda^2}{4\pi^2}\frac{4\pi^2}{M^2\Delta_x^2}-\frac{q^2\lambda^2}{4\pi^2}\frac{4\pi^2}{N^2\Delta_y^2}}} \\ &= e^{i2\pi z_0\sqrt{\frac{1}{\lambda^2}-p^2\Delta f_x^2-q^2\Delta f_y^2}}.\end{aligned} \tag{3.24}$$

con $\Delta f_x = \frac{1}{M\Delta_x}$ e $\Delta f_y = \frac{1}{N\Delta_y}$.

Esempio diffrazione pinhole con il metodo D-FFT utilizzando Octave.

```
clear;
close all;
% Dati
delta_x=delta_y=4.65*10^-6; % intervallo di campionamento in metri
M=N=1024;                   % numero di campionamenti in x e y
l=0.65*10^-6;               % lunghezza d'onda in metri
k=(2*pi)/l;                 % modulo vettore d'onda
R=100*10^-6;                % raggio pinhole in metri
z=0.03;                     % distanza pinhole-ccd

% scalo gli assi
X=axis_scaling(M,delta_x);
Y=axis_scaling(N,delta_y);

% disegno il pinhole
U0=pinhole(X,Y,R);
figure(1),imagesc(X,Y,U0), colormap('gray');

% trasformata dell'ampiezza complessa
F_U0=fftshift(fft2(fftshift(U0)));

% calcolo Q1
delta_kx=1/(M*delta_x);
delta_ky=1/(N*delta_y);
Kx=axis_scaling(M,delta_kx);
Ky=axis_scaling(N,delta_ky);
for a=1:M;
 for b=1:N;
 Q1(b,a)=exp(-j*2*pi*z*((1/l)^2-(Kx(a)^2)-(Ky(b)^2))^0.5);
 endfor
endfor

% distribuzione del campo diffratto
Uf=fftshift(ifft2(fftshift(F_U0.*Q1)));
Uf=normalization_2D(Uf);

% intensità del campo
I=(abs(Uf)).^2;

% grafico della distribuzione dell'intensità
figure(2),imagesc(X,Y,I), colormap('gray'), axis('tight','xy');
```

```
% profilo dell'intensità per y=0
figure(3),plot(X,I(512,:));
```

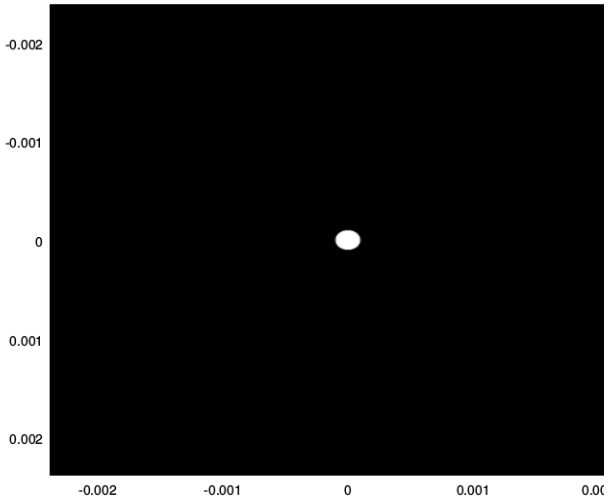

Figura 3.6.1

Distribuzione intensità D-FFT: in questo grafico non si vedeno i picchi secondari poichè sono molto più piccoli rispetto a quello centrale, quindi risultano quasi neri.

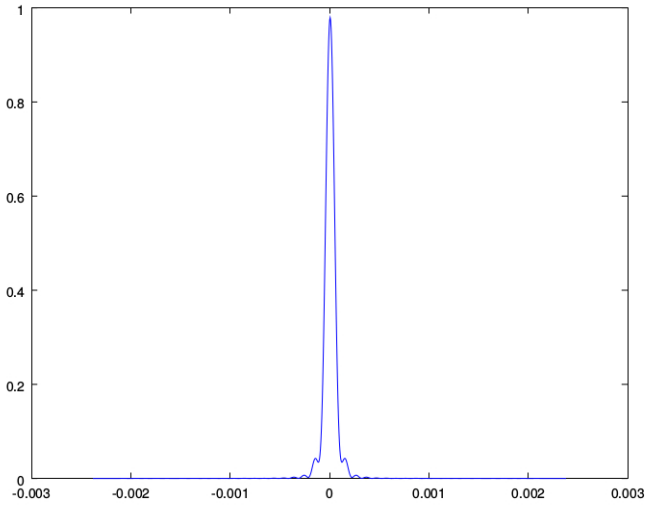

Figura 3.6.2

Profilo intensità D-FFT: plottando i valori del profilo al centro della figura metto in risalto i picchi ed i massimi secondari che verificano il fenomeno della diffrazione della luce.

Nell' immagine della distribuzione dell'intensità non si vedono i picchi secondari siccome sono molto più piccoli rispetto a quello centrale e risultano quasi neri. Nel grafico del profilo, invece, si riescono a mettere in evidenza.

3.3.2 Implementazione metodo S-FFT

Adesso andremo ad implementare l'equazione 3.18 per la distribuzione del campo diffratto a distanze maggiori:

$$\psi_p[p,q] = \frac{ik_0 e^{-ik_0 z_0}}{2\pi z} Q_2[p,q]\, fft2\{\psi_{p0}[m,n] Q_1[m,n]\}\,,$$

con

$$Q1[m,n] = e^{\left[\frac{-ik_0}{\lambda z_0}\left(m^2\Delta_x^2 + n^2\Delta_y^2\right)\right]},$$
$$Q2[p,q] = e^{\left[-i\pi\lambda z_0\left(\left(\frac{p}{M\Delta_x}\right)^2 + \left(\frac{q}{N\Delta_y}\right)^2\right)\right]},$$

ed infine:

$$\Delta_x^d = \frac{\lambda z_0}{M\Delta_x} \qquad \Delta_y^d = \frac{\lambda z_0}{N\Delta_y}. \tag{3.25}$$

Anche in questo caso andrò a modificare l'equazione in quanto quello che voglio calcolare è il valore assoluto del campo diffratto normalizzato ad 1, quindi si possono escudere dal calcolo sia $Q_2[p,q]$ (quando si applica il valore assoluto il suo contributo scompare), sia le costanti fuori dalla trasformata di Fourier. In questo modo avrò un'equazione del tipo:

$$|\psi_p[p,q]|_{normalizzata} = |\mathit{fft2}\{\psi_{p0}[m,n]Q_1[m,n]\}\,|_{normalizzata}. \tag{3.26}$$

Esempio diffrazione pinhole con il metodo S-FFT utilizzando Octave

```
% Metodo con una singola trasformata di Fourier
clear;
close all;

% Dati
delta_x=delta_y=4.65*10^-6; % intervallo di campionamento in metri
M=N=1024;                   % numero di campionamenti in x e y
l=0.65*10^-6;               % lunghezza d'onda in metri
k=(2*pi)/l;                 % modulo vettore d'onda
R=100*10^-6;                % raggio pinhole in metri
z=0.1;                      % distanza pinhole-ccd in metri

% scalo gli assi
X=axis_scaling(M,delta_x);
Y=axis_scaling(N,delta_y);

% disegno il pinhole
U0=pinhole(X,Y,R);
figure(1),imagesc(X,Y,U0), colormap('gray');

% calcolo Q1
cost=-(j*pi)/(l*z);
for a=1:M;
 for b=1:N;
 Q1(b,a)=exp(cost*(X(a)^2+Y(b)^2));
 endfor
endfor

% calcolo l'onda oggetto propagata di 0.04 metri
Uf=fftshift(fft2(fftshift(U0.*Q1)));

% normalizzo ad 1
Uf=normalization_2D(Uf);
```

```
% calcolo l'intensità
I=(abs(Uf)).^2;

% riscalo gli assi!
delta_xd=(l*z)/(M*delta_x);
delta_yd=(l*z)/(N*delta_y);
Xd=axis_scaling(M,delta_xd);
Yd=axis_scaling(N,delta_yd);

% grafico l'intensità dell'onda diffratta sugli assi riscalati
figure(2), imagesc(Xd,Yd,I), colormap('gray'), axis('tight','xy');

% prendo i valori centrali per mettere in evidenza la posizione ed
% i valori dei minimi e dei massimi dell'intensità, poi ne disegno
% il profilo per y=0
I1=I(412:612,412:612);
Xd1=Xd(412:612);
figure(3), plot(Xd1,I1(100,:));
```

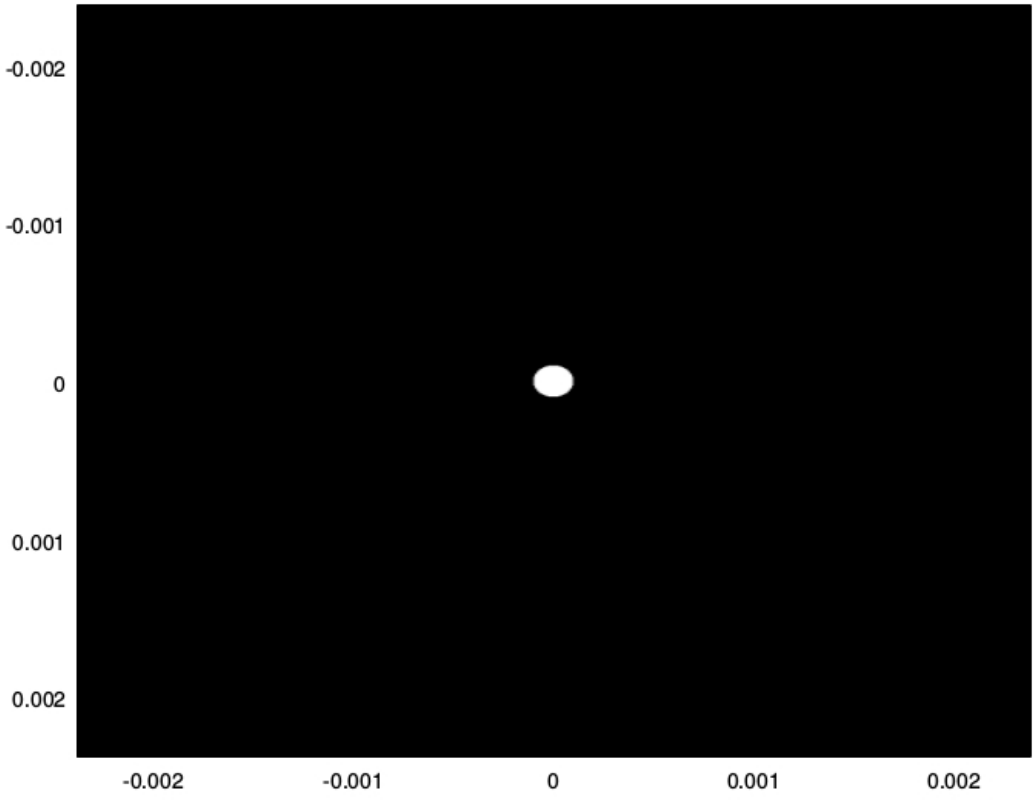

Figura 3.7.1

Distribuzione intensità S-FFT: come in figura 3.6.1 i picchi secondari, essendo molto più piccoli di quello centrale risultano neri.

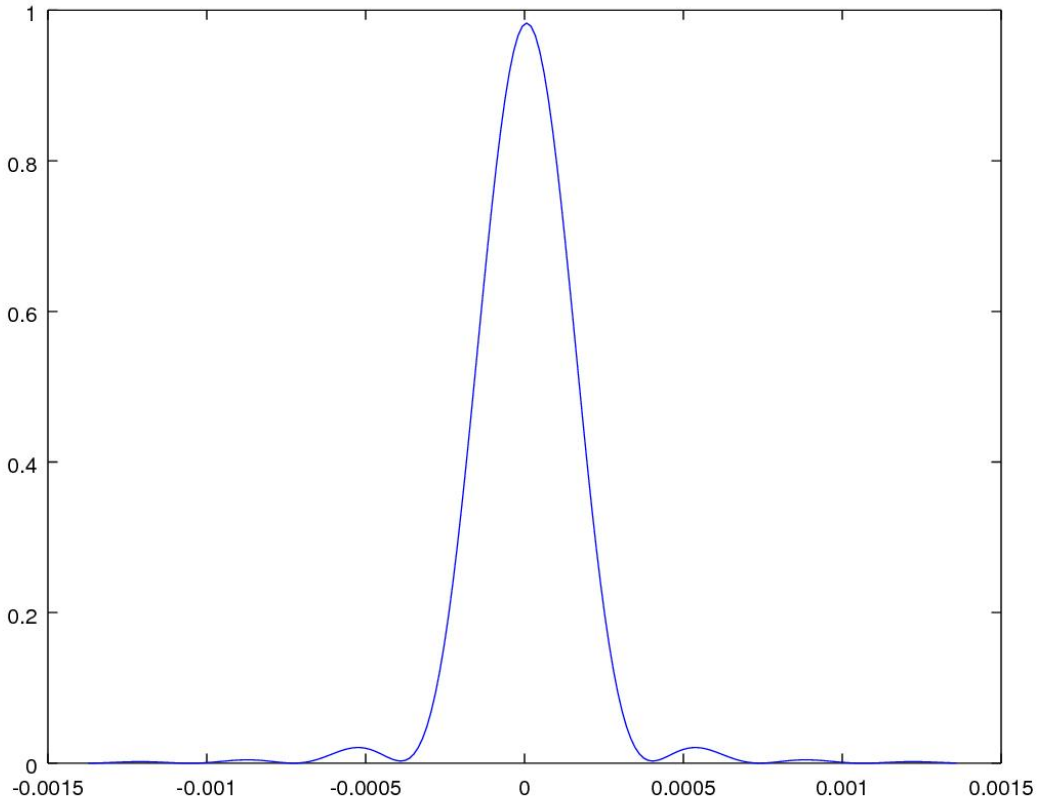

Figura 3.7.2
Profilo intensità S-FFT: anche in questo caso graficando il profilo dell'intensità si evidenziano i picchi secondari tipici del fenomeno di diffrazione.

Possiamo verificare se il grafico del profilo dell'intensità sià giusto in quanto esiste una semplice formula analitica che prevede la posizione dei massimi e dei minimi per la diffrazione di un pinhole.

Vado a riassumere le proprietà nella seguente figura:

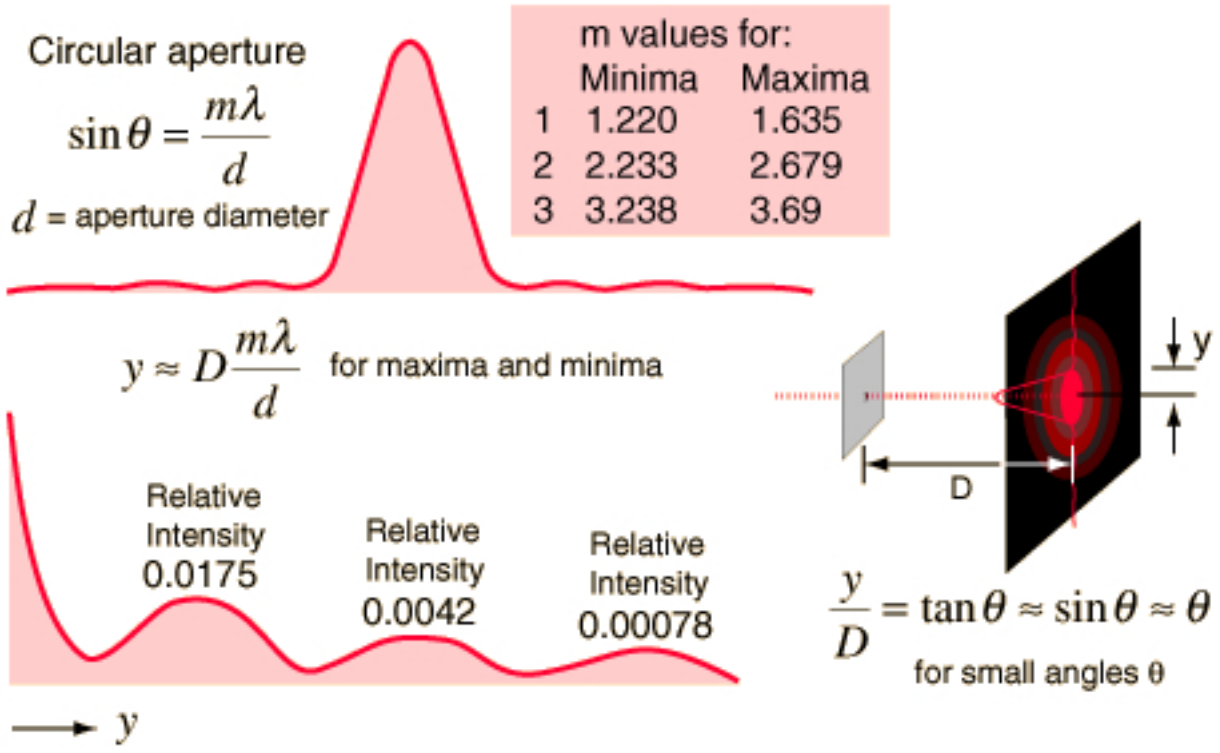

Figura 3.8

Questa immagine illustra in modo sintetico la formula analitica che utilizzerò per ricavare la posizione dei minimi e dei massimi di diffrazione, oltre che i valori delle intensità relative dei massimi secondari rispetto a quello centrale[10].

Analiticamente il caso preso in considerazione, quindi un'onda piana con $\lambda = 0,65\mu m$ su di un pinhole di diametro $d = 200\mu m$ che si propaga per un distanza D=z=0,1m, avrà come soluzioni:
posizione primo minimo $z * \frac{1,220*\lambda}{d} \approx 4,0 * 10^{-4}m$
posizione secondo minimo $z * \frac{2,233*\lambda}{d} \approx 7,3 * 10^{-4}m$
mentre nel grafico del profilo dell'intensità abbiamo i primi due minimi circa in $4,0 * 10^{-4}m$ ed $7,4 * 10^{-4}m$ molto simili a quelli analitici.
Inoltre anche la intensità relativa tra i massimi è circa la stessa, infatti prendendo i valori dal grafico 3.8, ottengo per il primo massimo un intensità relativa circa di 0,020 e per il secondo 0,0045.

Capitolo 4

Computer generated Holography di Fresnel fuori asse

In questo capitolo espliciterò il processo olografico per la ricostruzione digitale delle immagini virtuale e reale nell'apparato fuori asse simulando al computer l'intero processo, quindi andrò a fare un esempio di CGH. Generalmente per calcolare la diffrazione del campo si utilizza l'integrale di Fresnel, quindi il metodo S-FFT, poichè è sia l'algoritmo più veloce da calcolare sia è piu facile costruire degli apparati sperimentali che rispettino le condizioni imposte; infatti il metodo funziona per distanze z_0 più grandi.
Prima di iniziare la descrizione dei programmi per l'olografia fuori asse dovrò andare ad analizzare le condizioni sotto le quali si potranno ricostruire le immagini virtuale e reale e lo zero-order separati senza effetti di aliasing.
Andrò a generalizzare l'equazione 2.26, che descrive le frequenze locali generate dall'integrale di Fresnel di un singolo punto, ad un oggetto di dimensione finita; le frequenze più alte dovranno rispettare il teorema del campionamento.
Poi si dovrà analizzare per quale range di valori dell'inclinazione dell'onda di riferimento si riesca sia a dividere le immagini virtuale e reale e lo zero-order, sia a rispettare il teorema del campionamento; le frequenze spaziali aumentano quando ci si allontana dall'asse ottico, quindi aumentando l'inclinazione dell'onda di riferimento (questa proprietà si può sempre vedere dalla 2.26, poichè le frequenze aumentano con x ed y).

4.1 Condizione della risposta all'impulso spaziale di Fresnel generalizzata ad un oggetto di dimensione finita

Riprendendo l'equazione 2.26, che esprime la distribuzione delle frequenze spaziali locali generate da un singolo punto su di un piano a distanza z_0 se si utilizza l'integrale di Fresnel:

$$f_{local} = \frac{k_0 x}{2\pi z_0}.$$

Come si vede dall'equazione, le frequenze spaziali diventano più grandi man mano che si ingrandisce lo spostamento trasverale dal punto generatore, quindi per un oggetto di dimensione finita avrò la situazione in figura:

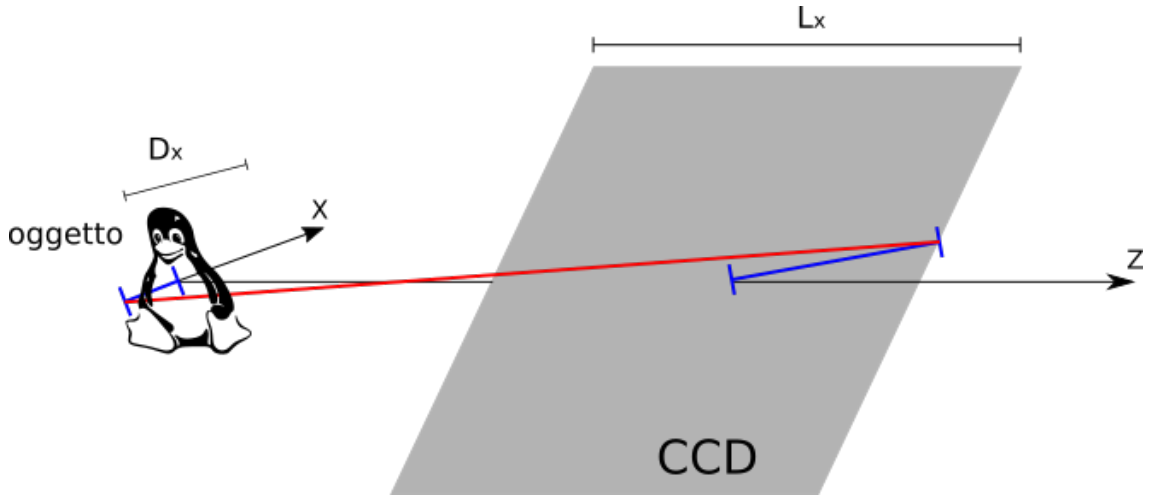

Figura 4.1

Le frequenze spaziali più alte generate dalla diffrazione della luce dall'ostacolo sul piano di registrazione dell'ologramma, sono quelle con un maggiore spostamento trasversale.

Chiamando D_x la massima estensione in x dell'oggetto e L_x la larghezza del CCD, il massimo spostamento trasversale si avrà per $(D_x/2 + L_x/2)$, quindi la 2.26 diventa:

$$\frac{D_x + M\Delta_x}{2\lambda z_0} \leq \frac{1}{2\Delta_x}. \tag{4.1}$$

Nell'equazione ho sostituito $L_x = M\Delta_x$, con M il numero dei pixel.
La condizione si può anche esprimere come:

$$z \geq \frac{\Delta_x}{\lambda}(D_x + M\Delta_x). \tag{4.2}$$

Infine aggiungendo l'effetto dell'onda di riferimento sulle frequenze spaziali, si ottiene:

$$z \geq \frac{\Delta_x(D_x + M\Delta_x)}{\lambda - 2sen\theta\,\Delta_x}. \tag{4.3}$$

I ragionamenti lungo y sono gli stessi; la condizione ci assicura che non ci siamo effetti di aliasing durante la ricostruzione dell'immagine dovuta alla distanza fra oggetto e CCD, ma non garantisce che le immagini virtuale e reale e lo zero-order siano divise alla fine.

4.2 Condizione sull'inclinazione dell'onda di riferimento

Riprendendo quanto scritto nel primo capitolo dovrò analizzare quale sarà l'angolo minimo per cui le immagini virtuale e reale e lo zero-order si dividano; inoltre si avrà una condizione sull'angolo massimo, infatti per quanto visto nel paragrafo 2.2.3 le frequenze spaziali risultanti dall'integrazione di Fresnel aumentanto all'aumentare di x e y (non potrò inclinare troppo l'onda di riferimento per rientrare nelle condizioni del teorema del campionamento).
Richiamando i termini utilizzati precedentemente avrò l'onda oggetto O(x,y), l'onda di riferimento $R(x,y) = R_0 e^{-ik_0 sen\theta(x+y)}$, ed infine l'onda di ricostruzione l'ha prenderò identica all'onda di riferimento, solo senza inclinazione come in figura 1.2, quindi $R_c(x,y) = R_0$. Sotto queste condizioni la formula analitica del campo finale, in cui ho sia l'immagine virtuale sia reale e lo zero-order sarà data da (ripresa dell'equazione 1.3):

$$\begin{aligned}
\psi_{finale} &= R_c(x,y)H(x,y) = R_c(x,y)\,|R(x,y)+O(x,y)|^2 = \\
&= R_c(x,y)\left[\,|R(x,y)|^2 + |O(x,y)|^2 + R_c(x,y)O(x,y)^* + R_c(x,y)^*O(x,y)\right] = \\
&= R_0\left[R_0^2 + |O(x,y)|^2 + R_0 O(x,y)^* e^{-ik_0 sen\theta(x+y)} + R_0 O(x,y) e^{ik_0 sen\theta(x+y)}\right] = \\
&= R_0^3 + R_0|O(x,y)|^2 + R_0^2 O(x,y)^* e^{-ik_0 sen\theta(x+y)} + R_0^2 O(x,y) e^{ik_0 sen\theta(x+y)} = \\
&= \psi^{(1)}_{finale} + \psi^{(2)}_{finale} + \psi^{(3)}_{finale} + \psi^{(4)}_{finale}.
\end{aligned} \tag{4.4}$$

I primi due termine sono lo zero-order e continuando a propagarsi senza deviazioni, il terzo è l'immagine virtuale mentre il quarto è l'immagine reale che rispettivamente ottengono un'inclinazione di $-sen\theta$ e $+sen\theta$.

4.2.1 Inclinazione minima per la separazione delle immagini

Per analizzare l'angolo minimo θ_{min}, per cui avremo la separazione dello zero-order e delle immagini, è più comodo studiare lo spettro dell'equazione d'onda finale termine per termine:

$$\begin{aligned}
&\Psi^{(1)}_{finale}(k_x,k_y) = \mathcal{F}\left\{R_0^3\right\} \\
&\Psi^{(2)}_{finale}(k_x,k_y) = \mathcal{F}\left\{R_0|O(x,y)|^2\right\} \\
&\Psi^{(3)}_{finale}(k_x,k_y) = \mathcal{F}\left\{R_0^2 O(x,y)^* e^{-ik_0 sen\theta\, x}\right\} = \mathcal{F}\left\{R_0^2 O(x,y)^*\right\} * \delta(k_x + k_0 sen\theta, ky) \\
&\Psi^{(4)}_{finale}(k_x,k_y) = \mathcal{F}\left\{R_0 O(x,y) e^{ik_0 sen\theta(x+y)}\right\} = \mathcal{F}\left\{R_0^2 O(x,y)\right\} * \delta(k_x - k_0 sen\theta, ky).
\end{aligned} \tag{4.5}$$

Nell'equazione precedente ho usato la solita notazione $\Psi(k_x, k_y) = \mathcal{F}\{\psi(x, y)\}$ ed ho espresso l'inclinazione dell'angolo solo sull'asse x per semplificare la trattazione, siccome i ragionamenti sull'asse y sono gli stessi.

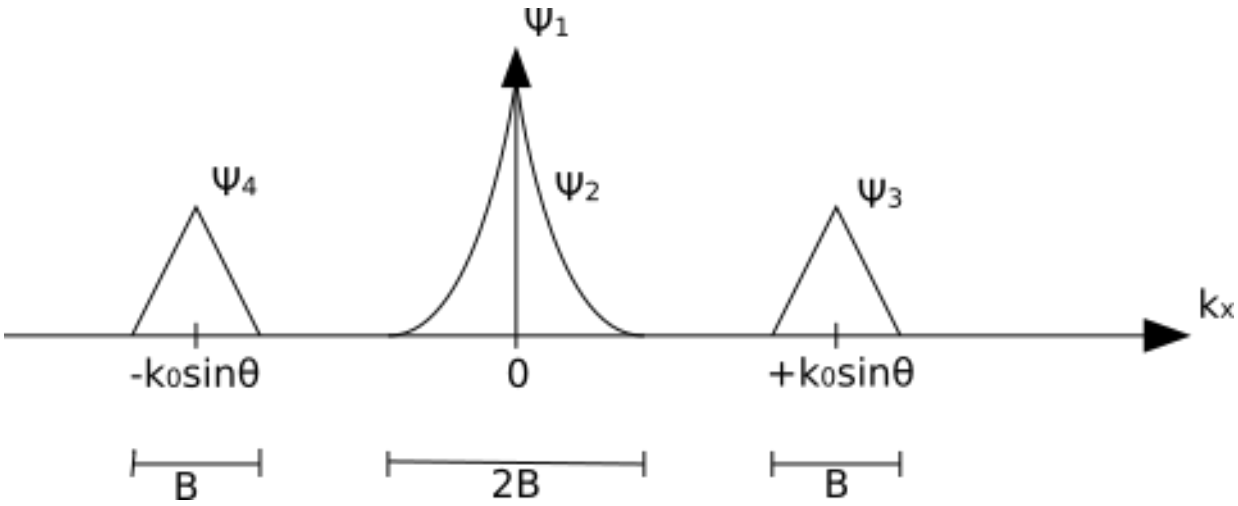

Figura 4.2

Spettro $\Psi_{finale}(k_x, k_y)$ dell'onda finale da cui possiamo osservare come lo zero-order e le due immagini virtuale e reale vengano divise spazialmente, durante la ricostruzione, dall'inclinazione data all'onda di riferimento.

Lo spettro $\Psi^{(1)}_{finale}$ è dato dalla trasformata di R_0, che è costante, quindi risulterà una pulsazione in $k_x = 0$.
$\Psi^{(3)}_{finale}$ e $\Psi^{(4)}_{finale}$, le immagini virtuale e reale, a cui ho dato una larghezza di banda di B, hanno lo spettro centrato rispettivamente in $+k_0 sen\theta$ ed $-k_0 sen\theta$.
Infine $\Psi^{(2)}_{finale}$ è proporzionale alla auto-correlazione di $\mathcal{F}\{O(x, y)\}$ e avrà una larghezza di banda di 2B centrata in $k_x = 0$.
Le immagini e lo zero-order saranno divise se i loro reciproci spettri non si sovrappongono l'uno con gli altri; quindi la distanza che li separa $k_0 sen\theta$ dovrà rispettare la seguente condizione:

$$k_0 sen\theta \geq \frac{3}{2} B. \tag{4.6}$$

La minima inclinazione dell'onda di riferimento per cui le immagini e lo zero-order si dividano sarà data da [11]:

$$\theta_{min} = sen^{-1}\left(\frac{3B}{2k_0}\right). \tag{4.7}$$

4.2.2 Inclinazione massima per rientrare nel teorema del campionamento

Come visto in precedenza utilizzando l'integrale di Fresnel per calcolare la diffrazione si ha che le frequenze spaziali (date dall'equazione 2.26), aumentano all'aumentare delle x ed y. La frequenza massima, includendo anche l'effetto dell'inclinazione portata dall'onda di riferimento, dovrà sottostare al teorema del campionamento per non avere aliasing.

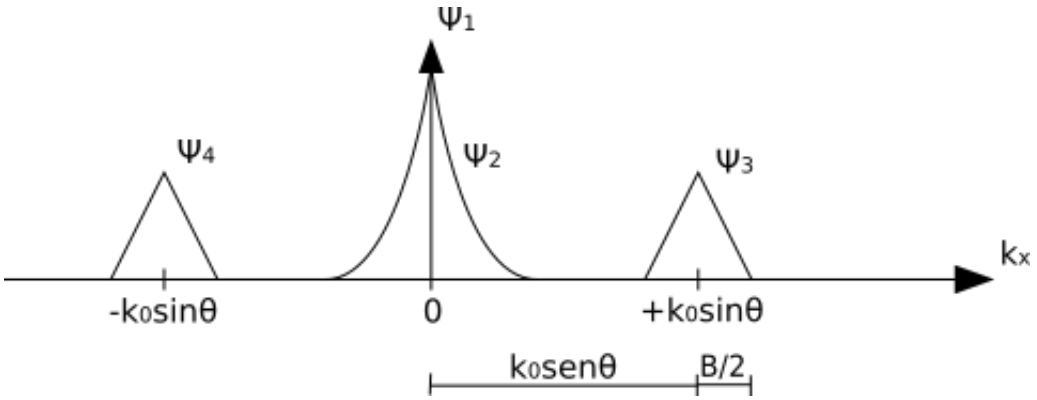

Figura 4.3

Riprendendo lo spettro dell'onda finale posso osservare quali sono le frequenze spaziali più alte, e queste dovranno rispettare il teorema del campionamento.

Anche in questo caso, osservando l'ultimo grafico, vedo che la frequenza massima è data da $k_0 sen\theta + \frac{B}{2}$ che dovrà rispettare il teorema del campionamento, quindi:

$$k_0 sen\theta + \frac{B}{2} \leq \frac{2\pi}{2\Delta_x}. \tag{4.8}$$

con Δ_x si intende sempre l'intervallo di campionamento.

4.2.3 Condizioni ottimali per l'olografia fuori asse di Fresnel

Per quanto detto nel paragrafo precedente si avrà un range di valori dell'inclinazione dell'onda di riferimento per cui le immagini saranno separate senza effetti di aliasing, dato da:

$$\frac{3}{2}B \leq k_0 sen\theta \leq \frac{2\pi}{2\Delta_x} - \frac{B}{2}.$$

Esplicitando $k_0 = \frac{2\pi}{\lambda}$ e mettendo in evidenza $sen\theta$ ottengo:

$$\frac{3}{2}\frac{\lambda}{2\pi}B \leq sen\theta \leq \frac{\lambda}{2\pi}\frac{\pi}{\Delta_x} - \frac{1}{2}\frac{\lambda}{\pi}B. \tag{4.9}$$

Similmente a quanto detto nel paragrafo 4.1, la frequenza massima B può essere espressa, prendendo in considerazione anche l'onda di riferimento, come:

$$B = 2\pi\frac{(D_x + M\Delta_x)}{\lambda z_0}. \tag{4.10}$$

Sostituendo la formula di B nella 4.9 ottengo:

$$\frac{3}{2}\frac{\lambda}{2\pi}\frac{2\pi(D_x+M\Delta_x)}{\lambda z_0} \leq sen\theta \leq \frac{\lambda}{2\Delta_x} - \frac{1}{2}\frac{\lambda}{2\pi}\frac{2\pi(D+M\Delta_x)}{\lambda z_0},$$

quindi:

$$\frac{3}{2}\frac{(D_x+M\Delta_x)}{z_0} \leq sen\theta \leq \frac{\lambda}{2\Delta_x} - \frac{1}{2}\frac{(D_x+M\Delta_x)}{z_0}. \tag{4.11}$$

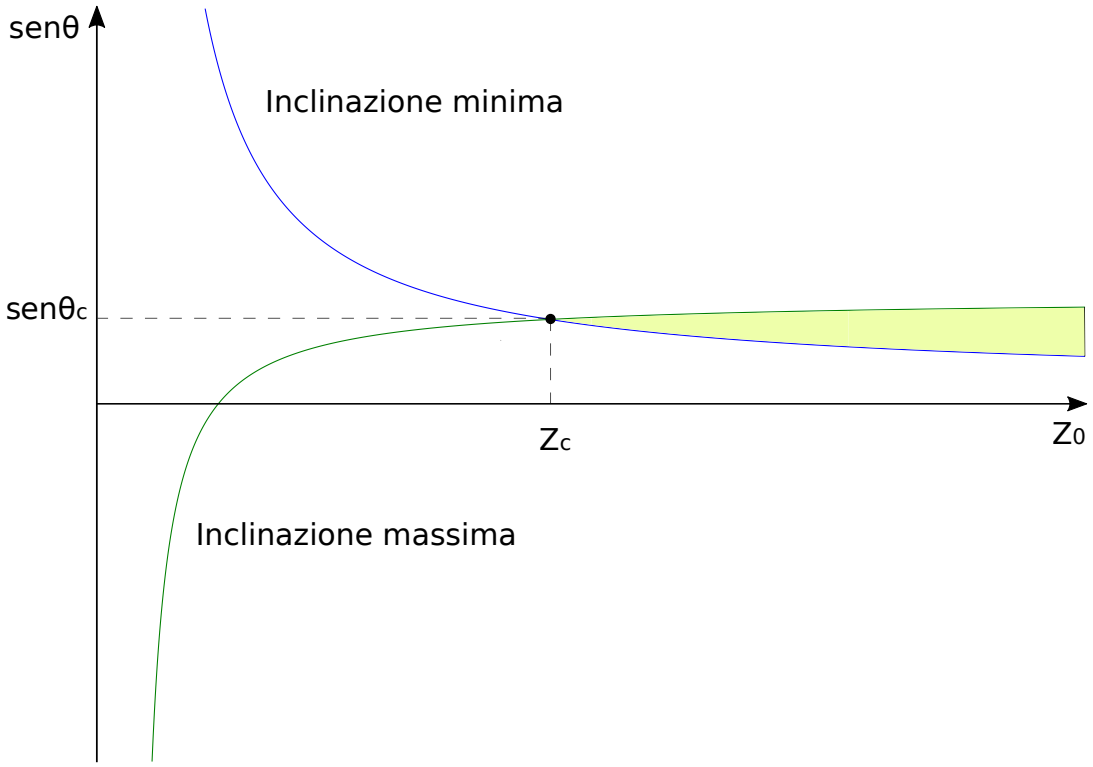

Figura 4.4

Range delle inclinazioni: la zona chiara fra i due grafici sono l'insieme dei valori con cui è possibile applicare l'olografia fuori asse, sia riuscendo a dividere le immagini virtuale, reale e lo zero-order sia a rispettare il teorema del campionamento.

Ottengo la condizione migliore per z_c (la distanza minima) poichè, riguardando le equazioni 3.17, all'aumentare di z_0 si ingrandisce il nuovo intervallo di campionamento, diminuendo la risoluzione finale.
Ricavo z_c uguagliando i due menbri della disequazioni, in modo da trovare il punto di intersezione fra i due grafici:

$$\frac{3}{2}\frac{D_x+M\Delta_x}{z_0} = \frac{\lambda}{2\Delta_x} - \frac{1}{2}\frac{D_x+M\Delta_x}{z_0}, \tag{4.12}$$

quindi:

$$z_c = \frac{4\Delta_x(D_x+M\Delta_x)}{\lambda}. \tag{4.13}$$

Infine sostituendo z_c in una delle due relazioni della 4.10 posso ricavare anche l'angolo ottimale θ_c:

$$\theta_c = sen^{-1}\left(\frac{3\lambda}{8\Delta_x}\right). \tag{4.14}$$

Le ultime due equazioni di θ_c e z_c sono le condizioni ottimali per registrare un ologramma fuori asse.

4.3 Simulazione con Octave dell'olografia fuori asse di Fresnel

In questo capitolo andrò ad illustrare il programma scritto con Octave per simulare i fenomeni fin qui descritti dell'olografia fuori asse, utilizzando la diffrazione di Fresnel.
Cercherò di simulare il più possibile i fenomeni che avverrebero se si registrasse l'ologramma attraverso il CCD, quindi darò in input i valori del CCD e del laser utilizzati in varie esperienze in laboratorio. Come CCD avevamo la camera Basler scA1390-17gc, la quale è costituita da 1390 pixel orizzontali e 1038 verticale ognuno di dimensione $4,65\mu m \cdot 4,65\mu m$, mentre come laser abbiamo usato un diodo-laser che produceva un'onda monocromatica di $0,65\mu m$.
Il programma sarà diviso in parti sia per poter illustrare meglio i passaggi, sia per dividere l'elaborazione (lanciare il programma in unico blocco potrebbe richiedere molto tempo). In ogni parte del programma utilizzerò *dlmwrite,dlmread*, funzioni base di Octave, per andare a leggere/scrivere i risultati delle elaborazioni delle altre parti del programma.
I passaggi fondamentali saranno:

- parte 1: *Calcolo dell'ampiezza complessa.*

- parte 2.0: *Calcolo della funzione di trasferimento dell'integrale di Fresnel a distanza* $+z_0$.
  parte 2.1: *Simulazione della diffrazione di Fresnel dell'ampiezza complessa utilizzando la funzione di trasferimento calcolata nella 2.0.*

- parte 3: *Calcolo dell'onda di riferimento introducendo l'inclinazione per l'olografia fuori asse.*

- parte 4: *Calcolo dell'ologramma*

- parte 5: *Ricostruzione dell'immagine reale simulando la diffrazione a distanza* $+z_0$*, quindi riutlizzerò la funzione di trasferimento ottenuta nella parte 2.0 .*

- parte 6.0: *Calcolo della funzione di trasferimento dell'integrale di Fresnel a distanza* $-z_0$.
  parte 6.1: *Ricostruzione dell'immagine reale simulando la diffrazione a distanza* $-z_0$ *andando ad utlizzare il risultato della parte 6.0 .*

Andrò a simulare la diffrazione prodotta dalla seguente immagine bidimensionale, composta da $256 \cdot 256$ pixel:

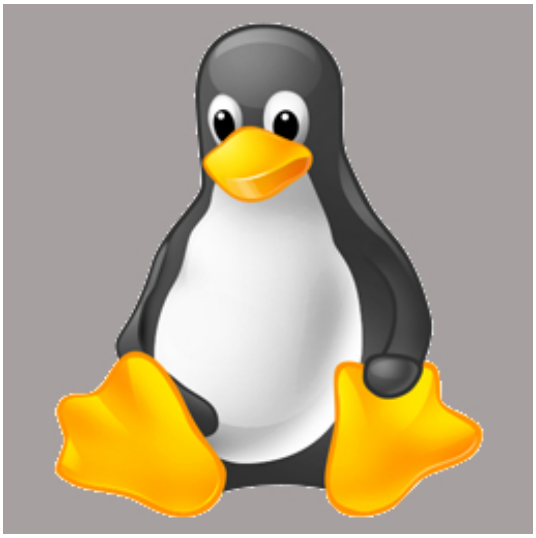

Figura 4.5
Immagine iniziale composta da $256 \cdot 256$ *pixel con cui andrò a fare un esempio di CGH.*

Siccome assegnerò ad ogni pixel una grandezza di $4,65\mu m$, l'immagine in figura 4.5 avrà una lunghezza (o larghezza) di $256 \cdot 4,65 \cdot 10^{-6} m = 1,1904 \cdot 10^{-3} m$; in pratica sto andando a simulare il processo olografico fuori asse per un oggetto delle dimensione del quadrato qui di fianco $\longrightarrow$.

4.3.1 Parte 1: ampiezza complessa

Innanzitutto si deve caricare l'immagine su cui si vuole lavorare in Octave utilizzando la funzione *imread*; per questo programma utilizzerò l'immagine in figura 4.5 che chiamo 'pinguino.jpg'. Successivamente si manipola l'immagine caricata utilizzando *mat2gray* (una funzione che si può trovare nel pacchetto opzionale 'octave-image'), che trasforma valori interi di tipo uint8 (vanno da 0 a 255) in valori di tipo double compresi fra 0 ed 1. Consecutivamente si applica *rgb2gray*, funzione che possiamo trovare ancora in 'octave-image'; *rgb2gray*, infine, converte la colorazione dell'immagine da RGB ad una scala di grigi.
Il modello RGB per ottenere i colori usa delle combinazioni di rosso, blu e verde dandogli dei valori compresi tra 0 e 255, quindi per rappresentare un immagine di $M \cdot N$ pixel si dovrà utilizzare una matrice $M \cdot N \cdot 3$. Utilizzando rgb2gray trasformiamo i colori in scala di grigi, quindi per rappresentare la stessa immagine si utilizzerà una matrice $M \cdot N$ con dei valori sempre compresi tra 0 e 255.
Dopo aver convertito l'immagine in scala di grigi andrò a normalizzare ad 1 i valori della matrice in modo da simulare la funzione di trasmittanza della luce, ottenendo:

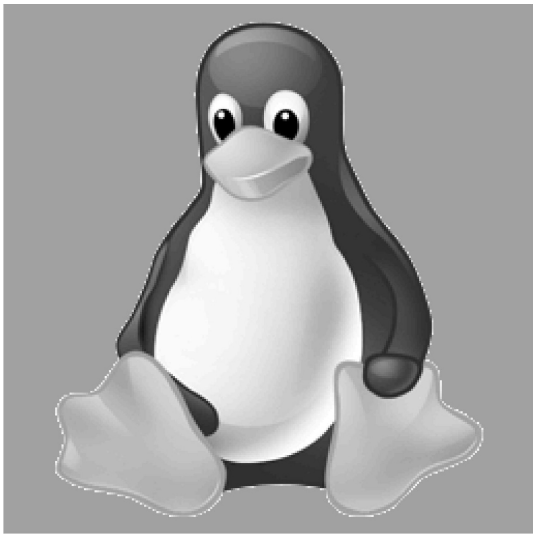

Figura 4.6
Ampiezza complessa: simula la funzione di trasmittanza della luce che attraversa un schermo bidimensionale sui cui è stampata l'immagine in figura 4.5

```
% ampiezza complessa

clear;
close all;

% apro l'immagine 'pinguino.jpg' e la porto in scala di grigi con valori
% compresi tra 0 e 1
U0=imread('pinguino.jpg');
U0=mat2gray(U0);
U0=rgb2gray(U0);
U0=normalization_2D(U0);

% dati dell' esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta_x=delta_y=delta=4.65*10^-6;% dimensione dei pixel
M=1390;                          % numero massimo di pixel del CCD
D=256*4.65*10^-6;                % dimensione pinguino
z=(4*delta*(D+1390*delta))/l;    % distanza pinhole-ccd in metri
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding

%-------------------------------------------------------------------------%
% zero padding, vado ad aggiungere zeri intorno all'immagine fino a
```

```
% farla diventare una matrice Mi*Mi, il significato di questo passaggio
% lo spiego appena finita questa parte del programma.
U0_zero_padding=zero_padding(U0,Mi);
%------------------------------------------------------------------------%

dlmwrite('q1_ampiezza_complessa.txt',U0_zero_padding);
```

Siccome applicherò spesso il zero-padding ho deciso di scrivere un piccolo script che automatizza il procedimento, il cui codice è scritto nell'appendice. La funzione zero_padding riceve come primo argomento un matrice di dimensioni generiche mentre nel secondo riceve un intero pari; lo script estende la dimensione della matrice ricevuta in input fino all'intero ricevuto nel secondo argomento, aggiungendo zeri.
Per spiegarne il significato bisogna osservare le relazioni 3.17, le quali descrivono come il metodo S-FFT vada a modificare la dimensione degli intervalli di campionamento.
Siccome lo scopo del programma è anche quello di simulare la cattura dell'immagine da parte del CCD, effettuando uno zero padding, voglio conservare la grandezza dell'intervallo di campionamento di $4,65\mu m$, che è proprio la grandezza del pixel.
Per ottenere M_i manipolo la relazione 3.17, impongo l'uguaglianza fra Δ_x^d e Δ_x facendo variare M.

$$\Delta_x^2 = \frac{\lambda z_0}{M_i} \quad \longrightarrow \quad M_i = \frac{\lambda z_0}{\Delta_x^2}. \tag{4.15}$$

Infine, sostituisco z_0 con l'equazione della distanza ottimale 4.13:

$$M_i = \frac{4\Delta_x(D_x + M\Delta_x)}{\lambda}\frac{\lambda}{\Delta_x^2} = \frac{4(D_x + M\Delta_x)}{\Delta_x}. \tag{4.16}$$

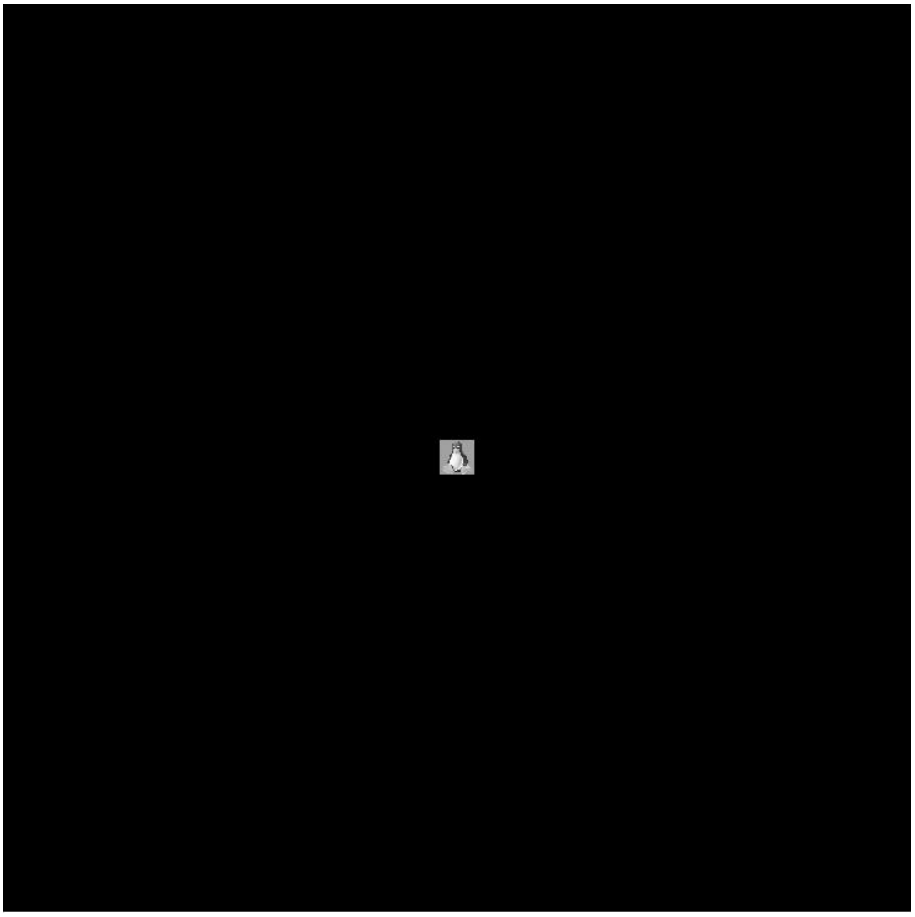

Figura 4.7
Ampiezza complessa dopo aver effettuato il zero-padding per conservare l'intervallo di campionamento iniziale se si utilizza il metodo S-FFT.

L'immagine caricata con i dati che andrò ad utilizzare per simulare la diffrazione della luce avrà una lunghezza ed una altezza di $\Delta_x \cdot$ numero pixel$= 4,65 \cdot 10^{-6} \cdot 256 = 1,1904 \cdot 10^{-3}$ m; quindi sto modellizzando un oggetto di dimesioni pari al quadratino qua di fianco $\longrightarrow$.

4.3.2 Parte 2: diffrazione dell'ampiezza complessa

In questa sezione simuleremò la diffrazione di un'onda piana con lunghezza d'onda di $0,65\mu m$ che incontra l'immagine caricata nella parte 1 nel caso di Fresnel, quindi andrò ad utilizzare l'equazione 3.18.
Siccome nella prima parte ho effettuato un zero-padding per conservare la grandezza dell'intervallo avrò che Q1 sarà uguale a Q2, perciò l'equazione sarà:

$$\Psi_p[m,n] = Q1 \cdot \mathit{fft}\{\Psi_{p0}[m,n] \cdot Q1\}\,. \tag{4.17}$$

Inoltre non calcolerò il fattore costante della 3.18 poichè dopo normalizzerò ad 1 il risultato, mentre il fattore di fase fuori la *fft* (che in questo caso è sempre Q1, ma teoricamente sarebbe

Q2) deve essere calcolato visto che poi dovrò andare a sommare il risultato con l'onda di riferimento prima di operare il valore assoluto per calcolare l'ologramma.

Parte 2.0: calcolo di $Q1(+z_0)$

Ho deciso di separare il calcolo di Q1 siccome è dispendioso ed impiega molto tempo, così una volta salvato in un file .txt ho potuto riutilizzarlo senza ogni volta ricalcolarlo.

```
% Q1

clear;
close all;

% dati dell‘ esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta_x=delta_y=delta=4.65*10^-6;% dimensione dei pixel
M=1390;                          % numero di pixel
D=256*4.65*10^-6;                % dimensione pinhole
z=(4*delta*(D+1390*delta))/l;    % distanza pinguino-ccd
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding

X0=Y0=axis_scaling(Mi,delta);

Q1=Q1_count(X0,Y0,l,z);

dlmwrite('q2_0_Q1.txt',Q1);
```

Ho preferito scrivere la funzione *Q1_count* per calcolarmi Q1 che riceve quattro argomenti, i primi due sono gli assi x ed y, mentre il terzo ed il quarto rispettivamente sono la lunghezza d'onda e la distanza fra l'ostacolo e dove si vuole conoscere la funzione d'onda diffratta. Il codice dello script *Q1_count* l'ho scritto in appendice.

Parte 2.1: simulazione della diffrazione a distanza +z dell'ampiezza complessa

```
clear;
close all;

U0=dlmread('q1_ampiezza_complessa.txt');
Q1=dlmread('q2_Q1.txt');

% dati dell' esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta_x=delta_y=delta=4.65*10^-6;% dimensione dei pixel
M=1390;                          % numero massimo di pixel del CCD
D=256*4.65*10^-6;                % dimensione pinguino
z=(4*delta*(D+1390*delta))/l;    % distanza pinguino-ccd in metri
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding


X0=Y0=axis_scaling(Mi,delta);
x_ccd=axis_scaling(1390,delta);
y_ccd=axis_scaling(1038,delta);

% METODO S-FFT
Uf=Q1.*fftshift(fft2(U0.*Q1));
Uf=normalization_2D(Uf);

Uf_ccd=Uf(2774:3811,2598:3987);
dlmwrite('q3_diffrazione_pinguino.txt',Uf_ccd);
```

Dopo aver calcolato la funzione d'onda diffratta su di una matrice ingrandita dal zero padding vado a prendere i valori centrali, riducendola ad una matrice $1390 \cdot 1038$ (il numero di pixel del CCD).

4.3.3 Parte 3: onda di riferimento

Calcolo l'equazione 1.3 dell'onda di riferimento su di una matrice $1390 \cdot 1038$ con degli intervalli di $4,65 \mu m$ in modo da simulare l'acquisizione da parte del CCD.

```
% ONDA DI RIFERIMENTO

clear;
close all;

% dati dell‘ esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta=4.65*10^-6;                % dimensione dei pixel
k=(2*pi)/l;                      % modulo vettore d'onda
theta=(asin((3*l)/(8*delta)));   % angolo off-axis holography
R0=1;                            % ampiezza onda riferimento


x_ccd=axis_scaling(1390,delta);
y_ccd=axis_scaling(1038,delta);

cost=j*k*sin(theta);
for a=1:1390
 for b=1:1038
 inclinazione(b,a)=exp(cost*(x_ccd(a)+y_ccd(b)));
 endfor
endfor

R=R0*inclinazione;

dlmwrite('q4_onda_riferimento.txt',R);
```

4.3.4 Parte 4: ologramma

Calcolo dell'ologramma, quindi si utilizza l'equazione 1.4.

```
% OLOGRAMMA
clear;
close all;

diffrazione_pinguino=dlmread('q3_diffrazione_pinguino.txt');
R=dlmread('q4_onda_riferimento.txt');

H=(diffrazione_pinguino+R).*conj(diffrazione_pinguino+R);
H_norm=normalization_2D(H);

dlmwrite('q5_ologramma.txt',H_norm);
```

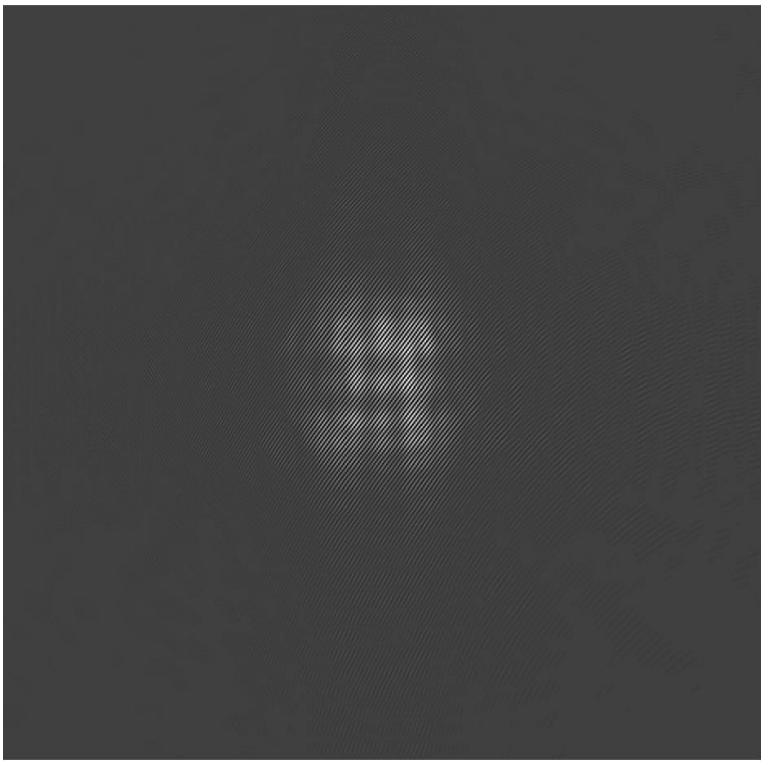

Figura 4.8
Simulazione della cattura dell'immagine dell'ologramma da parte del CCD, su ogni pixel viene calcolato il modulo al quadrato della interferenza prodotta dall'onda oggetto con l'onda di riferimento.

4.3.5 Parte 5: immagine reale

In questa parte ricostruirò l'immagine reale simulando la diffrazione di un'onda piana sull'ologramma a distanza z_0; per fare ciò andrò a riutilizzare il Q1 calcolato precedentemente ma l'equazione che elaborerò sarà:

$$\Psi_p[m,n] = \mathit{fft}\{\Psi_{p_0} \cdot Q1\}\,. \tag{4.18}$$

In questa momento non ho più bisogno del fattore di fase siccome andrò a fare il valore assoluto della funzione d'onda diffratta.

```
% RICOSTRUZIONE IMMAGINE REALE!!
clear;
close all;

H=dlmread('q4_ologramma.txt');
Q1=dlmread('q2_0_Q1.txt');

% dati dell' esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta_x=delta_y=delta=4.65*10^-6;% dimensione dei pixel
M=1390;                          % numero di pixel
D=256*delta;                     % dimensione pinguino
z=(4*delta*(D+1390*delta))/l;    % distanza pinguino-ccd
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding

H_zero_paddings=zero_padding(H,Mi);

X0=Y0=axis_scaling(Mi,delta);

% METODO S-FFT
Ui=ricostruzione=fftshift(fft2(fftshift(H_zero_paddings.*Q1)));

I=(abs(Ui)).^2;
I=normalization_2D(I);
gmin=min(min(I));
gmax=max(max(I));
p=input('Display parameter (>1) (0=end) : ');
while isempty(p) == 0
figure,imagesc(X0,Y0,I,[gmin,gmax/p]),colormap(gray),axis('tight','xy');
p=input('Display parameter (>1) (0=end) : ');
if p==0,
```

```
break
endif
endwhile
%-------------------------------------------------------------------------%
```

La maggior parte dell'energia, durante la ricostruzione, è distribuita nello zero-order e per rendere ben visibili le immagini ricostruite si va a riscalare la scala di grigi dell'immagine ricostruita. Nel programma ho inserito il parametro 'p', il quale divide l'elemento più grande della matrice e, grazie alla funzione imagesc, i colori si ridistribuiscono fra il valore minimo 'gmin' ed il valore massimo diviso p 'gmax/p'.

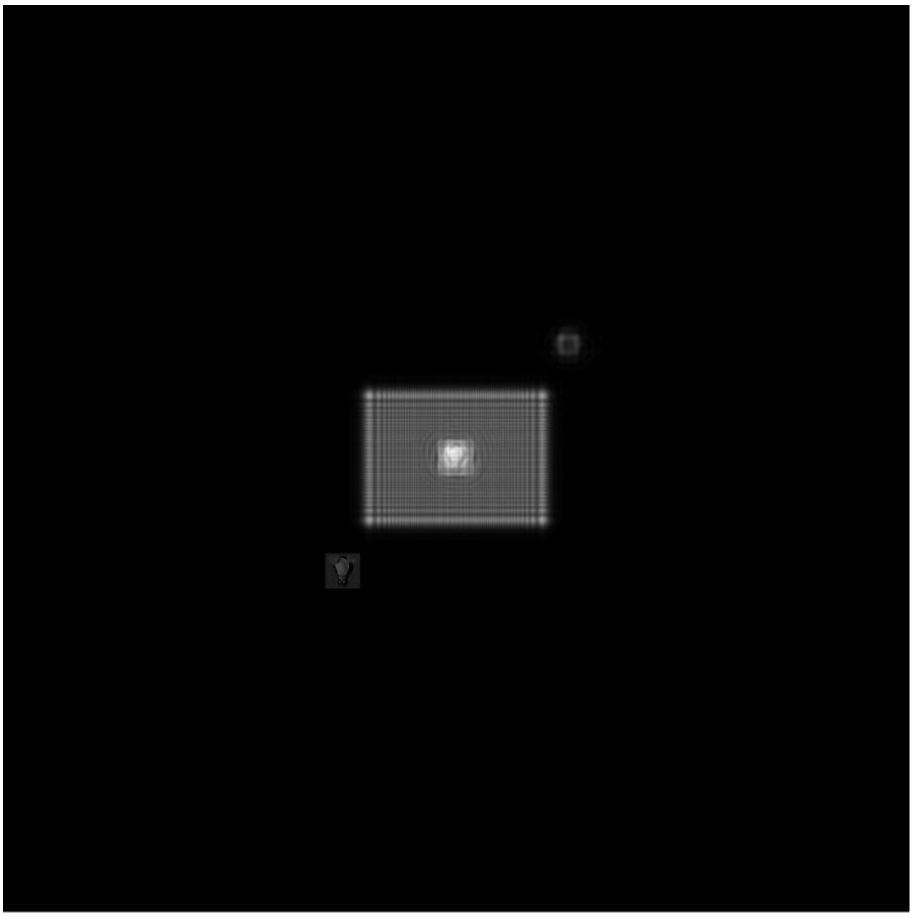

Figura 4.9

Ricostruzione dell'immagine reale con il parametro p=1, dall'immagine si può gia osservare la ricostruzione dell'immagine dell'oggetto iniziale, anche se risulta essere più scura.

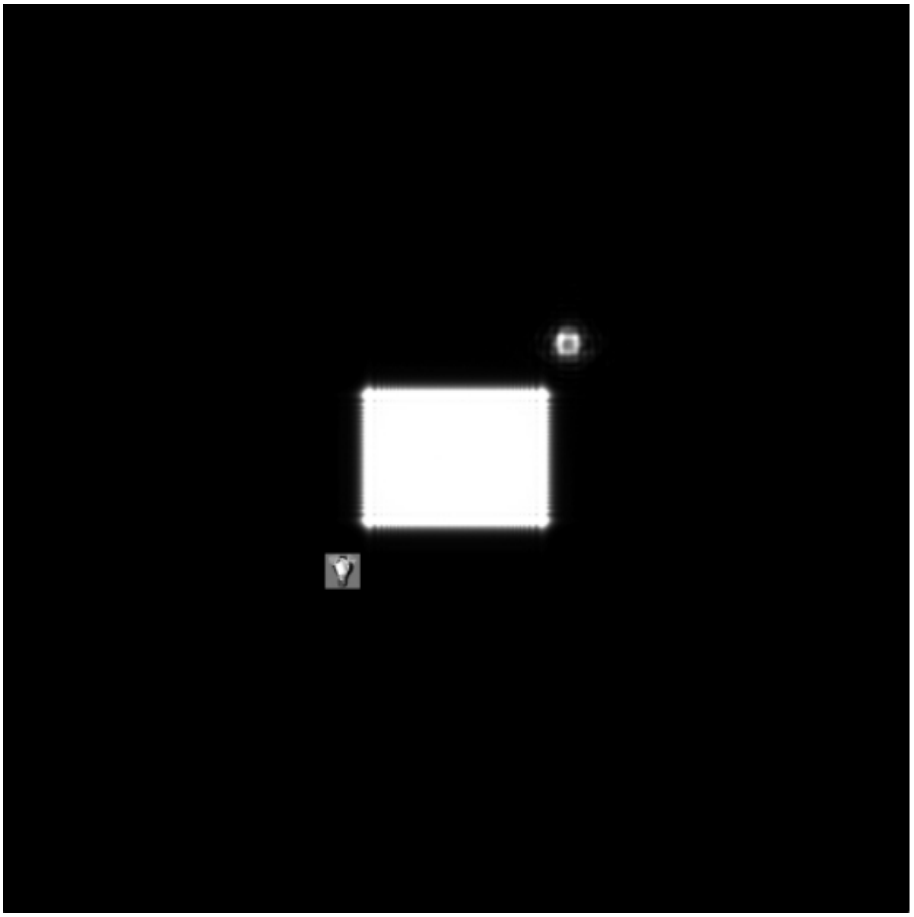

Figura 4.10

Ricostruzione dell'immagine reale con il parametro p=4: il processo di ricostruzione è lo stesso di quello in figura 4.9, ma ho schiarito l'immagine per far risaltare l'oggetto iniziale.

4.3.6 Parte 6: immagine virtuale

Per la ricostruzione dell'immagine virtuale si dovrà ricalcolare Q1 poichè la figura si forma ad una distanza $-z_0$ dall'ologramma; quindi nella parte 6.0 si calcola il fattore Q1, solo cambiando la distanza z_0, poi nella parte 6.1 si andrà a simulare la diffrazione dell'onda piana sull'ologramma con il consueto metodo.

Parte 6.0: calcolo di $Q1(-z_0)$

```
% Q1(-z)

clear;
close all;

% dati dell‘ esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta_x=delta_y=delta=4.65*10^-6;% dimensione dei pixel
M=1390;                          % numero di pixel
D=256*4.65*10^-6;                % dimensione pinguino
z=(4*delta*(D+1390*delta))/l;   % distanza pinguino-ccd
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding

X0=Y0=axis_scaling(Mi,delta);

Q1=Q1_count(X0,Y0,l,-z);

dlmwrite('q6_0_Q1(-z).txt',Q1);
```

parte 6.1: ricostruzione immagine virtuale a distanza $-z_0$ dall'ologramma

```
% ricostruzione -z

clear;
close all;

H=dlmread('q4_ologramma.txt');
Q1=dlmread('q6_0_Q1(-z).txt');

% dati dell‘ esperimento in metri
l=0.65*10^-6;                    % lunghezza d'onda
delta=4.65*10^-6;                % dimensione dei pixel
M=1390;                          % numero di pixel
D=256*delta;                     % dimensione pinhole
z=(4*delta*(D+1390*delta))/l;    % distanza pinhole-ccd
k=(2*pi)/l;                      % modulo vettore d'onda
Mi=(4*(D+M*delta))/delta;        % numero pixel zero padding
```

```
H_zero_paddings=zero_padding(H,Mi);

X0=Y0=axis_scaling(Mi,delta);

% METODO S-FFT
Ui=ricostruzione=fftshift(fft2(fftshift(H_zero_paddings.*Q1)));

I=(abs(Ui)).^2;
I=normalization_2D(I);
gmin=min(min(I));
gmax=max(max(I));
p=input('Display parameter (>1) (0=end) : ');
while isempty(p) == 0
figure,imagesc(X0,Y0,I,[gmin,gmax/p]),colormap(gray),axis('tight','xy');
p=input('Display parameter (>1) (0=end) : ');
if p==0,
break
endif
endwhile
```

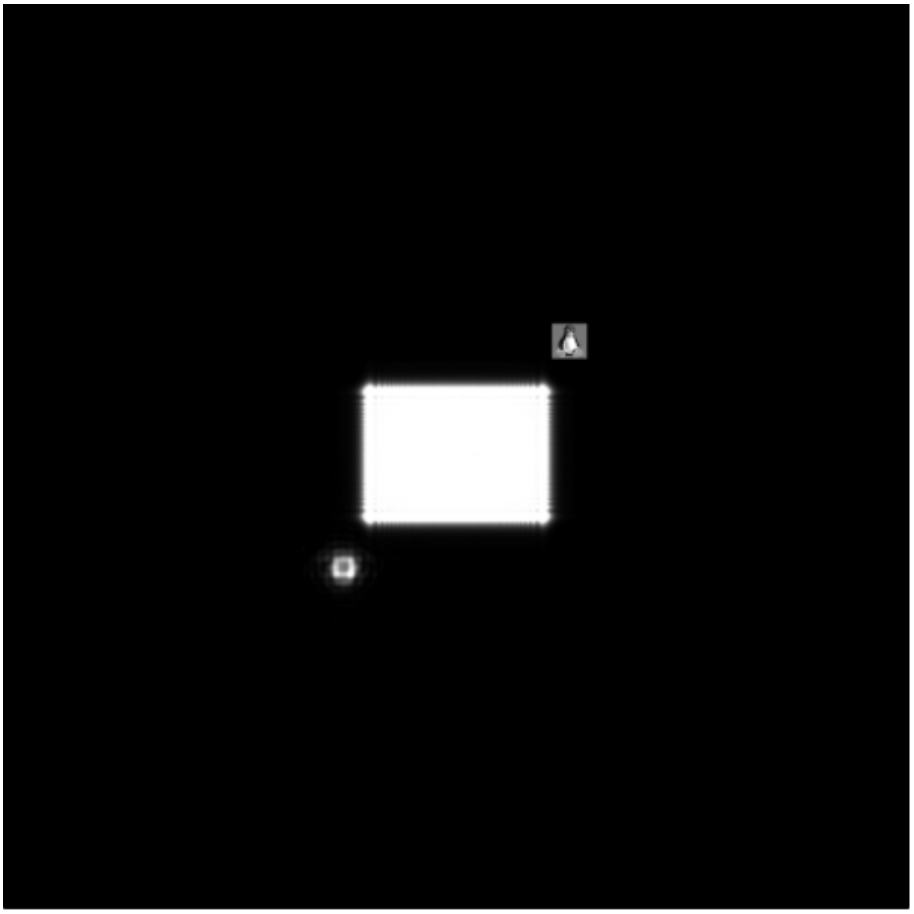

Figura 4.11
Ricostruzione dell'immagine virtuale con il parametro p=4: per la ricostruzione dell'immagine virtuale ho simulato la diffrazione alla distanza $-z_0$ siccome l'immagine si va formare nel piano dell'oggeto, solo traslata dall'inclinazione dell'onda di riferimento.

Infine vado a ritagliare dall'ultima figura l'immagine virtuale per metterla meglio in mostra:

Figura 4.12
Immagine virtuale: ho ritagliato dalla figura 4.11 la sezione in cui viene ricostruita l'immagine virtuale.

In figura 4.12 si può osservare la ricostruzione dell'immagine virtuale, simulando la diffrazione sull'ologramma con il metodo S-FFT; l'alternanza di piccoli chiari scuri che vanno a disturbare la qualità dell'immagine ricostruita sono riconducibili alla grata cosisunoidale introdotta nell'equazione 2.29. In questo programma ho cercato di simulare il più possibile la cattura di un ologramma da parte di un CCD, anche se nel caso reale ci sarebbero altri fattori non presi in considerazione come la non perfetta coerenza, spaziale e temporale, del laser e gli effetti prodotti dal vetrino disposto davanti alla camera del CCD, i quali andrebbero a creare nuovi disturbi durante il processo.

4.4 Simulazione della profondità dell'ologramma

In questa ultima sezione illustrerò un programma grazie al quale elaborerò due immagini diverse a due distanze diverse sullo stesso ologramma; in questo modo metto in evidenza le proprietà tridimensionali della tecnica olografica.
Anche in questo caso ho diviso il programma in più script, per suddividere il carico computazionale, andando a scrivere/leggere i risultati grazie alle funzioni *dlmwrite, dlmread*.
Riprenderò i dati usati nel programma nel pinguino con la differenza che, per semplicità, il piano di registrazione sarà una matrice $1024 \cdot 1024$ quadrata. Per lo svolgimento ho preso le due seguenti immagini:

Figura 4.13.1

Figura 4.13.2

Le due immagini iniziali, scaricate da internet, scelte per lo svolgimento del programma; tutte e due sono composte da $64 \cdot 64$ *pixel.*

Il programma inizierà simulando la diffrazione, con il metodo S-FFT, di *ghost_64.png* per poi andare a comporre la funzione d'onda diffratta con la seconda immagine *pacman_64.png*.
Successivamente simulerò la diffrazione sul risultato del precedente passaggio ed utilizzerò l'onda diffratta per sommarla all'onda di riferimento, quindi calcolerò l'ologramma.
Infine riandrò a simulare la diffrazione sull'ologramma per le due distanze delle immagini rispetto al piano di registrazione, ricostruendo le lo loro immagini reali una alla volta.
Nella figura successiva è riportato lo schema del programma:

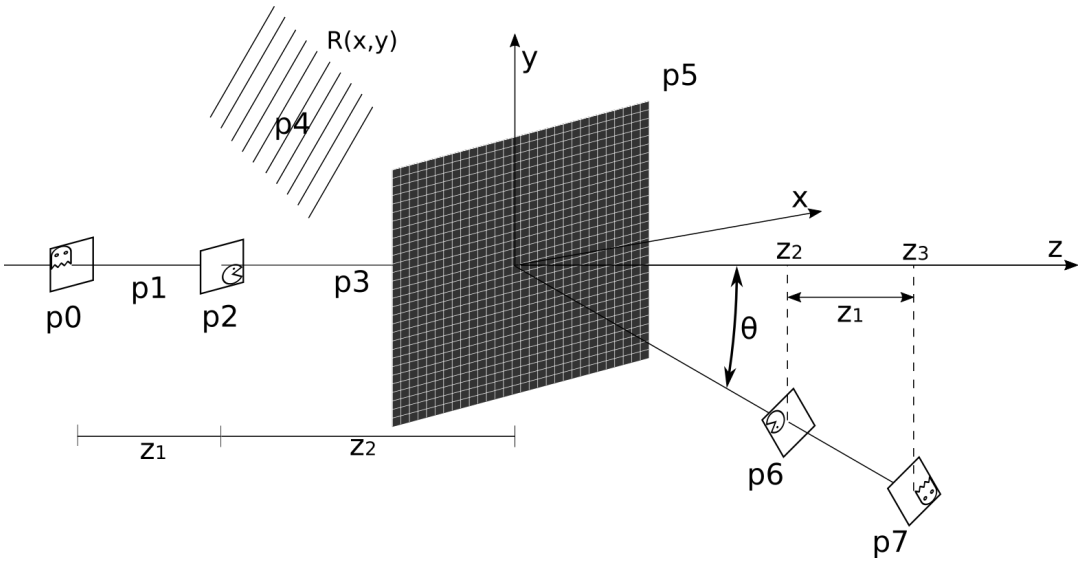

Figura 4.14

Nello schema ho raffigurato le parti del programma ($p_{0\to7}$) che ho sviluppato per ottenere la ricostruzione delle immagini dei due oggeti a due diverse distanze, mettendo in mostra la tridimensionalità della tecnica olografica.

Gli script che andrò ad illustrare sono:

- parte 0: *Apro l'immagine ghost_64.png e la pongo nel quadrante in alto a sinistra di una matrice* $128 \cdot 128$ *di uno, poichè nella parte 3 andrò a posizionare pacman_64 nella parte in basso a destra.*

- parte 1.0: *Calcolo della funzione di trasferimento dell'integrale di Fresnel, osservando che rispetti la condizione 3.23.*
  parte 1.1: *Simulazione della diffrazione di Fresnel dell'ampiezza complessa (salvata nella parte 0), utilizzando la funzione di trasferimento calcolata nella 1.0.*

- parte 2: *Compongo la funzione d'onda diffratta calcolata nella parte 1.1, con l'immagine di pacman posizionata nel quadrante in basso a destra di una matrice* $128 \cdot 128$ *di uno.*

- parte 3.0: *Calcolo della funzione di trasferimento dell'integrale di Fresnel, però in questo caso dovrà rispettare la condizione 4.13 data dall'olografia fuori asse.*
  parte 3.1: *Simulazione della diffrazione dell'oggetto salvato nella parte 2 con la funzione di trasferimento della 3.0 .*

- parte 4: *Calcolo dell'onda di riferimento.*

- parte 5: *Calcolo dell'ologramma.*

- parte 6.0: *Ricostruzione dell'immagine reale di pacman utilizzando la funzione di trasferimento calcolata nella parte 1.0.*

- parte 7.0: *Calcolo della funzione di trasferimento per la ricostruzione dell'immagine del fantasma.*
  parte 7.1: *Ricostruzione dell'immagine del fantasma*

4.4.1 parte 0: ampiezza complessa fantasma

In questo piccolo script eseguo quanto scritto sopra senza altro da aggiungere.

```
clear;
close all;

% apro l'immagine e la trasformo in una funzione di trasmittanza
% della luce
U0_G=imread('ghost_64.png');
U0_G=mat2gray(U0_G);
U0_G=rgb2gray(U0_G);
U0_G=normalization_2D(U0_G);

U0=ones(128);

% posiziono il fantasma in scala di grigi sul quadrante in altro a sinistra
% della matrice U0
for a=1:64;
 for b=1:64;
U0(b,a)=U0_G(b,a);
 endfor
endfor

dlmwrite('q0.txt',U0);
```

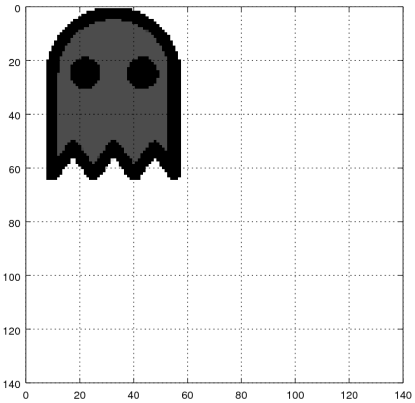

Figura 4.15

La figura mosta il risultato dello script p0 grazie al quale trasformo l'immagine in scala di grigi, per poi posizionarla nel quadrante in alto a sinista di una matrice $128 \cdot 128$.

4.4.2 parte 1: diffrazione fantasma

In questa parte 1 andrò a calcolare la funzione d'onda che arriva sul pacman dopo aver diffratto sul fantasma, che quindi ha percorso la distanza z1.

parte 1.0: Q1(z1)

Con questo script vado a calcolare la funzione di trasferimento per la diffrazione di Fresnel della luce che incontra l'immagine ghost_64 per arrivare all'immagine pacman_64 a distanza $z1$.

```
clear;
close all;

l=0.65*10^-6;                  % lunghezza d'onda in metri
delta=4.65*10^-6;              % intervallo di campionamento in metri
k=(2*pi)/l;                    % modulo vettore d'onda
m=n=128;                       % numero di campionamenti sull'oggetto
M=N=1024;                      % numero campionamenti del piano di registrazione
M1=N1=2052;                    % zero-padding
z1=(delta^2*M1)/l;             % distanza fantasma-pacman

X=Y=axis_scaling(M1,delta);

Q1=Q1_count(X,Y,l,z1);

dlmwrite('q1_0.txt',Q1);
```

Per calcolare la distanza $z1$ ho preso in considerazione la condizione 3.23, a cui vado a sostituire i valori presi in considerazione:

$$z_0 \geq \frac{2M\Delta_x^2}{\lambda} = \frac{2 \cdot 1024 \cdot (4,65 \cdot 10^{-6})^2}{0,65 \cdot 10^{-6}} \simeq 6,813 \cdot 10^{-2} m. \tag{4.19}$$

Per conservare l'intervallo di campionamento con l'integrale di Fresnel ho applicato un zero padding osservando la relazione 4.15, quindi:

$$M_{zero\ padding} = \frac{\lambda \cdot z_0}{\Delta_x^2} = \frac{\lambda}{\Delta_x^2} \frac{2 \cdot M \cdot \Delta_x^2}{\lambda} = 2 \cdot M = 2048. \tag{4.20}$$

Nel programma ho preferito aggiungere qualche altro zero intorno all'immagine in modo di aumentare, se pur di poco, la distanza in modo da evitare errori. Il valore che ho assegnato ad $M_{zero\ padding}$ è di 2052, il quale corrisponde ad una distanza $z1 \simeq 6,826 \cdot 10^{-2} m$.

parte 1.1: Metodo S-FFT sul fantasma per la distanza z1

Adesso calcolerò la funzione d'onda diffratta dal fantasma ad una distanza z_1, utilizzando il metodo S-FFT.

```
clear;
close all;

M1=N1=2052;                    % zero-padding

U0_G=dlmread('q0.txt');
Q1=dlmread('q1_0.txt');

U0_G_padding=zero_padding(U0_G,M1);

% METODO S-FFT
Uf_U0_G_padding=Q1.*fftshift(fft2(fftshift(U0_G_padding.*Q1)));
Uf_U0_G_padding=normalization_2D(Uf_U0_G_padding);

dlmwrite('q1_1.txt',Uf_U0_G_padding);
```

4.4.3 parte 2: composizione dell'onda diffratta dal fantasma con pacman

In questa parte 2 andrò a comporre l'onda diffratta del fantasma con la funzione di trasmittanza del pacman; il pacman, al contrario del fantasma, lo posiziono nel quadrante in basso a destra di una matrice di dimensione $128 \cdot 128$ formata da uno.

```
clear;
close all;

m=n=128;                       % numero di campionamenti sull'oggetto
M=N=1024;                      % numero campionamenti del piano di registrazione

% condizione dell'olografia fuori asse per conservare l'intervallo
% di campionamento
N2=M2=4*(m+M);

% funzione d'onda diffratta dal fantasma
```

```
Uf_G=dlmread('q1_1.txt');

% apro l'immagine di pacman e la trasformo in una funzione di trasmittanza
U0_P=imread('pacman_64.png');
U0_P=mat2gray(U0_P);
U0_P=rgb2gray(U0_P);
U0_P=normalization_2D(U0_P);

% posizione pacman in scala di grigi nel riquadro in basso a di una
% matrice 128*128 di uno
U0_P_128=ones(128);
for a=1:64;
 for b=1:64;
 U0_P_128(b+64,a+64)=U0_P(b,a);
 endfor
endfor

% applico uno zero padding per conservare l'intervallo di campionamento
% rifacendo gli stessi ragionameti fatti per il programma precedente del
% pinguino, prendendo in considerazionr le condizioni portate dall'ologra-
% fia fuori asse
Uf_G_padding=zero_padding(Uf_G,M2);

% In questo caso invece applico degli uno intorno al pacman sennò quando
% andrò a comporre la funzione d'onda diffratta del fantasma perderei
% l'informazione al di fuori dei 128*128 pixel centrali
U0_P_padding=one_padding(U0_P_128,M2);

% composizione dell'onda del fantasma diffratta con la funzione di
% trasmittanza del pacman
U0_padding=Uf_G_padding.*U0_P_padding;

dlmwrite('q2.txt',U0_padding);
```

4.4.4 parte 3: propagazione dell'onda dal pacman al piano di registrazione

Simulerò la diffrazione dell'onda che parte dal pacman (quindi il risultato calcolato nella parte 2) e che raggiunge il piano di registrazione, la quale andrà a comporre insieme all'onda di riferimento l'ologramma.

parte 3.0: Q1(z2)

Elaborazione della funzione Q1 per una distanza z2, che in questo caso sarà la distanza critica 4.13 data dalle condizioni dell'olografia fuori asse.

```
clear;
close all;

l=0.65*10^-6;                   % lunghezza d'onda in metri
delta=4.65*10^-6;               % intervallo di campionamento in metri
k=(2*pi)/l;                     % modulo vettore d'onda
m=n=128;                        % numero di campionamenti sull'oggetto
M=N=1024;                       % numero campionamenti del piano di registrazione

% condizione olografia fuori asse
N2=M2=4*(m+M);

z2=(delta^2*M2)/l;              % distanza pacman-piano di registrazione

X=Y=axis_scaling(M2,delta);
Q1=Q1_count(X,Y,l,z2);

dlmwrite('q3_0.txt',Q1);
```

---

parte 3.1: metodo S-FFT dal pacman per la distanza z2

Simulazione della diffrazione dell'onda dal fantasma al piano di registrazione, che ho deciso di formarlo da una matrice $1024 \cdot 1024$.

```
clear;
close all;
% apro in U0_padding il risultato della parte 2, quindi l'onda diffratta
% dal fantasma che ho poi composto con la funzione di trasmittanza del
```

```
% pacman ed a cui ho applicato uno zero padding per conservare l'intervallo
% di campionamento.
% In Q1 apro la funzione di trasferimento per la distanza z2, che equivale
% alla distanza critica dell'olografia fuori asse.
U0_padding=dlmread('q2.txt');
Q1=dlmread('q3_0.txt');

% METODO S-FFT
Uf_padding=Q1.*fftshift(fft2(fftshift(U0_padding.*Q1)));
Uf_padding=normalization_2D(Uf_padding);

% Siccome fin da l'inizio ho deciso che il piano di registrazione è composto
% da 1024*1024 pixel vado a ritagliare da Uf_U0_padding la parte centrale
% della matrice
Uf_1024=Uf_padding(1793:2816,1793:2816);

dlmwrite('q3_1.txt',Uf_1024);
```

4.4.5 parte 4: onda di riferimento

Calcolo dell'onda di riferimento sul piano di registrazione, la cui inclinazione è data dall'equazione 4.14 (condizione ottimale per l'olografia fuori asse).

```
clear;
close all;

l=0.65*10^-6;                  % lunghezza d'onda in metri
delta=4.65*10^-6;              % intervallo di campionamento in metri
k=(2*pi)/l;                    % modulo vettore d'onda
m=n=128;                       % numero di campionamenti sull'oggetto
M=N=1024;                      % numero campionamenti del piano di registrazione
theta=(asin((3*l)/(8*delta))); % inclinazione

X=Y=axis_scaling(M,delta);

cost=j*k*sin(theta);
for a=1:M
 for b=1:N
 R(b,a)=exp(cost*(X(a)+Y(b)));
 endfor
```

```
endfor

dlmwrite('q4.txt',R);
```

4.4.6 parte 5: ologramma

Calcolo l'ologramma, quindi il modulo quadro della somma dell'onda diffratta dal fantasma e dal pacman con l'onda di riferimento.

```
% OLOGRAMMA
clear;
close all;

Uf_GP=dlmread('q3_1.txt');
R=dlmread('q4.txt');

H=(Uf_GP+R).*conj(Uf_GP+R);
H_norm=normalization_2D(H);

dlmwrite('q5.txt',H_norm);
```

4.4.7 parte 6 ricostruzione immagine reale del pacman

Con questo script ricostruirò l'immagine reale del pacman che si formerà alla distanza z2 dall'ologramma, quindi utilizzerò il Q1 calcolato nella parte 3.0 .

```
clear;
close all;

l=0.65*10^-6;                % lunghezza d'onda in metri
delta=4.65*10^-6;            % intervallo di campionamento in metri
k=(2*pi)/l;                  % modulo vettore d'onda
m=n=128;                     % numero di campionamenti sull'oggetto
M=N=1024;                    % numero campionamenti del piano di registrazione

H=dlmread('q5.txt');         % ologramma
Q1=dlmread('q3_0.txt');      % funzione di trasferimento per la distanza z2
```

```
% condizione olografia fuori asse
N2=M2=4*(m+M);

z2=(delta^2*M2)/l;              % distanza pacman-piano di registrazione

X=Y=axis_scaling(N2,delta);
H_padding=zero_padding(H,M2);

% METODO S-FFT
Ui=fftshift(fft2(fftshift(H_padding.*Q1)));

I=(abs(Ui)).^2;
I=normalization_2D(I);
gmin=min(min(I));
gmax=max(max(I));
p=input('Display parameter (>1) (0=end) : ');
while isempty(p) == 0
figure,imagesc(X,Y,I,[gmin,gmax/p]),colormap(gray),axis('tight','xy');
p=input('Display parameter (>1) (0=end) : ');
if p==0,
break
endif
endwhile
```

Il risultato di questo script è illustrato nella seguente figura:

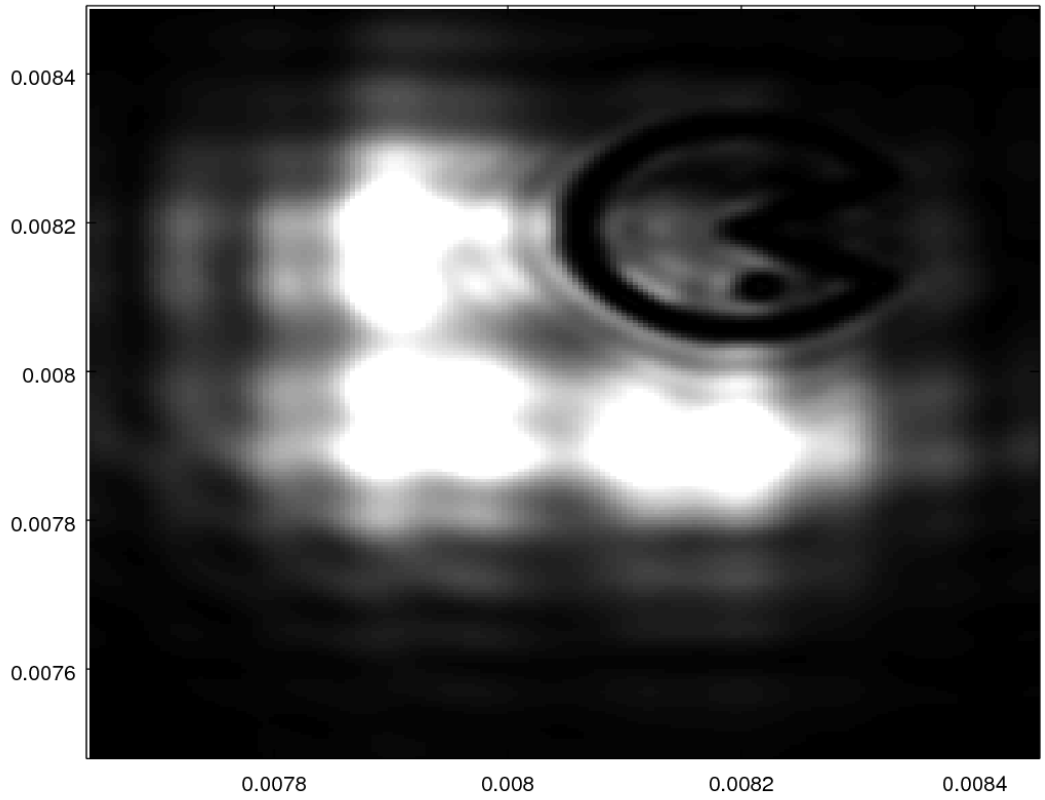

Figura 4.16
In questa figura posso osservare la ricostruzione dell'immagine reale pacman che si forma alla distanza z2 dal piano di registrazione.

Nella figura 4.16, come per il programma precedente del pinguino, ho riscalato la scala di colori per mettere in evidenza l'immagine del pacman, assegnando alla variabile p il valore di 4.

4.4.8 parte 7: ricostruzione immagine reale del fantasma

Ricostruirò l'immagine reale del fantasma che si formerà alla distanza $z_1 + z_2$ dal piano di registrazione, quindi dovrò calcolare una nuova funzione di trasferimento per questa distanza, che chiamo z_3 nello script. Per conservare l'intervallo di campionamento dovrò applicare un zero-padding abbastanza grande che ho calcolato prendendo in considerazione sempre la relazione 4.15:

$$M_3 = \frac{\lambda z_3}{\Delta_x^2} = \frac{\lambda(z_1 + z_2)}{\Delta_x^2} = \frac{\lambda}{\Delta_x^2}\frac{\Delta_x^2}{\lambda}(M1 + M2) = M1 + M2; \tag{4.21}$$

parte 7.0: Q1(z3)

Calcolo della funzione di trasferimento per la distanza z_3 in cui si ricostruirà l'immagine reale del fantasma.

```
clear;
close all;


l=0.65*10^-6;                   % lunghezza d'onda in metri
delta=4.65*10^-6;               % intervallo di campionamento in metri
k=(2*pi)/l;                     % modulo vettore d'onda
m=n=128;                        % numero di campionamenti sull'oggetto
M=N=1024;                       % numero campionamenti del piano di registrazione

M1=N1=2052;                     % zero-padding per z1
M2=N2=4*(m+M);                  % pero-pagging per z2

% zero-padding per la distanza z3=z1+z2 dove si andrà a ricostruire l'immagine
% reale del fantasma
M3=N3=M1+M2;

z3=(delta^2*M3)/l;              % distanza piano di registrazione-fantasma

X=Y=axis_scaling(M3,delta);

Q1=Q1_count(X,Y,l,z3);

dlmwrite('q7_0.txt',Q1);
```

---

parte 7.1: ricostruzione dell'immagine reale del fantasma

Adesso andrò a ricostruire l'immagine reale del fantasma simulando la diffrazione della luce sull'ologramma fino alla distanza z_3.

```
clear;
close all;

l=0.65*10^-6;                   % lunghezza d'onda in metri
delta=4.65*10^-6;               % intervallo di campionamento in metri
k=(2*pi)/l;                     % modulo vettore d'onda
m=n=128;                        % numero di campionamenti sull'oggetto
M=N=1024;                       % numero campionamenti del piano di registrazione

H=dlmread('q5.txt');            % ologramma
Q1=dlmread('q7_0.txt');         % funzione di trasferimento per la distanza z3

M1=N1=2052;                     % zero-padding per z1
M2=N2=4*(m+M);                  % zero-pagging per z2

% zero-padding per la distanza z3=z1+z2 dove si andrà a ricostruire l'immagine
% reale del fantasma
M3=N3=M1+M2;

X=Y=axis_scaling(N3,delta);

H_padding=zero_padding(H,N3);

% METODO S-FFT
Ui=fftshift(fft2(H_padding.*Q1));

I=(abs(Ui)).^2;
I=normalization_2D(I);
gmin=min(min(I));
gmax=max(max(I));
p=input('Display parameter (>1) (0=end) : ');
while isempty(p) == 0
figure,imagesc(X,Y,I,[gmin,gmax/p]),colormap(gray),axis('tight','xy');
p=input('Display parameter (>1) (0=end) : ');
if p==0,
break
endif
endwhile
```

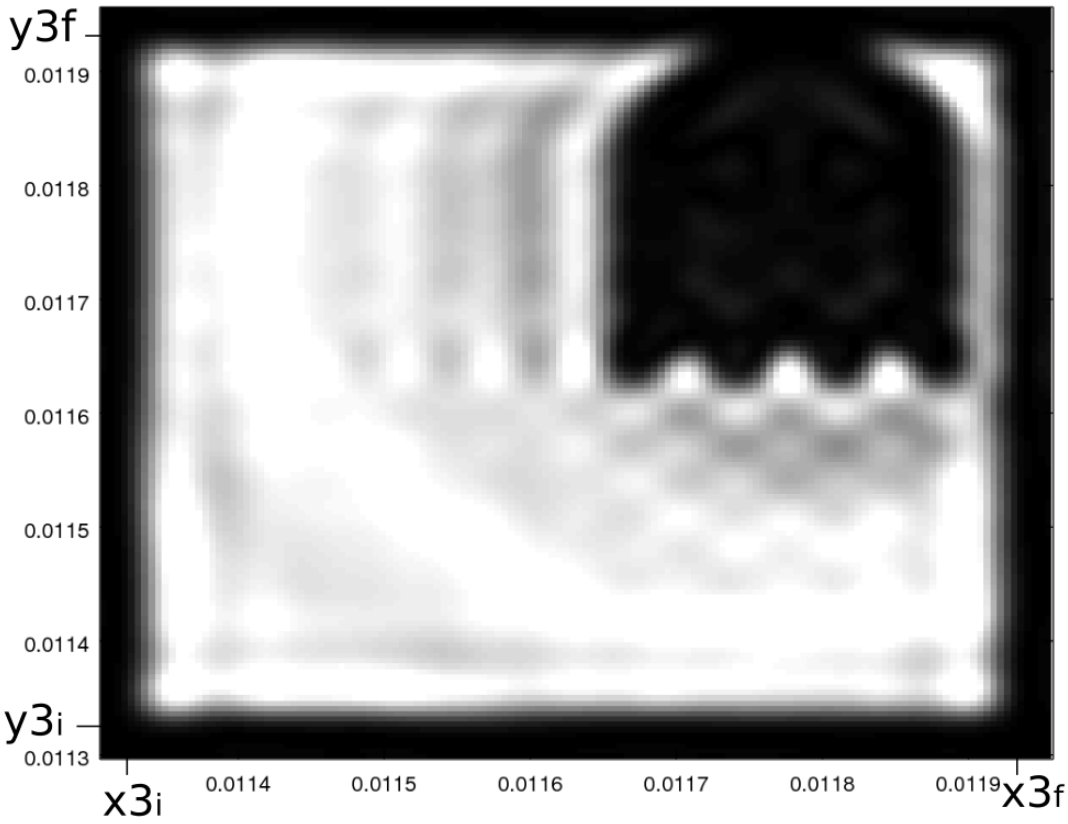

Figura 4.17

Nella figura posso osservare la ricostruzione dell'immagine del fantasma che si forma a distanza z_3, in cui c'è più traccia del fantasma.

Nella figura 4.17 non si forma l'immagine del pacman siccome ho simulato la diffrazione della luce alla distanza z_3 andando a provare che l'ologramma possiede in se l'informazione della profondità, anche essendo una lastra bidimensionale.
Posso verificare che l'immagine ricostruita al computer si posiziona correttemente sul piano xy; il fantasma, come si può vedere dalla figura, si ricostruisce nell'intervallo che va da $x3_i = y3_i = 1,132 \cdot 10^{-2}m$ a $x3_f = y3_f = 1,192 \cdot 10^{-2}m$. Teoricamente l'immagine si posiziona a distanza z_3 dall'ologramma traslata sul piano xy dall'angolo θ dell'onda di riferimento, utilizzando le formule trigonometriche del triangolo rettangolo posso ricavarmi le coordinate in x ed y:
$x3_i = y3_i = z3 \cdot tan(\theta) - 0,5 \cdot (128 \cdot \Delta) = 1,1332 \cdot 10^{-2}m$,
$x3_f = y3_f = z3 \cdot tan(\theta) + 0,5 \cdot (128 \cdot \Delta) = 1,1927 \cdot 10^{-2}m$.
Quindi le coordinate in cui si va a formare l'immagine nel piano xy elaborate al computer sono in accordo con i risultati analitici.

Conclusioni

Questo scritto ha lo scopo di illustrare i principi della olografia digitale, analizzando i metodi numerici grazie ai quali è possibile simulare al computer la diffrazione della luce su di un ostacolo.
Nei primi due capitoli ho ripercorso la fasi principali della tecnica olografica la quale è sempre composta da due passaggi, prima la registrazione dell'ologramma e poi la ricostruzione dell'immagine; successivamente ho illustrato la teoria della diffrazione con cui ho potuto dare una descrizione matematica al fenomeno della diffrazione della luce.
Nel terzo capitolo sono andato a studiare i metodi numerici D-FFT e S-FFT grazie ai quali si può simulare al computer la diffrazione applicando un particolare accorgimento in relazione alle condizioni che sorgono dalla digitalizzazione del segnale; per il metodo S-FFT ho confrontato i valori teorici attesi con quelli elaborati al computer per la diffrazione di un'onda piana su di un pinhole riscontrando un buon accordo.
Infine, nel quarto capitolo, ho scritto due programmi che ripercorrono le fasi della *Computer Generated Holography* mettendo in mostra le principali funzioni di questa tecnica, cioè di saper ricostruire le immagini di oggetti molto piccoli e di saper elaborare la tridimensionalità dell'oggetto in questione.
Nel primo, in cui ho simulato il processo olografico utilizzando l'immagine di un pinguino (*Tux*, mascotte del kernel Linux), ho impostato il programma in modo che la cattura dell'ologramma somigliasse il più possibile alla registrazione sul CCD utilizzato in degli esperimenti in laboratorio; l'immagine presa in considerazione con i dati in input simula un oggetto di circa $1,2 \cdot 10^{-3} m$, riuscendo a ricostruire un'immagine con una buona risoluzione.
Nel secondo programma ho simulato la registrazione sull'ologramma di due oggetti a due distanze diverse dal piano di registrazione, per poi andare a ricostruire le due immagini separatamente cambiando la funzione di trasferimento per le due distanze in considerazione. Il programma simulando la diffrazione della luce del laser sull'ologramma riesce a metter a fuoco due piani diversi, che stanno a due diverse distanze dall'ologramma stesso, in modo da poter ricostruire indipendentemente l'immagine del fantasma o di pacman.

Appendices

In appendice riporto gli script non inclusi in Octave che ho utilizzato nei programmi per snellirne il testo.

axis scaling

```
function[scaled_axis]=axis_scaling(intervals_numbers,interval_length)

for i=1:intervals_numbers
 scaled_axis(i)=((intervals_numbers/(intervals_numbers-1))*
 (interval_length*(i-1)))-(interval_length*intervals_numbers)/2;
endfor

endfunction
```

pinhole

```
function[pinhole_matrix]=pinhole(M,N,X,Y,R)

for a=1:N;
 for b=1:M;
  if (X(a)^2+Y(b)^2)^(1/2)<=R;
       pinhole_matrix(b,a)=1;
  else pinhole_matrix(b,a)=0;
  endif
 endfor
endfor

endfunction
```

normalization 2D

```
function[output]=normalization_2D(input)

Vmax=max(max(abs(input)));
output=input/Vmax;

endfunction
```

Q1_count

```
function[Q1]=Q1_count(X,Y,l,z)

cost=-((j*pi)/(l*z));
M=length(X);
N=length(Y);
for a=1:M;
 for b=1:N;
 Q1(b,a)=exp(cost*(X(a)^2+Y(b)^2));
 endfor
endfor

endfunction
```

zero_padding

```
function[output]=zero_padding(input,K)

[M,N]=size(input);
Z1=zeros(K,(K-N)/2);
Z2=zeros((K-M)/2,N);
output=[Z1,[Z2;input;Z2],Z1];

endfunction
```

one_padding

```
function[output]=one_padding(input,K)

[M,N]=size(input);
Z1=ones(K,(K-N)/2);
Z2=ones((K-M)/2,N);
output=[Z1,[Z2;input;Z2],Z1];

endfunction
```

Bibliografia

[1] Gabor D., *"A new microscopic principle". Nature 161, 777–778 (1948).*

[2] Leith E.N., Upatnieks J., *"Wavefront reconstruction with diffused illumination and three-dimensional objects," J.Opt. Soc. Am.54, 1295–1301 (1964).*

[3] Goodman J.W., Lawrence R.W., *Digital image formation from electronically detected holograms," Appl. Phys. Lett. 11(3), 77–79 (1967).*

[4] Kreis T. *"Applications of Digital Holography: From Microscopy to 3D-Television" J. Europ. Opt. Soc. Rap. Public. 7, 12006 (2012).*

[5] Picart P, Li C., *Digital Holography, John Wiley and Sons, 2013, paragrafo 4.1.1*

[6] De Dominicis L. *"Studio preliminare per olografia digitale", Tesi di Laurea Triennale, Bologna, 2015/2016.*

[7] Poon T.C., Liu J.P. *"Introduction to modern digital holography" 2014, paragrafo 2.2 .*

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery *"Numerical Recipes, The Art of Scientific Computing, Third Edition", 2007.*

[9] Poon T.C., Liu J.P. *"Introduction to modern digital holography" 2014, paragrafo 4.3 .*

[10] http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/cirapp2.html .

[11] Picart P, Li C., *Digital Holography, John Wiley and Sons, 2013, paragrafo 4.1.2 .*