

**I  
N  
A  
O  
E**

**OBTENCIÓN DEL FRENTE DE ONDA  
AUTOMATIZADO PARA SISTEMAS ÓPTICOS  
CONVERGENTES MEDIANTE LA  
INTERFEROMETRÍA DE DESPLAZAMIENTO  
LATERAL**

Por:

**Jaime Sánchez Paredes**

Tesis sometida como requisito parcial para obtener el  
grado de

**DOCTOR EN CIENCIAS EN LA  
ESPECIALIDAD DE ÓPTICA**

En el

**Instituto Nacional de Astrofísica, Óptica y  
Electrónica.**

Agosto 2013  
Tonantzintla, Puebla

Supervisada por:

**Dr. Jorge Castro Ramos**

© INAOE

Derechos Reservados

El autor otorga al INAOE el permiso de  
reproducir y distribuir copias de esta tesis en  
su totalidad o en partes.



# *RESUMEN*

Esta tesis de manera general se basa en la implementación de 2 dispositivos (uno con motores a pasos y el otro con una pantalla de cristal líquido) para probar sistemas ópticos convergentes de manera cualitativa y cuantitativa usando la Interferometría de Desplazamiento de Fase Lateral (PSI). Los desplazamientos de fase se obtienen de dos formas: en la primera colocamos una rejilla de Ronchi en una montura que está adaptada a unos motores a pasos desde los cuales se da el desplazamiento de fase deseado y el segundo método consiste en desplegar en una Pantalla de Cristal Líquido (LCD) rejillas de Ronchi desplazadas, las rejillas que se despliegan en la LCD son previamente creadas en un programa que hemos desarrollado. En ambos casos los patrones de franjas son digitalizados con una cámara CCD y son analizados mediante un software que hemos desarrollado. Con el método donde usamos una LCD se pueden implementar varias pruebas ópticas sin requerir ningún elemento óptico adicional usando la configuración de la prueba de Ronchi, tales pruebas son la de Foucault, la del alambre, la de Hartmann, la de Ronchi incluso la de Ronchi nula.

Existen diferentes algoritmos de PSI para tomar los patrones de franjas, de los cuales hemos implementado 8, estos difieren en el número de patrones de franjas que son digitalizados y en el tamaño de paso del desplazamiento de fase introducido.

Los patrones de franjas experimentales son digitalizados con una cámara CCD, cuando se digitaliza un patrón de franjas los píxeles de fase válida quedan dentro de una forma similar a la superficie bajo prueba en el patrón digitalizado. En la literatura existen unas ecuaciones de modulación de datos para encontrar los píxeles de fase válida, nosotros desarrollamos unas ecuaciones propias de modulación de datos (a las que llamamos ecuaciones fondo) que nos sirven para diferenciar los píxeles de fase válida de los de

fase no valida. Una vez que tenemos los píxeles de fase valida obtenemos una máscara a la cual le aplicamos los momentos geométricos para obtener el centroide y el radio de la fase valida en píxeles de los patrones de franjas digitalizados, las ecuaciones de fondo nos sirven en la diferenciación de los píxeles de fase valida de los de fase no valida y junto con los momentos geométricos obtenemos el radio y el centroide de la máscara de la fase valida, lo cual es de gran importancia y ayuda en el análisis de los patrones de franjas experimentales obteniendo buenos resultados.

Después obtenemos la fase envuelta de los patrones de franjas, posteriormente se desenvuelve la fase para esto hemos desarrollado 3 métodos diferentes de los muchos que hay en la literatura.

Finalmente se recupera la fase del frente de onda, para la prueba de Ronchi tenemos que integrar las aberraciones horizontal y vertical, la integración se la hacemos con la regla de Simpson y después se suman para recuperar el frente de onda.

Hacemos un ajuste con los polinomios de Zernike y finalmente hacemos una comparación cualitativa y cuantitativa con el software comercial Durango®.

Todo esto lo hacemos desde un software que hemos desarrollado en Visual C++ 6.0, en el cual están programados todos los métodos mencionados.

# *ABSTRACT*

This thesis is based on the implementation of 2 devices (one with stepper motors and the other with a liquid crystal display) to test convergent optical systems qualitatively and quantitatively using the Interferometry Lateral Phase Shift. Phase shifts are obtained in two ways: in the first we place a Ronchi grating on a mount that is adapted to a stepper motors from which is gave the desired phase shift and in the second method Ronchi gratings are displayed in a screen on a Liquid Crystal Display (LCD), the gratings that are displayed on the LCD are previously created in a program that we have developed. In both cases the fringe patterns are digitized with a CCD camera and analyzed using software that we have developed.

There are different algorithms PSI to take the fringe patterns, of which we have implemented eight, they differ in the number of patterns that are digitized and on the step size of the phase shift introduced.

The experimental fringe patterns are digitized with a CCD camera, when a fringe pattern is digitized the valid phase pixels are within a similar form to the surface under test in the digitized pattern. In the literature there are data modulation equations to find valid phase pixels, we develop some equations (which we call background equations), which serve to distinguish the pixels of valid phase from pixels of non-valid phase. Once we have valid phase pixels, we get a mask to which we apply the geometric moments to get the centroid and radius of valid phase in pixels of the digitized fringe patterns, background equations with geometric moments are used in the differentiation of the valid phase pixels from invalid phase validates, which serves is of great importance and help in the analysis of experimental fringe patterns obtaining good results.

After we get the wrapped phase of the fringe patterns, and then the phase is unwrapped, we have developed three different methods of many that exist in the literature.

Finally recovers the phase of the wavefront, for the Ronchi test we have to integrate horizontally and vertically, and add them to recover the wavefront.

We fit with Zernike polynomials and finally we compare with commercial software Durango ®.

We do this with software we have developed in Visual C++ 6.0, on which are programmed all the methods mentioned.

# *Agradecimientos*

*En primer lugar gracias a dios por darme vida, salud y fuerza para lograr mis metas.*

*Un agradecimiento especial a mi director de tesis al Dr. Jorge Castro Ramos por sus consejos, por su confianza y por su apoyo para terminar esta etapa de mi vida.*

*A mis sinodales por su tiempo en la revisión de esta tesis.*

*A todos mis compañeros y amigos en el INAOE que me brindaron su apoyo, consejos, sinceridad y por todos los momentos agradables que pasamos juntos.*

*También quiero agradecer a todas las personas que laboran en el INAOE, en especial a las personas que laboran en la Coordinación de Óptica, en Coordinación Docente por su apoyo y atención brindada en los trámites administrativos.*

*Por último quiero agradecer a CONACYT por haberme otorgado el apoyo económico y administrativo para terminar mis estudios de Doctorado.*

# *D e d i c a t o r i a*

*En primer lugar a mis adorados hijos Psiqué Mae Sánchez Muñoz y Évariste Sánchez Muñoz y a mi querida esposa Alys Mae Muñoz Flores quienes son el amor de mi vida, una fuente de constante inspiración y mi motivación para seguir superándome día a día.*

*A mis padres Jaime Sánchez Ortega y Cecilia Paredes Méndez.*

*A mis hermanos Yadira, Alejandra, Erick, Ivan y Joshua.*

*A todos mis familiares, amigos y personas con las que convivo por darme ánimos de seguir adelante.*

*A mi familia*



Jaime Sánchez Paredes

Agosto 2013

# ÍNDICE GENERAL

ÍNDICE GENERAL .....	10
ÍNDICE DE FIGURAS .....	13
ÍNDICE DE TABLAS .....	15
<b>1. INTRODUCCIÓN. ....</b>	<b>16</b>
<b>2. TEORÍA BÁSICA. ....</b>	<b>19</b>
2.1. INTRODUCCIÓN.....	19
2.2. INTERFERENCIA.....	19
2.2.1. <i>Interferencia de dos ondas monocromáticas.</i> .....	20
2.2.2. <i>Interferómetro de división del frente de onda.</i> .....	24
2.2.3. <i>Interferómetro de división de amplitud.</i> .....	24
2.2.3.1. Interferómetro de Newton. ....	24
2.2.3.2. Interferómetro Twyman-Green. ....	26
2.2.4. <i>Sistema del interferómetro Zygo.</i> .....	27
2.2.4.1. Capacidades del sistema Zygo. ....	27
2.2.4.2. Esquema. ....	28
2.3. INTERFEROMETRÍA DE DESPLAZAMIENTO DE FASE (PSI).....	29
2.3.1. <i>Ventajas de PSI.</i> .....	29
2.3.2. <i>Métodos de desplazamiento de fase.</i> .....	30
2.3.2.1. Desplazamiento de fase lateral.....	31
2.3.3. <i>Detectando la fase del frente de onda.</i> .....	33
2.3.4. <i>Algoritmos PSI.</i> .....	33
2.3.4.1. Técnica de detección de fase.....	34
2.3.4.2. Algoritmo de 3 pasos de 0, 90 y 180 grados. ....	37
2.3.4.3. Algoritmo de 3 pasos de 0, 120 y 240 grados. ....	37
2.3.4.4. Algoritmo de 4 pasos de 0, 90, 180 y 270 grados. ....	38
2.3.4.5. Algoritmo de 4 pasos de 0, 120, 240 y 360 grados. ....	38
2.3.4.6. Algoritmo de 5 pasos de 0, 45, 90, 135 y 180 grados. ....	38
2.3.4.7. Algoritmo Hariharan de 5 pasos de 0, 90, 180, 270 y 360 grados.....	38
2.3.4.8. Algoritmo de 6 pasos de 0, 90, 180, 240, 360 y 450 grados. ....	39
2.3.4.9. Algoritmo de 7 pasos de 0, 60, 120, 180, 240, 300 y 360 grados.....	40
2.4. PRUEBA DE RONCHI. ....	40
2.4.1. <i>Teoría geométrica.</i> .....	40
2.5. MOMENTOS GEOMÉTRICOS.....	43
2.5.1. <i>Definición y propiedades.</i> .....	43
2.5.2. <i>Representación de forma usando momentos geométricos.</i> .....	44
2.6. DESARROLLO DE FASE.....	46
2.6.1. <i>Desarrollo de fase, en 2-D por Tomas Kreis.</i> .....	48
2.6.2. <i>Desarrollo de fase, de mapas de fase. consistente de campo completo</i> .....	49

2.6.3.	<i>Desenvolvimiento de fase, de mapas de fase consistente dentro de una región conectada simple.</i>	51
2.6.4.	<i>Método robusto para desenvolver la fase en dos dimensiones.</i>	52
2.7.	POLINOMIOS DE ZERNIKE	57
2.8.	CONCLUSIÓN	58
<b>3.</b>	<b>ECUACIONES DE FONDO</b>	<b>59</b>
3.1.	INTRODUCCIÓN	59
3.2.	ECUACIÓN DE MODULACIÓN DE SCHREIBER	60
3.2.1.	<i>Ecuación de modulación para algoritmo PSI de 3 pasos de 0, 90 y 180 grados.</i>	60
3.2.2.	<i>Ecuación de modulación para algoritmo PSI de 3 pasos de 0, 120 y 240 grados.</i>	61
3.2.3.	<i>Ecuación de modulación para algoritmo PSI de 4 pasos de 0, 90, 180 y 270 grados.</i>	61
3.2.4.	<i>Ecuación de modulación para algoritmo de 5 pasos de Hariharan 0, 90, 180, 270 y 360 grados.</i>	61
3.3.	MÉTODO PARA LA OBTENCIÓN DE LA ECUACIÓN DE FONDO	61
3.4.	ECUACIONES DE FONDO PARA VARIOS ALGORITMOS PSI	63
3.4.1.	<i>Ecuación de fondo para algoritmo PSI de 3 pasos de 0, 90 y 180 grados.</i>	63
3.4.2.	<i>Ecuación de fondo para algoritmo PSI de 3 pasos de 0, 120 y 240 grados.</i>	64
3.4.3.	<i>Ecuación de fondo para algoritmo PSI de 4 pasos de 0, 90, 180 y 270 grados.</i>	64
3.4.4.	<i>Ecuación de fondo para algoritmo PSI de 4 pasos de 0, 120, 240 y 360 grados.</i>	64
3.4.5.	<i>Ecuación de fondo para algoritmo PSI de 5 pasos 0, 45, 90, 135 y 180 grados.</i>	64
3.4.6.	<i>Ecuación de fondo para algoritmo PSI Hariharan de 5 pasos de 0, 90, 180, 270 y 360.</i>	64
3.4.7.	<i>Ecuación de fondo para algoritmo PSI de 6 pasos de 0, 90, 180, 240, 360 y 450 grados.</i>	65
3.4.8.	<i>Ecuación de fondo para algoritmo PSI de 7 pasos de 0, 60, 120, 180, 240, 300 y 360 grados.</i>	65
3.5.	CONCLUSIÓN	65
<b>4.</b>	<b>PANTALLA DE CRISTAL LÍQUIDO (LCD)</b>	<b>66</b>
4.1.	INTRODUCCIÓN	66
4.2.	CARACTERÍSTICAS DE LA LCD	66
4.3.	PRUEBAS IMPLEMENTADAS CON LA LCD	69
4.4.	RESOLUCIÓN DE LA LCD	69
4.4.1.	<i>Resolución de la LCD para pruebas con PSI.</i>	70
4.5.	CONCLUSIÓN	72
<b>5.</b>	<b>ARREGLOS EXPERIMENTALES</b>	<b>73</b>
5.1.	INTRODUCCIÓN	73
5.2.	PRUEBA DE RONCHI CON MOTORES A PASOS	73
5.2.1.	<i>Fuente de poder para la fuente de luz.</i>	74
5.3.	ARREGLO EXPERIMENTAL CON LA LCD	76
5.4.	PRUEBA DE FOUCAULT	77
5.5.	PRUEBA DEL ALAMBRE	79
5.6.	PRUEBA DE HARTMANN	80
5.7.	PRUEBA DE RONCHI CON REJILLA BIRONCHI, CIRCULAR Y NULA	80
5.8.	CONCLUSIÓN	81

<b>6.</b>	<b>SOFTWARE DESARROLLADO.</b>	<b>82</b>
6.1.	INTRODUCCIÓN.	82
6.2.	SOFTWARE PARA MOVER LOS MOTORES A PASOS.	82
6.3.	SOFTWARE PARA DESPLEGAR REJILLAS EN LA LCD.	83
6.4.	SOFTWARE PARA ANALIZAR INTERFEROGRAMAS Y RONCHIGRAMAS.	98
6.4.1.	<i>Selección de tipo de patrones de franjas a analizar, algoritmo PSI y método de desenvolvimiento.</i>	98
6.4.2.	<i>Abrir patrones de franjas a analizar.</i>	99
6.4.3.	<i>Cálculo de las ecuaciones de fondo.</i>	100
6.4.4.	<i>Máscara de los píxeles de fase valida.</i>	102
6.4.5.	<i>Ecuación de modulación de datos y momentos geométricos.</i>	104
6.4.6.	<i>Fase envuelta, traslación de la fase y desenvolvimiento de la fase.</i>	106
6.5.	CONCLUSIÓN.	115
<b>7.</b>	<b>RESULTADOS.</b>	<b>116</b>
7.1.	INTRODUCCIÓN.	116
7.2.	PRUEBA DE RONCHI CON LOS MOTORES A PASOS.	116
7.2.1.	<i>Comparación entre Zemax y nuestros resultados.</i>	118
7.3.	PRUEBA DE RONCHI CON LCD.	119
7.4.	COMPARACIÓN DE LOS RESULTADOS CON DURANGO®.	123
7.5.	ERROR PICO-VALLE, RMS Y AJUSTE CON LOS POLINOMIOS DE ZERNIKE.	126
7.6.	CONCLUSIÓN.	128
<b>8.</b>	<b>CONCLUSIONES.</b>	<b>130</b>
	<b>REFERENCIAS.</b>	<b>132</b>

# ÍNDICE DE FIGURAS

FIG. 2.1. VARIACIÓN DE INTENSIDAD CON LA DIFERENCIA DE FASE. ....	23
FIG. 2.2. EXPERIMENTO DE YOUNG. ....	24
FIG. 2.3. FRANJAS DE INTERFERENCIA DE YOUNG. ....	24
FIG. 2.4. FORMACIÓN DE FRANJAS RECTAS ENTRE DOS SUPERFICIES ÓPTICAS PLANAS. ....	25
FIG. 2.5. INTERFERÓMETRO DE MICHELSON CON LUZ COLIMADA. ....	27
FIG. 2.6. A) SISTEMA DEL INTERFERÓMETRO ZYGO, B) ESQUEMA USADO PARA PROBAR ESPEJOS CÓNCAVOS. ....	28
FIG. 2.7. FRENTE DE ONDA ORIGINAL Y DESPLAZADO DE UNA APERTURA CIRCULAR. ....	32
FIG. 2.8. INTERFERÓMETRO TWYMAN-GREEN CON EL PIEZOELECTRICO EN EL BRAZO DE REFERENCIA PARA CONTROLAR LA LONGITUD DE LA TRAYECTORIA DE CAMINO ÓPTICO. ....	35
FIG. 2.9. DIAGRAMA DE LA PRUEBA DE RONCHI. ....	41
FIG. 2.10. UNA IMAGEN Y LA ELIPSE CORRESPONDIENTE DE LA IMAGEN. ....	46
FIG. 2.11 A) CONVERSIÓN DE LA FASE CALCULADA POR EL ARCOTANGENTE, A EL MÓDULO $2\pi$ DE LA FASE DEL FRENTE DE ONDA; B) PROCESO DE DESENVOLVIMIENTO DE FASE EN UNA DIMENSIÓN. ...	48
FIG. 2.12. A), B) Y C) TRAYECTORIAS PARA DESENVOLVER LA FASE EN DOS DIMENSIONES. ....	49
FIG. 2.13. TRAYECTORIA QUE SE SIGUE PARA DESENVOLVER LA FASE EN DOS DIMENSIONES. ....	50
FIG. 2.14 EJEMPLO DE UNA REGIÓN CONECTA SIMPLE QUE CONTIENE DATOS DE FASE VALIDOS. ....	51
FIG. 3.1. COMPUTANDO AUTOMÁTICAMENTE EL VALOR DE UMBRAL CON AJUSTE POLINOMIAL DE TERCER ORDEN. ....	63
FIG. 5.1. ARREGLOS EXPERIMENTALES DE LA PRUEBA DE RONCHI PARA PROBAR: A) LENTES, B) ESPEJOS. .....	74
FIG. 5.2. PICOS ESPECTRALES DE LOS LEDs USADOS COMO FUENTE DE ILUMINACIÓN. ....	74
FIG. 5.3. LAYOUT DE LA FUENTE DE PODER. ....	76
FIG. 5.4. ARREGLO EXPERIMENTAL DE LA PRUEBA DE RONCHI CON LA LCD. ....	76
FIG. 5.5. DIAGRAMA DEL DISPOSITIVO IMPLEMENTADO CON LA LCD. ....	77
FIG. 5.6. POSICIÓN DEL FILO DE NAVAJA PROYECTADO EN EL PLANO DE LA PUPILA DE ENTRADA. ....	79
FIG. 6.1. VENTANA PRINCIPAL DEL SOFTWARE USADO PARA MOVER LOS MOTORES A PASOS. ....	82
FIG. 6.2. VENTANA PRINCIPAL DEL SOFTWARE USADO PARA LA CREACIÓN DE REJILLAS. ....	83
FIG. 6.3. INGRESO DE DATOS PARA LA CREACIÓN DE LA REJILLA DE LA PRUEBA DE LA NAVAJA. ....	83
FIG. 6.4. A, B, C Y D) FILOS DE NAVAJA VERTICALES DESPLEGADOS EN LA LCD CON POSICIONES Y ANCHOS VARIABLES. ....	84
FIG. 6.5. A, B, C Y D) FILOS DE NAVAJA DESPLEGADOS EN LA LCD CON POSICIONES Y ANCHOS VARIABLES HORIZONTALMENTE. ....	85
FIG. 6.6. INGRESO DE DATOS PARA DOS CÍRCULOS. ....	86
FIG. 6.7. A, B) CÍRCULOS DESPLEGADOS EN LA LCD CON DIÁMETRO Y POSICIÓN VARIABLE. ....	87
FIG. 6.8. INGRESO DE DATOS PARA LA PRUEBA DEL ALAMBRE. ....	87
FIG. 6.9. A, B) ALAMBRES HORIZONTALES DESPLEGADOS EN LA LCD. ....	88
FIG. 6.10. ALAMBRES VERTICALES DESPLEGADOS EN LA LCD. ....	88
FIG. 6.11. ALAMBRES INCLINADOS DESPLEGADOS EN LA LCD. ....	89
FIG. 6.12. SELECCIÓN DE ALGORITMO PSI, E INGRESO DE ANCHO DE LÍNEA EN PÍXELES. ....	90
FIG. 6.13. BOTONES PARA DESPLEGAR LAS REJILLAS SELECCIONADAS. ....	90
FIG. 6.14. REJILLAS DE RONCHI HORIZONTALES CON DESPLAZAMIENTOS DE FASE DE 90 GRADOS. ....	92
FIG. 6.15. REJILLAS DE RONCHI HORIZONTALES CON DESPLAZAMIENTOS DE FASE DE 120 GRADOS. ....	93
FIG. 6.16. REJILLAS DE RONCHI INCLINADAS CON DESPLAZAMIENTOS DE FASE DE 45 GRADOS. ....	94
FIG. 6.17. REJILLAS BIRONCHI CON UN DESPLAZAMIENTO DE FASE DE 120 PÍXELES POR FRANJA. ....	96
FIG. 6.18. INGRESO DE DATOS DE REJILLA CIRCULAR. ....	96
FIG. 6.19. REJILLA CIRCULARES CON DESPLAZAMIENTOS DE FASE DE 0, 90, 180, 240, 360 Y 450 GRADOS .....	97

FIG. 6.20. VENTANA PRINCIPAL DEL SOFTWARE DESARROLLADO PARA EL ANÁLISIS DE PATRONES DE FRANJAS. ....	98
FIG. 6.21. SELECCIÓN DE TIPO DE FRANJAS A ANALIZAR, MÉTODO PSI Y MÉTODO DE DESENVOLVIMIENTO. ....	99
FIG. 6.22. LECTURA DE IMÁGENES FONDO DE LAS IMÁGENES Y UMBRAL. ....	99
FIG. 6.23. FONDO COMBINADO DE UNA SERIE DE RONCHIGRAMAS. ....	102
FIG. 6.24. MÁSCARA DE LOS PÍXELES DE FASE VÁLIDA PARA UNA SERIE DE RONCHIGRAMAS. ....	104
FIG. 6.25. MÁSCARA DE PÍXELES DE FASE VALIDA OBTENIDOS CON LA ECUACIÓN DE MODULACIÓN ..	104
FIG. 6.26. CENTROIDE Y RADIO DE LA MÁSCARA DE LOS PÍXELES DE FASE VALIDA. ....	106
FIG. 6.27. FASE ENVUELTA. ....	108
FIG. 6.28. TRASLACIONES DE LA FASE. ....	109
FIG. 6.29. DESENVOLVIMIENTO DE LA FASE. ....	115
FIG. 7.1. RONCHIGRAMAS VERTICALES. ....	117
FIG. 7.2. RONCHIGRAMAS HORIZONTALES. ....	117
FIG. 7.3. ABERRACIÓN TRANSVERSAL Y ABERRACIÓN DEL FRENTE DE ONDA. ....	118
FIG. 7.4. GRÁFICAS PARA ABERRACIÓN TRANSVERSAL EN ZEMAX Y ABERRACIÓN TRANSVERSAL CON EL SOFTWARE DESARROLLADO. ....	119
FIG. 7.5. FRANJAS OBTENIDAS DESPLEGANDO DIFERENTES REJILLAS EN LA LCD. ....	121
FIG. 7.6. RONCHIGRAMAS VERTICALES CON DESPLAZAMIENTO DE FASE DE 90°. ....	121
FIG. 7.7. MÁSCARAS DE LOS PÍXELES DE FASE VALIDA OBTENIDAS CON LAS ECUACIONES DE MODULACIÓN Y FONDO APLICÁNDOLES LOS MOMENTOS GEOMÉTRICOS. ....	122
FIG. 7.8. ABERRACIONES TRANSVERSALES HORIZONTAL Y VERTICAL; Y FRENTE DE ONDA. ....	122
FIG. 7.9. INTERFEROGRAMAS DIGITALIZADOS CON EL PROGRAMA DURANGO® DESDE EL ZYGO. ....	123
FIG. 7.10. A) VALOR DE LA ECUACIÓN DE FONDO, B) MÁSCARA DE LOS PÍXELES DE FASE VÁLIDA Y C) INTENSIDAD DE LA IMAGEN CON DURANGO®. ....	124
FIG. 7.11 ECUACIÓN DE MODULACIÓN A) CON NUESTRO PROGRAMA Y B) CON DURANGO®. ....	124
FIG. 7.12. CÍRCULO EN EL CONTORNO DE LA FASE VALIDA A) A PARTIR DE LA ECUACIÓN DE FONDO Y B) A PARTIR DE LA ECUACIÓN DE MODULACIÓN. ....	125
FIG. 7.13. FASE ENVUELTA A) CALCULADA CON DURANGO® ANTES DE SELECCIONAR LA APERTURA B) CALCULADA CON DURANGO® ANTES DE SELECCIONAR LA APERTURA Y C) CON NUESTRO PROGRAMA. ....	125
FIG. 7.14. FRENTE DE ONDA A) CON NUESTRO PROGRAMA Y B) CON DURANGO®. ....	126
FIG. 7.15. FRENTE DE ONDA OBTENIDO A PARTIR DE LOS COEFICIENTES AJUSTADOS DE ZERNIKE. ....	127

# *ÍNDICE DE TABLAS*

TABLA 2.1. CORRECCIÓN DEL MÓDULO $2\pi$ DE LA FASE .....	47
TABLA 2.2. POLINOMIOS DE ZERNIKE. ....	57
TABLA 6.1. EQUIVALENCIA DE LOS PERIODOS DE LAS REJILLAS EN ANCHOS DE LÍNEA EN PÍXELES .....	90
TABLA 7.1. VALORES DE LOS COEFICIENTES DE ZERNIKE DEL ESPEJO BAJO PRUEBA. ....	118
TABLA 7.2. COEFICIENTES DE ZERNIKE AJUSTADOS .....	126
TABLA 7.3. COMPARACIÓN DE LOS ERRORES PICO-VALLE Y RMS.....	128

# *Capítulo 1*

## *1. Introducción.*

Esta tesis se basa en la implementación de 2 dispositivos (uno con motores a pasos [1] y el otro con una pantalla de cristal líquido [2]), ambos usados para probar sistemas ópticos convergentes de manera cualitativa y cuantitativa usando la técnica de Desplazamiento de Fase Lateral.

Desde que Vasco Ronchi desarrolló la prueba de Ronchi [3-5], se ha estudiado y mejorado por varios autores por un largo tiempo. Nuñez y Cordero [6] propusieron utilizar un LED como fuente de luz en la prueba de Ronchi. La pantalla LCD se ha utilizado como rejilla de Ronchi en la prueba de Ronchi usando un láser como fuente de iluminación [7-10] y se ha desarrollado por diferentes autores pero han obtenido una pobre resolución y visibilidad en sus pruebas. Omura y Yatagai [11], Sánchez & Silva [1] publicaron un método para probar lentes, que utiliza un motor a pasos con una interfaz de PC para mover una rejilla de Ronchi.

Usando varios algoritmos PSI y el algoritmo Hariharan [12] se calcula la fase envuelta del frente de onda.

Con el uso de un LED como fuente de iluminación se evita el uso de láseres costosos o el uso de una lámpara de tungsteno como fuente de iluminación en la prueba de superficies con pulido convencional. Con la configuración usada no se necesitan elementos ópticos adicionales para observar los patrones de Ronchi como se ha mostrado en algunas configuraciones experimentales [4].

En este trabajo pusimos en práctica un conjunto de pruebas ópticas de desplazamiento lateral que aún no se han implementado en una pantalla de



cristal líquido (LCD), a pesar de que muchos años han pasado estas pruebas todavía aún son muy utilizadas cuando alguien intenta probar telescopios, objetivos de microscopio y otros instrumentos ópticos. Con el uso de un LCD como pantalla evitamos comprar un sistema costoso de rejillas de Ronchi, también, evitamos emplear mucho tiempo la construcción de pantallas de Hartmann. Además de que elaborar rejillas Ronchi nulas siempre ha sido un problema, con el software desarrollado estas se podrían diseñar y desplegar en la LCD de forma casi instantánea. Además, no sólo damos los resultados cualitativos incluso se muestra que con el software desarrollado es posible obtener resultados cuantitativos. Se muestra la implementación de la prueba de Hartmann y la prueba bironchi, evitamos utilizar costosas monturas de precisión para girar las rejillas de Ronchi, también se implementó la técnica de desplazamiento de fase sin el uso de motores paso a paso o un dispositivo piezoeléctrico u otro, evitando los errores de desplazamiento de fase como Cheng & Wyant lo compensaron [13].

Existen diferentes algoritmos de PSI para tomar los patrones de franjas, de los cuales hemos implementado 8 [14,15], estos difieren en el número de patrones de franjas que son digitalizados y en el tamaño de paso del desplazamiento de fase introducido.

Los patrones de franjas experimentales son digitalizados con una cámara CCD, cuando se digitaliza un patrón de franjas la fase válida queda dentro de una forma similar a la superficie o sistema óptico bajo prueba en el patrón digitalizado. En la literatura existen unas ecuaciones de modulación [15] de datos para encontrar los píxeles de fase válida, nosotros encontramos unas ecuaciones propias (a las que llamamos ecuaciones de fondo), que nos sirven para diferenciar los píxeles de fase válida de los de fase no válida. Una vez que tenemos los píxeles de fase válida obtenemos una máscara a la cual le aplicamos los momentos geométricos [3,16] para obtener el centroide y el radio en píxeles de los patrones de franjas digitalizados, las ecuaciones

de fondo junto con los momentos geométricos nos sirven en el análisis de los patrones de franjas obtenidos experimentalmente obteniendo muy buenos resultados.

Los métodos de Kreis [17] y Ghiglia [18] se utilizaron para desenvolver la fase de la aberración de frente de onda. También se explica la resolución de las rejillas en la pantalla LCD y, finalmente, los resultados experimentales se compararon con el software de Durango® [19] para demostrar la validez de nuestras pruebas aplicadas en la pantalla LCD, se dan finalmente conclusiones.

Todo esto lo hacemos desde software que hemos desarrollado en Visual C++ 6.0, en el cual están programados todos los métodos mencionados.

## *Capítulo 2*

### *2. Teoría básica.*

#### *2.1. Introducción.*

En este capítulo se explica el fenómeno físico de interferencia, se explica en que consiste la Interferometría de desplazamiento de fase PSI, se dan los principios básicos de la prueba de Ronchi, se explican los momentos geométricos, se explican las técnicas de desenvolvimiento de fase, se describe brevemente el Interferometro Zygo y se describen brevemente los polinomios de Zernike.

#### *2.2. Interferencia.*

El fenómeno de interferencia [20,21] ha sido usado para establecer la naturaleza ondulatoria de la luz y ahora tiene usos de importancia práctica por ejemplo en espectroscopia y metrología. Cuando dos a más haces de luz se superponen, la distribución de intensidad no puede ser descrita de una manera sencilla. Así, si la luz de una fuente es dividida en dos haces los cuales son superpuestos, la intensidad en la región de superposición es una variación punto a punto entre un valor máximo que puede ser igual a la suma de las intensidades y un mínimo que puede ser cero, a este fenómeno se le llama interferencia. Si los dos haces son originados en la misma fuente las fluctuaciones en los dos haces están en general correlacionadas y se dice que los haces son completa o parcialmente coherentes. En haces de

diferentes fuentes, las fluctuaciones no están correlacionadas y se dice que los haces son mutuamente incoherentes. Cuando los haces de diferentes fuentes son superpuestos, no se observa interferencia.

Existen dos métodos para obtener interferencia de un solo haz de luz y estos proveen una base para clasificar los arreglos usados para producir interferencia. En un método el haz se divide al pasar a través de aberturas colocadas lado a lado en una pantalla opaca. Este método es llamado división de frente de onda, es de uso únicamente con fuentes suficientemente pequeñas. De manera alternativa si el haz se divide en una o más superficies parcialmente reflectoras, en las cuales una parte de la luz es reflejada y otra parte transmitida. Este método es llamado división de amplitud, este método se puede usar con fuentes extendidas y por lo tanto los efectos pueden ser de mayor intensidad que con el método de división de frente de onda.

### 2.2.1. Interferencia de dos ondas monocromáticas.

La intensidad  $I$  de la luz se define como el promedio temporal de la cantidad de energía que atraviesa en una unidad de tiempo una unidad de área perpendicular a la dirección del flujo de energía [20,21]. Para una onda plana

$$I = \frac{c}{4\pi} \sqrt{\frac{\epsilon}{\mu}} \langle E^2 \rangle \quad (2.1)$$

donde  $\epsilon$  es conocida como la constante dieléctrica (o permitividad),  $\mu$  es la permeabilidad magnética,  $E$  es el vector del campo eléctrico y  $c$  es la velocidad de la luz en el vacío. Si se trabaja principalmente con campos monocromáticos se representa el vector del campo eléctrico  $E$  en la forma

$$E(r, t) = \Re \{ A(r) e^{-i\omega t} \} = \frac{1}{2} [ A(r) e^{-i\omega t} + A^*(r) e^{i\omega t} ] \quad (2.2)$$

donde  $\omega$  es la frecuencia angular temporal,  $i = \sqrt{-1}$ ,  $t$  es el tiempo y  $\Re$  es la parte real del número complejo,  $e$  es la función exponencial y  $*$  indica que es el complejo conjugado de  $A$ . Aquí  $A$  es un vector complejo con componentes cartesianas rectangulares

$$A_x = a_1(r)e^{ig_1(r)}, \quad A_y = a_2(r)e^{ig_2(r)}, \quad A_z = a_3(r)e^{ig_3(r)} \quad (2.3)$$

donde  $a_j$  y  $g_j$  ( $j=1,2,3$ ) son funciones reales. Para una onda plana homogénea las amplitudes  $a_j$  son constantes, mientras que las funciones de fase son de la forma  $g_j(r) = \mathbf{K} \cdot \mathbf{r} - \delta_j$ , donde  $\mathbf{K}$  es el vector de propagación y las  $\delta_j$  son las constantes de fase las cuales especifican el estado de la polarización. De la ec. 2.2,

$$E^2 = \frac{1}{4} (A^2 e^{-2i\omega t} + A^{*2} e^{-2i\omega t} + 2A \cdot A^*) \quad (2.4)$$

de donde, tomando el promedio temporal sobre un intervalo grande comparado con el periodo  $T = 2\pi/\omega$

$$\langle E^2 \rangle = \frac{1}{2} A \cdot A^* = \frac{1}{2} (|A_x|^2 + |A_y|^2 + |A_z|^2) = \frac{1}{2} (a_1^2 + a_2^2 + a_3^2) \quad (2.5)$$

Suponiendo que dos ondas monocromáticas  $E_1$  y  $E_2$  se superponen en un punto  $P$ . El campo eléctrico total en  $P$  es

$$\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2 \quad (2.6)$$

de modo que

$$E^2 = E_1^2 + E_2^2 + 2E_1 \cdot E_2. \quad (2.7)$$

Por lo tanto la intensidad total en  $P$  es

$$\mathbf{I} = \mathbf{I}_1 + \mathbf{I}_2 + \mathbf{I}_{12} \quad (2.8)$$

donde

$$I_1 = \langle E_1^2 \rangle, \quad I_2 = \langle E_2^2 \rangle \quad (2.9)$$

son las intensidades de las dos ondas,

$$I_{12} = 2\langle E_1 \cdot E_2 \rangle \quad (2.10)$$

es el termino de interferencia. Si  $A$  y  $B$  son las amplitudes complejas de las ondas donde

$$A_x = a_1 e^{ig_1}, \dots \quad B_x = b_1 e^{ih_1}, \dots \quad (2.11)$$

Las fases  $g_j$  y  $h_j$  de las dos ondas por lo general serán diferentes, desde que las ondas hayan llegado hasta  $P$  por diferentes trayectorias, pero si las condiciones experimentales son tales que se introduce la misma diferencia de fase  $\delta$  entre las componentes correspondientes, tenemos

$$g_1 - h_1 = g_2 - h_2 = g_3 - h_3 = \delta = \frac{2\pi}{\lambda_0} \Delta S, \quad (2.12)$$

donde  $\Delta S$  es la diferencia entre las trayectorias ópticas de las dos ondas desde el punto común de la fuente hasta  $P$  y  $\lambda_0$  es la longitud de onda en el vacío. En términos de  $A$  y  $B$ ,

$$E_1 \cdot E_2 = \frac{1}{4} (Ae^{-i\omega t} + A^* e^{i\omega t}) \cdot (Be^{-i\omega t} + B^* e^{i\omega t}) = \frac{1}{4} (A \cdot B e^{-2i\omega t} + A^* \cdot B^* e^{2i\omega t} + A \cdot B^* + A^* \cdot B) \quad (2.13)$$

de modo que

$$I_{12} = 2\langle E_1 \cdot E_2 \rangle = \frac{1}{2} (A \cdot B^* + A^* \cdot B) = a_1 b_1 \cos(g_1 - h_1) + a_2 b_2 \cos(g_2 - h_2) + a_3 b_3 \cos(g_3 - h_3) \\ I_{12} = (a_1 b_1 + a_2 b_2 + a_3 b_3) \cos \delta. \quad (2.14)$$

donde  $\cos$  es la función trigonométrica coseno.

Esta expresión muestra la dependencia del término de interferencia en las componentes de amplitud y en la diferencia de fase de las dos ondas. Fresnel de Arago encontró que dos haces de luz polarizados a ciertos ángulos uno con respecto del otro, no interfieren de lo que se concluye que las vibraciones deben ser transversales. Considerando la distribución de intensidad que resulta de superponer dos ondas que se encuentran

propagando en la dirección  $z$  y están linealmente polarizados sus vectores  $E$  en la dirección  $x$ . Entonces

$$a_2 = a_3 = b_2 = b_3 = 0 \quad (2.15)$$

usando las ecs. 2.5, 2.9, 2.10 y 2.14,

$$I_1 = \frac{1}{2}a_1^2, \quad I_2 = \frac{1}{2}a_2^2 \quad (2.16)$$

$$I_{12} = a_1 b_1 \cos \delta = 2\sqrt{I_1 I_2} \cos \delta \quad (2.17)$$

La intensidad total es dada por la ec. 2.8 como

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos \delta \quad (2.18)$$

Los máximos de intensidad se darán cuando

$$I_{MAX} = I_1 + I_2 + 2\sqrt{I_1 I_2} \quad (2.19)$$

$$|\delta| = 0, 2\pi, 4\pi, \dots,$$

y los mínimos de intensidad cuando

$$I_{MIN} = I_1 + I_2 - 2\sqrt{I_1 I_2} \quad (2.20)$$

$$|\delta| = \pi, 3\pi, \dots,$$

Cuando  $I_1 = I_2$ , ec. 2.18 se reduce a

$$I = 2I_1(1 + \cos \delta) = 4I_1 \cos^2 \frac{\delta}{2} \quad (2.21)$$

y el valor de la intensidad varía entre un valor máximo  $I_{MAX} = 4I_1$ , y un valor mínimo  $I_{MIN} = 0$  (Fig. 2.1.).

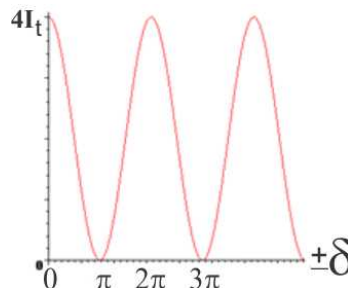


Fig. 2.1. Variación de intensidad con la diferencia de fase.

### 2.2.2. Interferómetro de división del frente de onda.

El Interferómetro de Young [20,21] se usó para demostrar la interferencia de la luz. La luz que proviene de una fuente puntual monocromática  $S$  pasa a través de dos aberturas pequeñas  $S_1$  y  $S_2$  las cuales están relativamente cerca una de la otra, en una pantalla  $A$  y equidistantes con respecto a  $S$  (Fig. 2.2.). Las dos aberturas actúan como fuentes puntuales secundarias monocromáticas las cuales están en fase y los haces de las aberturas se superponen lejos de la pantalla  $A$ . En esta región es donde se forma el patrón de interferencia.

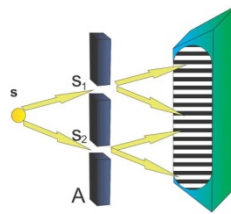


Fig. 2.2. Experimento de Young.

El patrón de interferencia en el plano *lejos de la pantalla A*, consiste de bandas brillantes y oscuras llamadas franjas de interferencia como se ilustra en la Fig. 2.3.



Fig. 2.3. Franjas de interferencia de Young.

### 2.2.3. Interferómetro de división de amplitud.

#### 2.2.3.1. Interferómetro de Newton.

Se puede decir que un interferómetro de Newton [22] es cualquier arreglo de dos superficies en contacto, que se encuentran iluminadas por una fuente de luz monocromática.



Cuando dos superficies ópticas perfectamente planas se ponen en contacto una con otra formando una cuña de aire como se ilustra en la Fig. 2.4. La cuña de aire es delgada a la izquierda y gruesa a la derecha. El espesor de la abertura de aire incluso del lado angosto no será cero, tendrá un valor finito y aunque uno presione muy fuerte, es difícil hacer esta abertura de aire cero. Imaginemos dos planos proyectados hacia atrás como en la Fig. 2.4 y conociendo la línea de intersección.

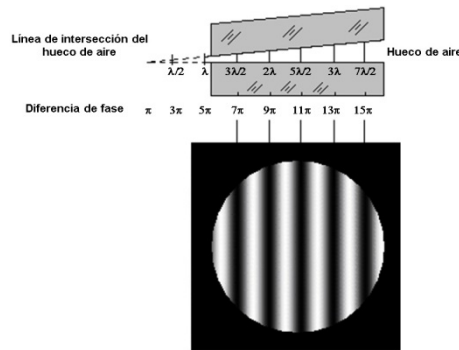


Fig. 2.4. Formación de franjas rectas entre dos superficies ópticas planas.

Si tenemos una fuente de luz monocromática de longitud de onda  $\lambda$  que incide normalmente en una superficie óptica plana, si  $\alpha$  es el ángulo de la cuña entre las dos superficies y la abertura de aire a una distancia  $x$  de la línea de intersección de dos planos es  $\alpha x$ . A esto le debemos sumar  $\lambda/2$  debido al cambio de fase de  $\pi$  resultado de la reflexión en la superficie plana. Así tenemos una diferencia de camino óptico (OPD) en la distancia  $x$  dada por  $(2\alpha x + \lambda/2)$ .

Entonces las franjas oscuras pueden ser representadas por

$$2\alpha x = n\lambda \quad (2.22)$$

donde  $n$  es un entero y la franja brillante puede ser representada por

$$2\alpha x + \lambda/2 = n\lambda \quad (2.23)$$

Cada una de estas ecuaciones representa un sistema de franjas rectas igualmente espaciadas y la distancia  $d$  entre dos franjas oscuras o brillantes es

$$d = \lambda/2\alpha \quad (2.24)$$

### 2.2.3.2. Interferómetro Twyman-Green.

Si el interferómetro de Michelson [20,21] es iluminado por una fuente de luz monocromática puntual  $S$  en el foco de una lente bien corregida  $L_1$  y la luz emergente del interferómetro es colimada por una segunda lente bien corregida  $L_2$ , el arreglo es equivalente al interferómetro de Fizeau pero los haces tienen trayectorias claramente separadas Fig. 2.5. Si  $w_1$  es un frente de onda plano del haz que regresa de  $M_1$ ,  $w_2$  es el frente de onda correspondiente al haz que regresa de  $M_2$ ; y  $w_1'$  es el frente de onda plano virtual que regresa de  $M_2$  el cual emergerá del divisor de haz coincidente y en fase con  $w_1$ . La diferencia de camino óptico entre los rayos emergentes los cuales se interceptan virtualmente en el punto  $P$  en  $w_2$  es

$$\Delta S = nh \quad (2.25)$$

donde  $h = PN$  es la distancia normal de  $w_1'$  a  $P$  y  $n$  es el índice de refracción del medio entre  $w_1'$  y  $w_2$ . La diferencia de fase correspondiente es

$$\delta = \frac{2\pi}{\lambda_0} nh \quad (2.26)$$

y un ocular colocado en el plano focal de  $L_2$  y enfocado en  $w_2$  veremos en  $P$  una franja brillante si

$$nh = m\lambda_0, \quad |m| = 0,1,2,\dots \quad (2.27)$$

y una franja oscura si

$$nh = m\lambda_0, \quad |m| = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots \quad (2.28)$$

Debido al monocromatismo de la fuente, las franjas únicamente son visibles si las trayectorias ópticas en los brazos son casi iguales. Esta modificación del interferómetro de Michelson fue introducida por Twyman y Green [20] para hacer pruebas de elementos ópticos.

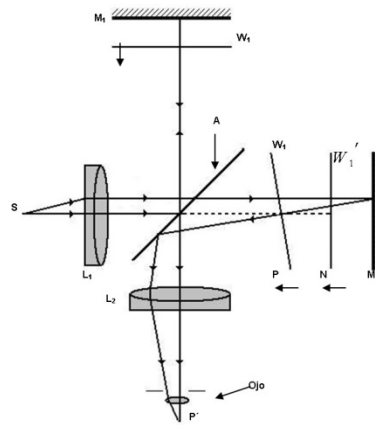


Fig. 2.5. Interferómetro de Michelson con luz colimada.

#### 2.2.4. Sistema del interferómetro Zygo.

El sistema del interferómetro Zygo [23] está diseñado para ser usado en una gran variedad de aplicaciones, debido a la viabilidad de la arquitectura del sistema sobre la cual se basa.

##### 2.2.4.1. Capacidades del sistema Zygo.

Las principales capacidades de un sistema del interferómetro Zygo son:

1. Medir la distorsión de la superficie de:
  - Superficies planas
  - Superficies esféricas convexas
  - Superficies esféricas cóncavas
  - Superficies asféricas
2. Medir la calidad del frente de onda transmitido de:
  - Vidrios
  - Prismas
  - Lentes
  - Sistemas de lentes
  - Sistemas ópticos
3. Medir el radio de curvatura de
  - Superficies esféricas convexas
  - Superficies esféricas cóncavas

#### 2.2.4.2. Esquema.

Este sistema interferométrico usa un sistema de construir bloques ya que al parecer es la mejor forma práctica de trabajar con una gran diversidad de aplicaciones para las cuales el sistema ha sido diseñado. La Fig. 2.6 a) muestra una fotografía del Zygo que es el núcleo del sistema de medida. Tiene un haz láser colimado de 100mm de diámetro el cual emerge del lado derecho de la fotografía. Usando los accesorios externos apropiados, se puede lograr ensamblar varias configuraciones. Por ejemplo un interferómetro de Fizeau, un interferómetro Twyman- Green o un arreglo para probar espejos cóncavos como se ilustra en la Fig. 2.6 b). Este sistema ha sido diseñado para tener una gran versatilidad simplemente con colocar accesorios externos. Este sistema solo sirve para observar las franjas de interferencia de la superficie que se desea probar, la digitalización de las imágenes y el proceso de análisis se realiza a través del programa Durango® diseñado y distribuido por la compañía Diffraction International [19], dicho programa hace el análisis dinámico de varios interferogramas y hace el ajuste a los polinomios de Zernike del frente de onda de la superficie probada.

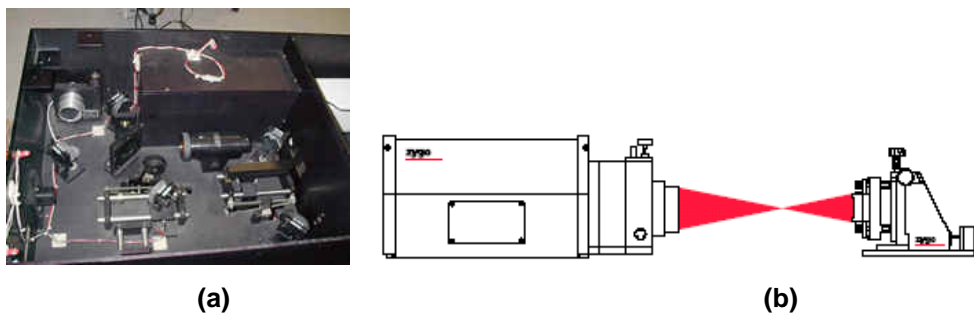


Fig. 2.6. a) Sistema del interferómetro Zygo, b) Esquema usado para probar espejos cóncavos.

## ***2.3. Interferometría de desplazamiento de fase***

### ***(PSI).***

La Interferometría de Desplazamiento de Fase (PSI) [15] no es una configuración óptica específica de hardware es más bien una forma de obtener datos y un método de análisis que puede ser aplicado a una gran variedad de situaciones de prueba.

Los análisis de interferogramas estáticos sufren la necesidad de encontrar los centros de las franjas. Los datos son colectados únicamente a lo largo de los centros de las franjas. Con un interferograma estático se requiere un poco de información adicional para determinar la polaridad del haz (saber si aumenta o disminuye la diferencia de camino óptico).

Electrónicamente con la técnica PSI se graba una serie de interferogramas cuando la fase de referencia del interferograma es cambiada. La fase del frente de onda está codificada en las variaciones en el patrón de intensidad de los interferogramas grabados y un simple cálculo punto a punto recupera la fase. La necesidad de localizar el centro de las franjas se elimina.

#### ***2.3.1. Ventajas de PSI.***

Tradicionalmente los interferogramas han sido analizados identificando el centro de las franjas y asignando una superficie de altura constante a lo largo de cada franja. Franjas adyacentes representan un cambio de altura de media onda. Encontrar el centro de las franjas ha sido un límite inherente en la precisión de esta técnica y también ha restringido la cantidad de datos procesados que se pueden hacer a los resultados. Visualmente la localización del centro de las franjas puede ser alrededor de un décimo o un veinteavo del espacio de una franja en condiciones ideales. Y es difícil localizar visualmente con precisión el máximo o el mínimo de intensidad del

patrón de franjas. Con un interferograma digitalizado por una cámara electrónica o un escáner de entrada esta situación se puede mejorar hasta un cincuentavo de una onda bajo condiciones ideales.

Debido a que el análisis requerido para PSI no requiere encontrar el centro de las franjas o la franja siguiente, cualquier tipo de patrón de franjas puede ser analizado. Un patrón de franjas sin franjas (una franja que cubre el campo entero de visión) o con una serie complicada de franjas cerradas es correctamente analizado. Se obtiene una mayor precisión cuando el número de franjas en el campo es minimizado. Por la misma razón los datos pueden ser tomados y analizados en cualquier arreglo de rejilla que se desee. La rejilla más común es un arreglo regular de píxeles en un sensor de estado sólido el cual es compatible con la mayoría de programas de procesamiento de datos. Otra ventaja de PSI es que es insensible a variaciones espaciales de intensidad, sensibilidad en el detector y patrones fijos de ruido.

La repetibilidad y precisión de PSI es mucho mayor que el análisis estático de un interferograma. Debido a que los datos son obtenidos sobre un intervalo de tiempo, se gana la ventaja en promediar secuencialmente los interferogramas grabados. La obtención de datos y el procedimiento de análisis para PSI son fundamentalmente diferentes de los usados para interferogramas estáticos y resulta en una gran mejora en repetibilidad para PSI. Repetibilidades de uno sobre cien de una longitud de onda se pueden obtener con PSI, y valores de uno sobre mil sobre una longitud de onda han sido reportados y pueden ser obtenidos con los equipos comerciales de estos días [24].

### **2.3.2. Métodos de desplazamiento de fase.**

El método más usado para introducir un desplazamiento de fase en un sistema PSI, consiste en trasladar uno de los espejos o superficies ópticas

en el interferómetro con un piezoeléctrico [25]. Estos dispositivos se expanden o contraen cuando se les aplica un voltaje externo. Dependiendo de la configuración se necesitan unos cuantos volts para obtener el movimiento deseado en longitudes de onda o movimientos menores.

Haciendo un cambio discreto en el voltaje aplicado, el desplazamiento de fase inducido varía en una serie de pasos. Si el voltaje se programa para variar suavemente, un desplazamiento de fase de una forma funcional deseada puede ser producido. La información de la prueba interferométrica puede ser usada para calcular el movimiento del piezoeléctrico. Los interferómetros más comunes para obtener desplazamiento de fase son: el Twyman-Green, el Mach-Zender y el Fizeau. Los primeros dos interferómetros trasladan uno de los espejos en el brazo de referencia para obtener el desplazamiento de fase. En incidencia normal en el interferómetro Twyman-Green se necesita un desplazamiento de media longitud de onda en el espejo para obtener un desplazamiento de fase de  $360^\circ$ .

#### **2.3.2.1. Desplazamiento de fase lateral.**

La interferometría de desplazamiento lateral es un área importante de la interferometría y ha sido usada extensamente en aplicaciones tales como la prueba de componentes y sistemas ópticos y el estudio del fenómeno de difusión y flujo en gases y líquidos. Básicamente el método de la interferometría de desplazamiento general consiste en duplicar el frente de onda bajo prueba desplazándolo lateralmente un poco y obteniendo el patrón de interferencia entre el frente de onda original y desplazado.

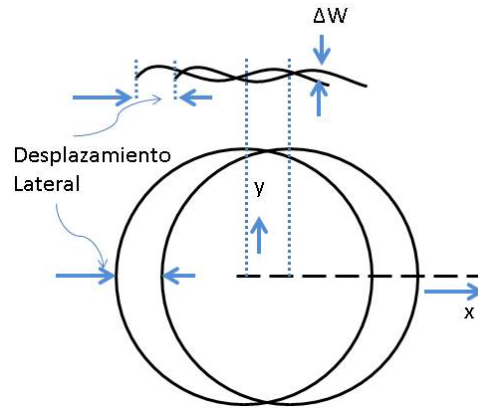


Fig. 2.7. Frente de onda original y desplazado de una apertura circular.

La teoría básica del desplazamiento de fase lateral nos dice que el error del frente de onda  $W(x,y)$  es la diferencia entre el frente de onda actual y el deseado. El frente de onda es el lugar de todos los puntos con la misma fase en un espacio tri-dimensional. En interferometría y en óptica se expresa en términos de la longitud de onda de la fuente de iluminación. En a Fig. 2.7 se observa el frente de onda original y el desplazado lateralmente. Se asume un frente de onda plano para que los errores del frente de onda sean considerados como pequeñas desviaciones de este plano. El error del frente de onda es denotado por  $W(x,y)$ , donde  $(x,y)$  son las coordenadas de un punto arbitrario  $P(x,y)$ . Cuando el frente de onda es desplazado en la dirección  $x$  una distancia  $S$ , el error en cada punto en el frente de onda desplazado es  $W(x-S,y)$ . Resultando una diferencia del frente de onda  $\Delta W(x,y)$  en  $P(x,y)$  entre el frente de onda original y el desplazado  $W(x,y) - \Delta W(x-S,y)$ . Así en interferometría  $\Delta W(x,y)$  es la medida encontrada. La diferencia del frente de onda  $\Delta W(x,y)$  puede ser expresada en términos de la longitud de onda como

$$\Delta W(x,y) = n\lambda \quad (2.29)$$

donde  $n$  es el orden de la franja de interferencia y  $\lambda$  es el frente de onda usado. La ec. 2.29 se puede multiplicar por 1 en la forma específica  $(S/\delta x)$ .



Cuando el desplazamiento  $S$  se hace muy pequeño y teóricamente se aproxima a cero el cambio en la diferencia del frente de onda con respecto al desplazamiento  $\Delta x$  se convierte en una derivada parcial y la ec. 2.29 se puede escribir de la siguiente forma

$$\frac{\partial W(x, y)}{\partial x} S = n\lambda \quad (2.30)$$

Así la información obtenida en el interferómetro de desplazamiento lateral es la aberración del rayo ( $\delta W/\delta x$ ) en unidades angulares.

Nosotros usamos la prueba de Ronchi con la LCD como rejilla para convertir esta configuración en un interferómetro de desplazamiento lateral.

### **2.3.3. Detectando la fase del frente de onda.**

La diferencia entre varios esquemas de detección PSI es la forma en la cual la fase de referencia se varía, el número de veces que se varía y el rango en el cual el patrón de interferencia es medido. En nuestro caso empleamos la ec. 2.51 del algoritmo Hariharan [14,15], la cual es evaluada en cada punto para obtener un mapa de fase del frente de onda. Este mapa de fase del frente de onda puede ser fácilmente relacionado a la altura de la superficie o a la diferencia de camino óptico (OPD) [15].

$$OPD(x, y) = \lambda \varphi(x, y) / 4\pi \quad (2.31)$$

### **2.3.4. Algoritmos PSI.**

Los algoritmos usan tanto la fase por pasos, como los métodos de integración que son los más comunes para PSI. Un número de estrategias diferentes para obtener la fase se han desarrollado de las cuales los algoritmos de los tres pasos, cuatro pasos y Hariharan, son solo unos ejemplos. Todos estos algoritmos tienen características comunes: requieren que una serie de interferogramas sean grabados cuando se cambia la fase

de referencia. El módulo  $2\pi$  de la fase del frente de onda es calculado en cada punto, así como la función arcotangente de la intensidad del interferograma en cada punto individual. El mapa final del frente de onda es obtenido desarrollando las fases para remover las discontinuidades de fase de  $2\pi$ . Las diferencias entre varios algoritmos se relacionan por el número de interferogramas grabados, el desplazamiento de fase entre estos interferogramas y la susceptibilidad del algoritmo a errores en el desplazamiento de fase o ruido ambiental tal como vibración o turbulencia.

Todos los algoritmos de PSI requieren que una serie de interferogramas sean grabados cuando la fase de referencia es desplazada y la fase del frente de onda es calculada en cada sitio de medida como una función de las intensidades medidas en ese sitio el resultado de estos cálculos debe de pasar a través de un proceso de desenvolvimiento de fase antes de que el mapa del frente de onda final sea producido. Los algoritmos son igualmente validos tanto para métodos de incrementos de fase, obtención de datos por integración bucket [15] u obtención de datos con desplazamiento de fase lateral. Para muchas aplicaciones el algoritmo Hariharan provee un buen resultado entre complejidad computacional y susceptibilidad a errores.

Con los algoritmos PSI el valor inicial de la fase de referencia a menudo se selecciona para producir una expresión matemática más simple del frente de onda medido. En la práctica no se conoce (se necesita conocer) la fase de referencia absoluta, que es importante para los algoritmos en el desplazamiento de fase entre medidas.

#### 2.3.4.1. Técnica de detección de fase.

Si consideramos un interferómetro Twyman-Green como el de la Fig. 2.8, si se asume que las componentes son ideales, el frente de onda de referencia y el frente de onda bajo prueba son dados por las ecs. 2.32 y 2.33.

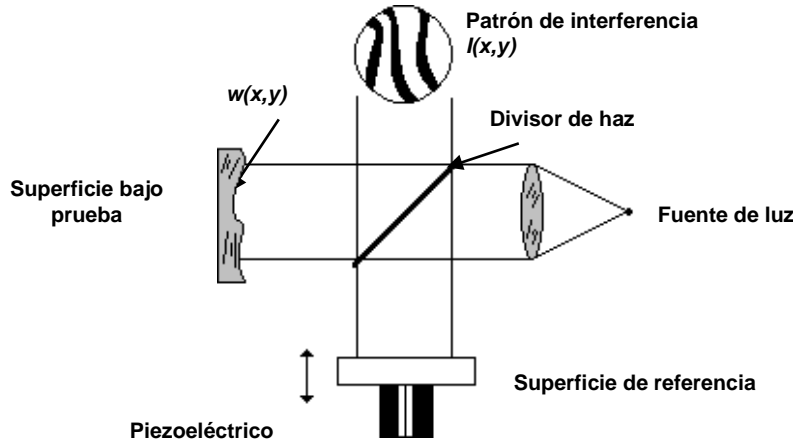
$$\omega_1 = a \exp(2ikt) \quad (2.32)$$

$$\omega_2 = b \exp[2ikw(x, y)] \quad (2.33)$$

$$k = 2\pi / \lambda \quad (2.34)$$

donde  $l$  es la longitud de camino medio del divisor de haz a la superficie de referencia,  $w(x, y)$  representa el perfil buscado de la superficie bajo prueba y  $a$  y  $b$  son las amplitudes de los frentes de onda. De las ecs. 2.32 y 2.33 encontramos que la distribución de intensidad del patrón de interferencia es

$$I(x, y, l) = (\omega_1 + \omega_2)(\omega_1 + \omega_2)^* = a^2 + b^2 + 2ab \cos 2k[\omega(x, y) - l] \quad (2.35)$$



**Fig. 2.8. Interferómetro Twyman-Green con el piezoeléctrico en el brazo de referencia para controlar la longitud de la trayectoria de camino óptico.**

El espaciamiento entre los mínimos de intensidad o entre franjas, corresponde a la diferencia de camino óptico (OPD) en los dos brazos del interferómetro es de  $\lambda/2$ . Lo que realmente se quiere es que la función  $w(x, y)$  sea una superficie o una OPD. De la ec. 2.35 se ve claramente que  $I(x, y)$  es una función senoidal de  $l$  para toda  $(x, y)$  dentro del patrón de franjas.

Considerando una representación alternativa para la ec. 2.35 se obtiene [25]

$$I(x, y, l) = a_0 + a_1 \cos 2kl + b_1 \sin 2kl \quad (2.36)$$

Esta es una representación en series de Fourier, únicamente con el término de dc y los primeros armónicos. Los coeficientes  $a_1$  y  $b_1$  son funciones de  $x$  y  $y$ . En el sentido de datos muestreados, los coeficientes  $a_1$  y  $b_1$  son

encontrados en cada punto  $(x,y)$ , muestreando secuencialmente el patrón de franjas y haciendo uso de las propiedades de ortogonalidad de las funciones trigonométricas:

$$a_0 = \frac{1}{np} \sum_{i=1}^{np} I(x, y, l_i) = a^2 + b^2 \quad (2.37)$$

$$a_1 = \frac{2}{np} \sum_{i=1}^{np} I(x, y, l_i) \cos 2kl_i = 2ab \cos 2k\omega(x, y) \quad (2.38)$$

$$b_1 = \frac{2}{np} \sum_{i=1}^{np} I(x, y, l_i) \text{sen} 2kl_i = 2ab \text{sen} 2k\omega(x, y) \quad (2.39)$$

así

$$2k\omega(x, y) = \tan^{-1} \left( \frac{b_1}{a_1} \right) \text{mod } 2\pi \quad (2.40)$$

$$l = l_i = \frac{i\lambda}{n2} \quad i = 1, 2, \dots, np \quad (2.41)$$

El entero  $p$  denota el número de periodos (franjas) sobre la cual el patrón de interferencia es muestreado y  $n$  es la cuantización de cada periodo. Por lo tanto en cada punto del patrón de interferencia, la fase del frente de onda se encuentra dentro de algún múltiplo de  $2\pi$ . Debido a que  $\omega(x, y)$  es una función continua de  $(x, y)$  dentro de la apertura, discontinuidades de  $\lambda/2$  son fácilmente resueltas.

La representación de series de Fourier es una autocorrelación o técnica de detección de sincronía. Los coeficientes determinados, representan la mejor aproximación en el sentido de mínimos cuadrados para  $I(x, y, l)$ . Debido a que esta es una técnica de detección síncrona, es sensible a movimientos y turbulencia con componentes de frecuencia aproximados de  $2kl(t)$  el recíproco del tiempo de muestreo. El tiempo de muestreo en este caso es el tiempo requerido para mover el espejo de referencia  $p\lambda/2$  o  $p$  periodos de la función  $I[x, y, l(t)]$ . La sensibilidad de esta fuente de error es minimizada muestreando  $I(x, y, l_i)$  en un orden pseudoaleatorio. Haciendo aleatorio el

orden de los datos muestreados, transforma un movimiento con una componente principal cercana a la carga, en una con componentes frecuenciales uniformemente distribuidas sobre un espectro de frecuencia muy ancho. Si los datos son almacenados sobre varios periodos ( $p$  grande) en la presencia de movimientos lentos y grandes, la técnica de detección no dará resultados que mejoren con el tiempo de muestreo. Sin embargo, si varios frentes de onda son promediados, cada uno tomado de los datos acumulados sobre intervalos de tiempo relativamente pequeños con respecto al movimiento ( $p$  pequeño), los resultados mejoran con el número de datos. Esto resulta debido a que los movimientos tienden a ser lentos y lineales, estos efectos pueden ser removidos de los frentes de onda individuales con el cálculo y la extracción residual de los errores de inclinación (tilt) y foco. El movimiento puede ser evaluado para cada corrida, la cual normalmente consiste de 4 periodos con 25 muestras por periodo ( $p = 4, n = 25$ ).

#### 2.3.4.2. Algoritmo de 3 pasos de 0, 90 y 180 grados.

La ec. 2.42 corresponde al algoritmo PSI [15,26] el cual necesita tres patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0, \pi/2$  y  $\pi$  rad) de desplazamientos de fase.

$$\tan \varphi = -\frac{-I_1 + 2I_2 - I_3}{I_1 - I_3} \quad (2.42)$$

#### 2.3.4.3. Algoritmo de 3 pasos de 0, 120 y 240 grados.

La 2.43 corresponde al algoritmo PSI [15,26] el cual necesita tres patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0, 2\pi/3$  y  $4\pi/3$  rad) de desplazamientos de fase.

$$\tan \varphi = -\sqrt{3} \frac{I_1 - I_3}{I_1 - 2I_2 + I_3} \quad (2.43)$$

**2.3.4.4. Algoritmo de 4 pasos de 0, 90, 180 y 270 grados.**

La ec. 2.44 corresponde al algoritmo PSI [15,26] el cual necesita cuatro patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0$ ,  $\pi/2$ ,  $\pi$  y  $3\pi/2$  rad) de desplazamientos de fase.

$$\tan \varphi = -\frac{I_2 - I_4}{I_1 - I_3} \quad (2.44)$$

**2.3.4.5. Algoritmo de 4 pasos de 0, 120, 240 y 360 grados.**

La ec. 2.45 corresponde al algoritmo PSI [15,26] el cual necesita cuatro patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0$ ,  $2\pi/3$ ,  $4\pi/3$  y  $2\pi$  rad) de desplazamientos de fase.

$$\tan \varphi = -\sqrt{3} \frac{I_2 - I_3}{I_1 - I_2 - I_3 + I_4} \quad (2.45)$$

**2.3.4.6. Algoritmo de 5 pasos de 0, 45, 90, 135 y 180 grados.**

La ec. 2.46 corresponde al algoritmo PSI [15,26] el cual necesita cinco patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0$ ,  $\pi/4$ ,  $\pi/2$ ,  $3\pi/4$  y  $\pi$  rad) de desplazamientos de fase.

$$\tan \varphi = -\frac{-I_1 + 4I_2 - 4I_4 + I_5}{I_1 + 2I_2 - 6I_3 + 2I_4 + I_5} \quad (2.46)$$

**2.3.4.7. Algoritmo Hariharan de 5 pasos de 0, 90, 180, 270 y 360 grados.**

Este algoritmo de PSI es insensible a los errores en el desplazamiento de la fase de referencia y fue desarrollado por Hariharan [14,15]. Este algoritmo necesita cinco patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0$ ,  $\pi/2$ ,  $\pi$ ,  $4\pi/3$ ,  $2\pi$  and  $5\pi/2$  rad) de desplazamientos de fase como se expresa con las ecs. 2.47 y 2.48 donde  $\alpha=90$

$$\delta_i = -2\alpha, -\alpha, 0, \alpha, 2\alpha; \quad i = 1, 2, 3, 4, 5. \quad (2.47)$$

Entonces

$$\begin{aligned}
 I_1(x, y) &= I'(x, y) + I''(x, y, \Delta) \cos[\phi(x, y) - 2\alpha] \\
 I_2(x, y) &= I'(x, y) + I''(x, y, \Delta) \cos[\phi(x, y) - \alpha] \\
 I_3(x, y) &= I'(x, y) + I''(x, y, \Delta) \cos[\phi(x, y)] \\
 I_4(x, y) &= I'(x, y) + I''(x, y, \Delta) \cos[\phi(x, y) + \alpha] \\
 I_5(x, y) &= I'(x, y) + I''(x, y, \Delta) \cos[\phi(x, y) + 2\alpha]
 \end{aligned}
 \tag{2.48}$$

Estas cinco ecuaciones son expandidas y combinadas para dar el resultado

$$\frac{\tan[\phi(x, y)]}{2\text{sen}(\alpha)} = \frac{I_2(x, y) - I_4(x, y)}{2I_3(x, y) - I_5(x, y) - I_1(x, y)}
 \tag{2.49}$$

Podemos escoger el desplazamiento de fase  $\alpha$  y minimizar la variación de esta expresión a errores en la fase desplazada. Diferenciando esta ecuación con respecto a  $\alpha$  encontramos

$$\frac{d}{d\alpha} \left\{ \frac{\tan[\phi(x, y)]}{2\text{sen}(\alpha)} \right\} = \frac{-\cos(\alpha) \tan[\phi(x, y)]}{2\text{sen}^2(\alpha)}
 \tag{2.50}$$

La cual se hace cero cuando  $\alpha = \pi/2$ . Cuando se usa este valor para el desplazamiento de fase, la ec. 2.51 se vuelve insensible a los errores en el desplazamiento de la fase y reduce el algoritmo final a

$$\phi(x, y) = \tan^{-1} \left[ \frac{2(I_2(x, y) - I_4(x, y))}{2I_3(x, y) - I_5(x, y) - I_1(x, y)} \right]
 \tag{2.51}$$

#### 2.3.4.8. Algoritmo de 6 pasos de 0, 90, 180, 240, 360 y 450 grados.

La ec. 2.52 corresponde al algoritmo PSI [15,26] el cual necesita seis patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0, \pi/2, \pi, 4\pi/3, 2\pi$  y rad) de desplazamientos de fase.

$$\tan \varphi = \frac{-(I_1 + 3I_2 - 4I_4 - I_5 + I_6)}{I_1 - I_2 - 4I_3 + 3I_5 + I_6}
 \tag{2.52}$$

**2.3.4.9. Algoritmo de 7 pasos de 0, 60, 120, 180, 240, 300 y 360 grados.**

La ec. 2.53 corresponde al algoritmo PSI [15,26] el cual necesita siete patrones de franjas de la superficie bajo prueba con un tamaño de paso ( $0, \pi/3, 2\pi/3, \pi, 4\pi/3, 5\pi/3$  and  $2\pi$  rad) de desplazamientos de fase.

$$\tan \varphi = -\sqrt{3} \frac{I_2 + I_3 - I_5 - I_6}{I_1 + I_2 - I_3 + 2I_4 - I_5 + I_6 + I_7} \quad (2.53)$$

## ***2.4. Prueba de Ronchi.***

La prueba de Ronchi es muy usada, debido a que es fácil de hacer y fácil de interpretar las observaciones experimentales. Es una de las más simples y poderosas técnicas para medir las aberraciones de un sistema óptico.

La prueba de Ronchi tiene dos modelos equivalentes; unos es geométrico interpretando las franjas como sombras de las bandas de la rejilla, y la otra es física interpretando las franjas como difracción e interferencia.

### **2.4.1. Teoría geométrica.**

Como lo demostró Malacara [4] la Prueba de Ronchi realmente mide la aberración transversal directa como se muestra en la Fig. 2.9. En esta figura tanto el objeto como la imagen están en eje y se puede incluir defoco así como otras aberraciones.

La aberración de onda está definida en la pupila de salida del sistema óptico bajo prueba.



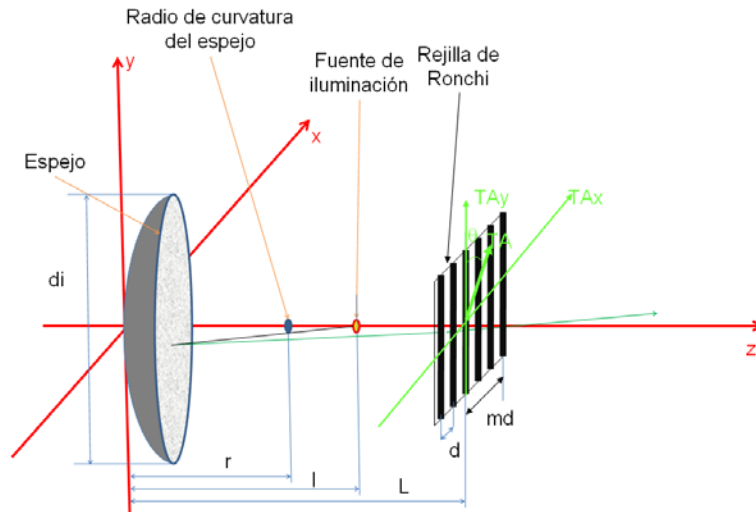


Fig. 2.9. Diagrama de la prueba de Ronchi

Usando la fórmula de Rayces

$$\frac{\partial W}{\partial x} = -\frac{TA_x}{r - W'} \quad \frac{\partial W}{\partial y} = -\frac{TA_y}{r - W'} \quad (2.54)$$

Por propósitos prácticos, se pueden describir de la siguiente manera

$$\frac{\partial W}{\partial x} = -\frac{TA_x}{r} \quad \frac{\partial W}{\partial y} = -\frac{TA_y}{r} \quad (2.55)$$

donde:  $r$  es el radio de curvatura del frente de onda

$W$  es el frente de onda

$TA_x$  es la aberración transversal en el eje  $x$

$TA_y$  es la aberración transversal en el eje  $y$

Así se asume que el espacio  $d$  de la rejilla de Ronchi de las líneas, para un punto  $(x,y)$  del  $m$ th franja puede ser escrita de manera general, como

$$\frac{\partial W}{\partial x} \cos \varphi - \frac{\partial W}{\partial y} \sin \varphi = -\frac{md}{r} \quad (2.56)$$

donde se asume que las líneas de la rejilla están inclinadas un ángulo  $\varphi$  con respecto al eje  $y$  y Así esta es la fórmula básica para el modelo geométrico de la prueba de Ronchi.

Las aberraciones primarias del frente de onda de un sistema pueden ser escritas como

$$W = A(x^2 + y^2)^2 + Bx(x^2 + y^2) + C(x^2 + 3y^2) + D(x^2 + y^2) \quad (2.57)$$

donde A, B, C son los coeficientes de las aberraciones esférica, coma y astigmatismo respectivamente. El coeficiente D es el defoco dado por  $l'$  que es la distancia de la rejilla de Ronchi al foco paraxial.

La prueba de Ronchi es usada para obtener la aberración transversal vertical **TA<sub>y</sub>** y la aberración transversal horizontal **TA<sub>x</sub>**, del sistema óptico bajo prueba; ambas aberraciones son necesarias para poder reconstruir el frente de onda del sistema óptico bajo prueba.

Nosotros obtenemos las aberraciones transversales implementando la Interferometría de desplazamiento de fase lateral, la cual nos pide tener 3 o más interferogramas del sistema óptico bajo prueba con diferentes desplazamientos de fase entre ellos.

Para dar estos desplazamientos de fase experimentalmente, se ha trabajado en 2 métodos diferentes para lograrlo:

- El primer método consiste en tener una rejilla de Ronchi comercial, y con un motor de pasos desplazar la rejilla lateralmente pequeñas distancias para obtener el desplazamiento de fase lateral deseado.
- El segundo método consiste en poner una pantalla de cristal líquido (LCD) en lugar de la rejilla y desde una computadora introducir los desplazamientos de fase correspondientes a través de la LCD.

Los interferogramas son capturados mediante una cámara CCD, con la cual los datos son digitalizados y pueden ser procesados, para obtener la fase envuelta, desenvolver dicha fase envuelta, obtener el frente de onda y finalmente graficar las aberraciones del sistema óptico y así mismo obtener las aberraciones del sistema óptico bajo prueba, con esto los sistemas ópticos quedan caracterizados cuantitativamente.

## 2.5. Momentos geométricos.

Los momentos geométricos [16] son las funciones más simples entre las funciones de los momentos, con la función kernel definida como el producto de las coordenadas de los píxeles. La principal ventaja de los momentos geométricos es que las transformaciones de las coordenadas de la imagen pueden ser fácilmente expresadas y analizadas en términos de las correspondientes transformaciones en los momentos espacio. Las funciones de los momentos geométricos que son invariantes con respecto a las transformaciones del plano imagen tienen demasiadas aplicaciones en la identificación de objetos y en su estimación de pose. El cómputo de los momentos geométricos en imágenes puede ser fácilmente desarrollado e implementado así como comparado con otros momentos con funciones de kernel complejo. Los momentos geométricos son también conocidos como momentos cartesianos o momentos regulares.

### 2.5.1. Definición y propiedades.

Los momentos geométricos son definidos con el conjunto base  $\{x^p, y^q\}$ . Los órdenes bidimensionales  $(p+q)^{th}$  son denotados por  $m_{pq}$ , y pueden ser expresados como

$$m_{pq} = \iint_{\zeta} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, 3, \dots \quad (2.58)$$

donde  $\zeta$  es la región del espacio en píxeles en la cual la función intensidad de la imagen  $f(x, y)$  está definida. La ec. 2.58 tiene la forma de la proyección de la función  $f(x, y)$  dentro del monomio  $x^p, y^q$ .

### Diferentes tipos de momentos geométricos

Variaciones en la definición de los momentos geométricos dada por la ec. 2.58 pueden ser encontradas en la literatura, dependiendo el área de aplicación.

Los momentos Silhouette se refieren a los momentos calculados de una imagen binaria. Aquí el valor de la intensidad  $f(x, y)$  toma únicamente dos valores 0 y 1. Los niveles de grises de las imágenes en ocasiones se usa un valor de umbral para discretizarlas y segmentar el objeto del fondo. A los píxeles en la región del objeto se les asigna el valor de 1 y a los píxeles que está en la región de fondo se les asigna el valor de 0. En este caso la región de la imagen  $\zeta$  consiste únicamente de aquellos píxeles los cuales corresponden a puntos en el objeto y tienen un valor 1 y tales imágenes son llamadas imágenes Silhouette.

Los momentos de frontera de una imagen son computados usando únicamente los puntos de frontera de la imagen. La forma de un objeto es a menudo representada por su contorno y consecutivamente los puntos de frontera son únicamente usados en tales aplicaciones, para minimizar el espacio de almacenamiento de datos y el tiempo de cómputo. Chen definió los momentos de frontera de una imagen binaria como

$$m_{pq} = \iint_C x^p y^q ds \quad p, q = 0, 1, 2, 3, \dots \quad (2.59)$$

donde  $ds = (dx^2 + dy^2)^{1/2}$  y  $C$  denota el contorno.

Los momentos estándar son los momentos geométricos usuales de las imágenes los cuales son normalizados con respecto a escala, traslación y rotación.

$$m_{00} = 1; \quad m_{10} = m_{01} = m_{11} = 0; \quad m_{20} \geq m_{02}; \quad m_{30} \geq 0 \quad (2.60)$$

### 2.5.2. Representación de forma usando momentos geométricos.

Los momentos geométricos de diferentes órdenes, representan diferentes características espaciales de la distribución de intensidades de la imagen.

Así un conjunto de momentos pueden formar un descriptor global de forma de una imagen.

Por definición, el momento de orden cero ( $m_{0,0}$ ) representa la intensidad total de una imagen. Para una imagen Silhouette este término da el área geométrica de la región de la imagen.

Las funciones de primer orden  $m_{1,0}$ ,  $m_{0,1}$  dan los momentos de intensidad sobre el eje  $y$  y el eje  $x$  de la imagen respectivamente. El centroide de intensidad  $(x_0, y_0)$  está dado por

$$x_0 = \frac{m_{10}}{m_{00}} ; \quad y_0 = \frac{m_{01}}{m_{00}} \quad (2.61)$$

Los momentos computados con respecto al centroide de intensidad son llamados momentos centrales y son definidos como

$$\mu_{pq} = \iint_{\mathcal{S}} (x - x_0)^p (y - y_0)^q dx dy, \quad p, q = 0, 1, 2, 3, \dots \quad (2.62)$$

De la definición de los momentos centrales, tenemos

$$\mu_{00} = m_{00}; \quad \mu_{10} = \mu_{01} = 0 \quad (2.63)$$

Los momentos de inercia ( $\mu_{20}, \mu_{02}$ ) de la imagen sobre el sistema de referencia son llamados los principales momentos de inercia de la imagen. Si

$\mu_{20}, \mu_{02}, \mu_{11}$  son los momentos centrales de segundo orden de la imagen en su cuadro actual de referencia de la imagen y si  $I_1, I_2$  se refieren a los principales valores de inercia del momento, entonces

$$I_1 = \frac{(\mu_{20} + \mu_{02}) + [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{2}, \quad I_2 = \frac{(\mu_{20} + \mu_{02}) - [(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2]^{1/2}}{2} \quad (2.64)$$

El ángulo de orientación  $\theta$  de uno de los principales ejes de inercia con el eje  $x$  está dado por la siguiente ecuación:

$$\theta = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (2.65)$$

Las ecuaciones de  $I_1, I_2, \theta$  pueden ser usadas para definir la elipse de una imagen la cual tiene los mismos momentos de inercia y ejes de dirección principales como la imagen original Fig. 2.10. Los radios  $a$  y  $b$  del eje mayor y menor de la elipse están dados por:

$$a = 2 \left( \frac{I_1}{\mu_{00}} \right)^{1/2}; \quad b = 2 \left( \frac{I_2}{\mu_{00}} \right)^{1/2} \quad (2.66)$$

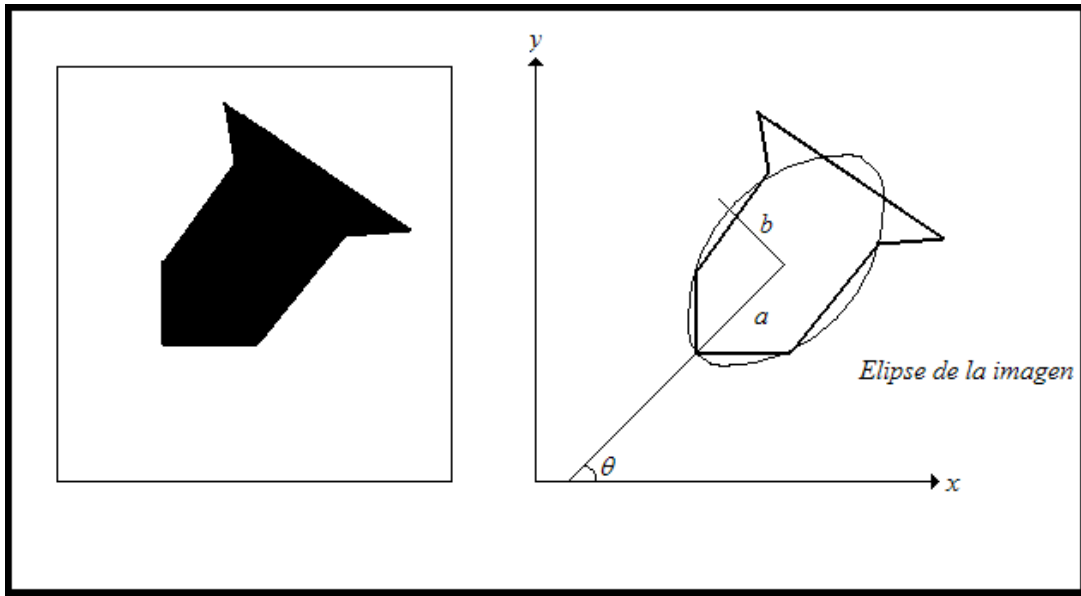


Fig. 2.10. Una imagen y la elipse correspondiente de la imagen

## 2.6. Desenvolvimiento de fase

Cuando se emplea la técnica del cambio de fase interferométrica (PSI) más de una operación debe ser desarrollada para la fase  $\phi(x, y)$  calculada, antes de que esté lista para ser mostrada y evaluada. Se deben corregir las discontinuidades que ocurren en el cálculo de la fase como resultado de la

función arcotangente. La función arcotangente está definida únicamente sobre el rango limitado por los ángulos,  $-\pi/2$  a  $\pi/2$ .

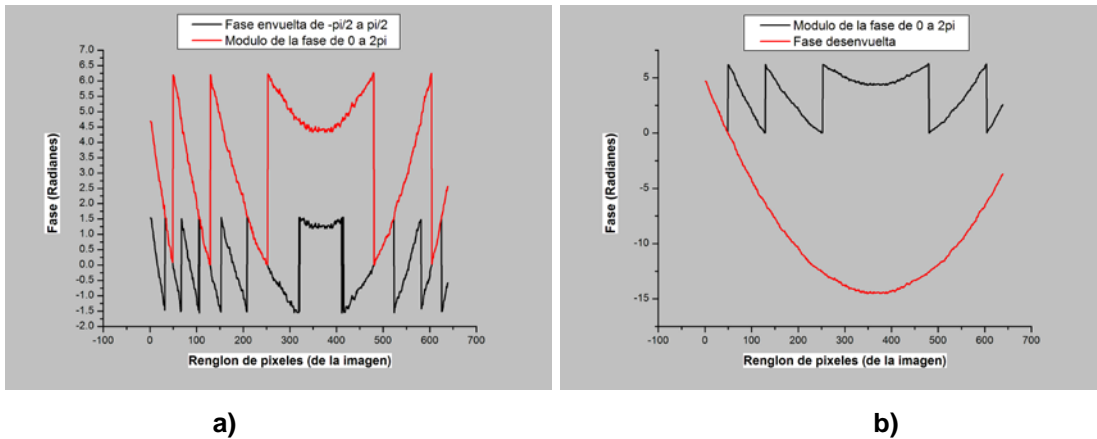
La primera corrección que puede ser usada para calcular la fase es extender el rango de cálculo de  $0$  a  $2\pi$ . Esto es posible ya que los signos del seno y coseno son conocidos independientemente del seno de la tangente. El numerador y denominador de la ecuación para calcular la fase con el algoritmo Hariharan, son directamente proporcionales al seno y al coseno. Relaciones similares existen para otros algoritmos la Tabla 2.1 da las condiciones necesarias para convertir los resultados del arcotangente a valores entre  $0$  y  $2\pi$  como una función de los valores de seno y coseno, este proceso es mostrado gráficamente en la Fig. 2.11 a).

**Tabla 2.1. Corrección del módulo  $2\pi$  de la fase**

Seno	Coseno	Fase corregida $\phi(x,y)$	Rango de fase
0	+	0	0
+	+	$\phi(x,y)$	0 to $\pi/2$
+	0	$\pi/2$	$\pi/2$
+	-	$\phi(x,y)+\pi$	$\pi/2$ to $\pi$
0	-	$\pi$	$\pi$
-	-	$\phi(x,y)+\pi$	$\pi$ to $3\pi/2$
-	0	$3\pi/2$	$3\pi/2$
-	+	$\phi(x,y)+2\pi$	$3\pi/2$ to $2\pi$

El paso final en el proceso de reconstrucción del frente de onda es quitar las discontinuidades  $2\pi$  que están presentes en los datos del módulo  $2\pi$  de la fase que han sido generados. Este proceso es conocido como desenvolvimiento de la fase, integración de la fase o continuidad de la fase y convierte los datos en módulo de fase  $2\pi$  a una representación continua del frente de onda bajo prueba. Siempre que una discontinuidad grande ocurre en la reconstrucción,  $2\pi$  o múltiplos de  $2\pi$  son sumados o restados

para unir los datos y remover la discontinuidad. Este proceso es mostrado gráficamente para datos de una dimensión en la Fig. 2.11 b).



**Fig. 2.11 a) Conversión de la fase calculada por el arcotangente, a el módulo  $2\pi$  de la fase del frente de onda; b) proceso de desenvolvimiento de fase en una dimensión.**

El problema de desenvolvimiento es trivial para mapas de fase calculados de interferogramas de buena calidad cuando las siguientes dos condiciones se satisfacen:

1. La señal está libre de ruido.
2. La condición de Nyquist no se viola, lo cual significa que el valor absoluto de la diferencia de fase entre dos píxeles adyacentes es menor que  $\pi$ .

### 2.6.1. Desenvolvimiento de fase, en 2-D por Tomas Kreis.

Para desenvolver la fase en dos dimensiones Tomas Kreis [17] se toman las diferencias de fase para saber si se suma o se resta un múltiplo de  $2\pi$  a la fase, se desenvuelve el primer renglón de la fase y después cada píxel de ese renglón sirve como condición inicial para desenvolver la fase en sentido vertical como se observa en la Fig. 2.12 a). Un algoritmo diferente para desenvolver la fase se ilustra en la Fig. 2.12 b). Para otro tipo de



desenvolvimiento usamos las mismas ecuaciones siguiendo las trayectorias de la Fig. 2.12. c).

$$n(x_0, y_0) = 0$$

$$n(x_1, y_i) = \begin{cases} n(x_1, y_{i-1}) & \text{if } |\phi(x_1, y_i) - \phi(x_1, y_{i-1})| < \pi \\ n(x_1, y_{i-1}) + 1 & \text{if } \phi(x_1, y_i) - \phi(x_1, y_{i-1}) \leq -\pi \\ n(x_1, y_{i-1}) - 1 & \text{if } \phi(x_1, y_i) - \phi(x_1, y_{i-1}) \geq \pi \end{cases} \quad i = 2, 3, \dots$$

$$n(x_j, y_i) = \begin{cases} n(x_{j-1}, y_i) & \text{if } |\phi(x_j, y_i) - \phi(x_{j-1}, y_i)| < \pi \\ n(x_{j-1}, y_i) + 1 & \text{if } \phi(x_j, y_i) - \phi(x_{j-1}, y_i) \leq -\pi \\ n(x_{j-1}, y_i) - 1 & \text{if } \phi(x_j, y_i) - \phi(x_{j-1}, y_i) \geq \pi \end{cases} \quad j = 2, 3, \dots$$

$$\phi_{desenvuelta}(x_j, y_i) = \phi(x_j, y_i) + 2\pi n(x_j, y_i) \quad i, j = 1, 2, 3, \dots \quad (2.67)$$

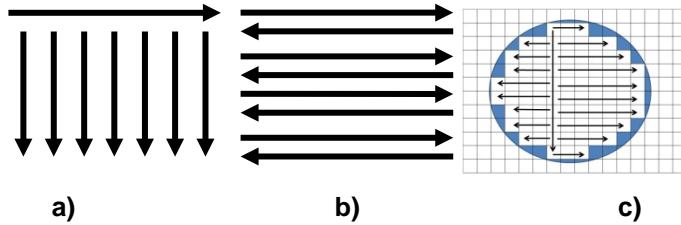


Fig. 2.12. a), b) y c) Trayectorias para desenvolver la fase en dos dimensiones.

### 2.6.2. **Desenvolvimiento de fase, de mapas de fase. consistente de campo completo**

Esta técnica para desenvolver la fase es una de las más simples para desenvolver un buen mapa de fase liso (pequeño ruido de error). La técnica consiste en integrar las diferencias de fase a lo largo de una trayectoria como se muestra en la Fig. 2.13, (Malacara and Servin [26]) Si asumimos que el mapa de fase de campo-completo está dado por  $\phi_w(x, y)$  en un arreglo regular bidimensional  $L$  de tamaño  $M \times N$  píxeles. Podemos desenvolver este mapa de fase desenvolviendo el primer renglón ( $j = 0$ ) y de este tomar el último valor como condición inicial para desenvolver el siguiente renglón

de mapa de fase en la dirección positiva. Esto se hace usando la siguiente formula a lo largo del primer renglón:

$$\phi(x_{i+1}, y_0) = \phi(x_i, y_0) + V(\phi_W(x_{i+1}, y_0) - \phi(x_i, y_0)); \quad 1 \leq i \leq N \quad (2.68)$$

donde  $V(x) = \left[ x - 2\pi \text{int}\left(\frac{x}{\pi}\right) \right]^2$  valida en el intervalo  $(-\pi, \pi)$  es la función envuelta. Esta función es igual a  $V(x) = \tan^{-1}\left[\frac{\sin x}{\cos x}\right]$  en el mismo rango.



Fig. 2.13. Trayectoria que se sigue para desenvolver la fase en dos dimensiones

En ec. 2.68 Debemos usar como condición inicial

$$\phi(x_0, y_0) = \phi_0 \quad (2.69)$$

Teniendo la fase desenvuelta del primer renglón, debemos usar el último valor de la fase desenvuelta como la condición inicial para desenvolver la fase del segundo renglón ( $j = 1$ ) en la dirección negativa que es

$$\phi(x_{i-1}, y_1) = \phi(x_i, y_1) + V(\phi_W(x_{i-1}, y_1) - \phi(x_i, y_1)), \quad 1 \leq i \leq N \quad (2.70)$$

Para la dirección de regreso, debemos usar como condición inicial

$$\phi(x_{N-1}, y) = \phi(x_{N-1}, y_0) + V(\phi_W(x_{N-1}, y_1) - \phi(x_{N-1}, y_0)) \quad (2.71)$$

El desenvolvimiento entonces procede para el siguiente renglón ( $j = 2$ ) en la dirección positiva

$$\phi(x_{i+1}, y_2) = \phi(x_i, y_2) + V(\phi_W(x_{i+1}, y_2) - \phi(x_i, y_2)) \quad 1 \leq i \leq N \quad (2.72)$$

y la condición inicial

$$\phi(x_0, y_2) = \phi(x_0, y_1) - V(\phi_W(x_0, y_2) - \phi(x_0, y_1)) \quad (2.73)$$

el procedimiento de escaneo aquí descrito se sigue hasta que el mapa de fase de campo-completo esta desenvuelto.

### 2.6.3. **Desenvolvimiento de fase, de mapas de fase consistente dentro de una región conectada simple.**

De acuerdo a Malacara-Servin [26], si la forma del mapa de fase consistente está rodeado por una región conectada simple arbitraria como se muestra en la Fig. 2.14, el algoritmo descrito en la sección anterior en este caso se debe de usar el siguiente algoritmo para desenvolver un mapa de fase consistente. Para iniciar, definir y poner a cero una función indicador  $\sigma(x, y)$  dentro del dominio  $D$  de datos de fase validos (como se muestra en la Fig. 2.14). Entonces se selecciona un punto de inicio dentro de la región del dominio  $D$  y asignarle un valor de fase arbitraria  $\phi(x, y) = \phi_0$ . Marcar el sitio visitado como desenvuelto, por ejemplo  $\sigma(x, y)$  igual a 1. Ahora que el píxel de fase de inicio está definido, se procede con el proceso de desenvolvimiento siguiente:

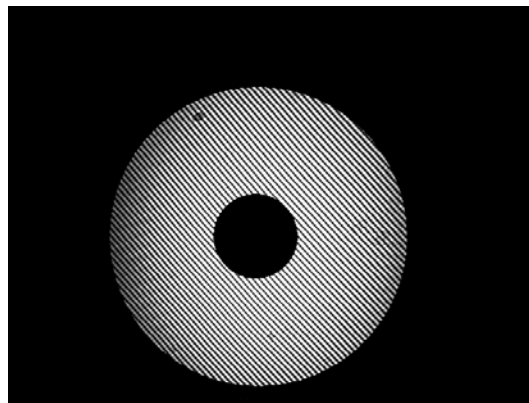


Fig. 2.14 Ejemplo de una región conectada simple que contiene datos de fase validos

1. Seleccionar un píxel  $(x, y)$  dentro de  $D$  (en un orden aleatorio o en un orden predescrito)
2. Verificar si el sitio visitado  $(x, y)$  dentro de  $D$  ya ha sido desenvuelto

- a) Si el sitio seleccionado está marcado como desenvuelto [ $\sigma(x, y) = 1$ ], regresar al paso 1.
- b) Si el sitio visitado está envuelto [por ejemplo  $\sigma(x, y) = 0$ ], entonces verificar por cualquier píxel adyacente desenvuelto  $(x', y')$ .
- c) Si ningún píxel adyacente ha sido desenvuelto, regresar al paso 1.
- d) Si algún píxel adyacente  $(x', y')$  ya ha sido desenvuelto, tomar su valor de fase  $\phi(x', y')$  y usar este valor para desenvolver el sitio actual  $(x, y)$  como

$$\phi(x, y) = \phi(x', y') + V(\phi_W(x, y) - \phi(x', y'))$$

donde  $V(\cdot)$  es la función envuelta

3. Marcar el sitio visitado como desenvuelto [ $\sigma(x, y) = 1$ ].
4. Regresar al paso 1 hasta que todos los píxeles en  $D$  estén desenvueltos.

#### 2.6.4. Método robusto para desenvolver la fase en dos dimensiones.

De acuerdo con Giglia y Romero [18] suponer que conocemos la fase,  $\phi$  módulo  $2\pi$  de una función en una rejilla de puntos discretos.

Dados los valores de fase envuelta  $\varphi_{i,j}$ , se desean determinar los valores de fase desenvuelta  $\phi_{i,j}$  en la misma localización de la rejilla, con el requerimiento de que las diferencias de fase de  $\phi_{i,j}$  coincidan con  $\varphi_{i,j}$  en un sentido de mínimos cuadrados. Para ver como se hace esto, primero se define un operador envuelto  $w$  que envuelve los argumentos de todos los

valores dentro del rango  $(-\pi, \pi)$ , esto se hace sumando o restando  $2\pi$  rad de su argumento. Por ejemplo  $W\{\phi_{i,j}\} = \varphi_{i,j}$

$$\varphi_{i,j} = \phi_{i,j} + 2\pi k, \quad \mathbf{k \text{ un entero}}, \quad (2.74)$$

$$-\pi < \varphi_{i,j} \leq \pi, \quad i = 0 \dots M-1, \quad j = 0 \dots N-1 \quad (2.75)$$

Después se hace el cómputo de dos diferencias de fase: estas diferencias con respecto al índice  $i$  y al índice  $j$ . Específicamente de los valores conocidos de fase envuelta  $\varphi_{i,j}$  se computan las siguientes diferencias de fase

$$\begin{aligned} \Delta_{i,j}^x &= W\{\varphi_{i+1,j} - \varphi_{i,j}\} \\ i &= 0 \dots M-2, \quad j = 0 \dots N-1 \\ \Delta_{i,j}^x &= 0, \quad \mathbf{\text{en otro lugar}}, \end{aligned} \quad (2.76)$$

$$\begin{aligned} \Delta_{i,j}^y &= W\{\varphi_{i,j+1} - \varphi_{i,j}\} \\ i &= 0 \dots M-1, \quad j = 0 \dots N-2 \\ \Delta_{i,j}^y &= 0, \quad \mathbf{\text{en otro lugar}} \end{aligned} \quad (2.77)$$

donde los superíndices  $x$  y  $y$  se refieren a las diferencias en los índices  $i$  y  $j$  respectivamente. La solución  $\phi_{i,j}$  que minimiza

$$\sum_{i=0}^{M-2} \sum_{j=0}^{N-1} (\phi_{i+1,j} - \phi_{i,j} - \Delta_{i,j}^x)^2 + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} (\phi_{i,j+1} - \phi_{i,j} - \Delta_{i,j}^y)^2 \quad (2.78)$$

es la solución de mínimos-cuadrados.

La formulación de la matriz de Hunt muestra que las ecuaciones normales que dan la solución al desenvolvimiento de la fase en mínimos-cuadrados puede ser resumida por la siguiente ecuación:

$$\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j} = \Delta_{i,j}^x - \Delta_{i-1,j}^x + \Delta_{i,j}^y - \Delta_{i,j-1}^y \quad (2.79)$$

La ec. 2.79 da la relación entre las diferencias de fase envuelta y los valores de fase desenvuelta  $\phi_{i,j}$ , en un sentido de error de mínimos-cuadrados. Una manipulación simple de la ec. 2.77 da

$$\left(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}\right) + \left(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}\right) = \rho_{i,j} \quad (2.80)$$

donde

$$\rho_{i,j} = (\Delta_{i,j}^x - \Delta_{i-1,j}^x) + (\Delta_{i,j}^y - \Delta_{i,j-1}^y) \quad (2.81)$$

Ahora es fácil observar que la ec. 2.80 es una discretización de la ecuación de Poisson en una rejilla rectangular de  $M \times N$ .

$$\frac{\partial^2}{\partial x^2} \phi(x, y) + \frac{\partial^2}{\partial y^2} \phi(x, y) = \rho(x, y) \quad (2.82)$$

Es importante notar que la ec. 2.80 es válida para todos los índices en la rejilla rectangular,  $i = 0 \dots M - 1$   $j = 0 \dots N - 1$  y que las diferencias de fase apropiadas que son usadas para computar  $\rho_{i,j}$  en la ec. 2.81 no son cero únicamente si ellos provienen de fases enteramente dentro de la rejilla rectangular [por ejemplo ecs. 2.76 y 2.77]. Este requerimiento resulta directamente de la formulación de mínimos-cuadrados y forma la equivalencia discreta de la imposición de las condiciones de frontera de Neumann en la ecuación de Poisson. Específicamente se requiere que

$$\begin{aligned} \Delta_{-1,j}^x &= 0, \\ \Delta_{M-1,j}^x &= 0, \quad j = 0 \dots N - 1, \end{aligned} \quad (2.83)$$

$$\begin{aligned} \Delta_{i,-1}^y &= 0, \\ \Delta_{i,N-1}^y &= 0, \quad i = 0 \dots M - 1, \end{aligned} \quad (2.84)$$

También es importante notar que las diferencias de fase dadas por las ecs. 2.76 Y 2.77 son el equivalente de las diferencias de fase medidas en la formulación de Hunt. Desde que el desenvolvimiento de fase tiene sus raíces

en la óptica adaptativa, las diferencias de fase obtenidas con sensores de frente de onda de inclinación fueron necesariamente valores envueltos. Por lo tanto, desde que se está formulando el desenvolvimiento bidimensional de valores de fase envuelta medidos o computados, las diferencias de fase envuelta, no diferencias de fase ordinarias, deben ser usadas en la formación del término  $\rho_{i,j}$  de la ecuación diferencial resultante.

Ahora se muestra cómo resolver la ec. 2.79 usando una forma específica de la expansión del coseno que proporciona una implementación de la Transformada Coseno Discreta Rápida (DCT). Una forma específica de la Transformada Coseno Discreta bidimensional es dada como sigue:

Transformada Coseno Discreta bidimensional

$$C_{m,n} = \begin{cases} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} 4x_{i,j} \cos\left[\frac{\pi}{2M} m(2i+1)\right] \cos\left[\frac{\pi}{2N} n(2j+1)\right] & 0 \leq m \leq M-1; \quad 0 \leq n \leq N-1 \quad (2.85) \\ 0 & \text{en otro lugar} \end{cases}$$

Transformada Coseno Discreta bidimensional inversa

$$x_{i,j} = \begin{cases} \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_1(m) w_2(n) C_{m,n} \cos\left[\frac{\pi}{2M} m(2i+1)\right] \cos\left[\frac{\pi}{2N} n(2j+1)\right] & 0 \leq i \leq M-1, \quad 0 \leq j \leq N-1 \quad (2.86) \\ 0 & \text{en otro lugar} \end{cases}$$

$$\begin{aligned} w_1(m) &= 1/2, & m &= 0, \\ w_1(m) &= 1, & 1 \leq m \leq M-1, \\ w_2(n) &= 1/2, & n &= 0, \\ w_2(n) &= 1, & 1 \leq n \leq N-1 \end{aligned} \quad (2.87)$$

Es importante notar que la expansión del coseno en las ecuaciones de arriba imponen las condiciones de frontera de Neumann,  $\nabla \phi \cdot \bar{n} = 0$ , automáticamente y da la solución exacta de la ec. 2.80 como se muestra en la ec. .

Se expande la solución deseada  $\phi_{i,j}$  en la forma de la ec. 2.86.:

$$\phi_{i,j} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_1(m)w_2(n)\hat{\phi}_{m,n} \times \cos\left[\frac{\pi}{2M}m(2i+1)\right] \cos\left[\frac{\pi}{2N}n(2j+1)\right] \quad (2.88)$$

y sustituir la ec. 2.88 en la ec. 2.79. Desarrollar una expansión similar y sustituir para el lado derecho de la 2.79 y reorganizando el resultado da la siguiente solución exacta en el dominio de DCT:

$$\hat{\phi}_{i,j} = \frac{\hat{\rho}_{i,j}}{2\left(\cos\frac{\pi i}{M} + \cos\frac{\pi j}{N} - 2\right)} \quad (2.89)$$

La fase desenvuelta  $\phi_{i,j}$  es fácilmente obtenida con la Transformada Coseno Discreta bidimensional inversa de la ec. 2.89.

Se muestra fácilmente por simple sustitución que la expansión dada por la ec. 2.88 automáticamente impone el equivalente discreto de las condiciones de frontera de Neumann requeridas:

$$\begin{aligned} \phi_{0,j} - \phi_{-1,j} &= 0, \\ \phi_{M,j} - \phi_{M-1,j} &= 0, \quad j = 0 \dots N-1 \\ \phi_{i,0} - \phi_{i,-1} &= 0, \\ \phi_{i,N} - \phi_{i,N-1} &= 0, \quad i = 0 \dots M-1 \end{aligned} \quad (2.90)$$

El algoritmo bidimensional para desenvolver la fase es resumido como sigue:

1. Desarrollar la Transformada Coseno Discreta bidimensional del arreglo de valores  $\rho_{i,j}$  computados por la ec.2.80 para dar los valores  $\hat{\rho}_{i,j}$ .
2. Modificar los valores  $\hat{\rho}_{i,j}$  de acuerdo a la ec. 2.89 para obtener  $\hat{\phi}_{i,j}$ .



3. Desarrollar la Transformada Coseno Discreta bidimensional inversa de  $\hat{\phi}_{i,j}$  para obtener los valores de fase desenvuelta en mínimos-cuadrados  $\phi_{i,j}$

Es importante notar que no se puede evaluar la ec. 2.88 para  $i=0$  y  $j=0$  simultáneamente, porque el denominador se convierte en cero. Esto significa que  $\hat{\phi}_{0,0}$  es indeterminado porque la ecuación de Poisson no puede ser resuelta para una constante. En la práctica usualmente se pone  $\hat{\phi}_{0,0} = \hat{\rho}_{0,0}$

## 2.7. Polinomios de Zernike.

La aberración del frente de onda puede ser expresada en término de las funciones de los polinomios de Zernike los cuales son ortogonales en una región circular de radio unitario de acuerdo con Wyant [27] están relacionados a las aberraciones ópticas más comunes. Se utilizan polinomios hasta de grado 6 los cuales son 21 que da como resultado:

$$\begin{aligned}
 W(x, y) = & a_1 Z_1 + a_2 Z_2 + a_3 Z_3 + a_4 Z_4 + a_5 Z_5 + a_6 Z_6 + a_7 Z_7 + a_8 Z_8 \\
 & + a_9 Z_9 + a_{10} Z_{10} + a_{11} Z_{11} + a_{12} Z_{12} + a_{13} Z_{13} + a_{14} Z_{14} + a_{15} Z_{15} \\
 & + a_{16} Z_{16} + a_{17} Z_{17} + a_{18} Z_{18} + a_{19} Z_{19} + a_{20} Z_{20}
 \end{aligned}$$

donde  $a_i$  son los coeficientes de los polinomios y  $Z_i$  son las expansiones de los polinomios de Zernike para  $W(x,y)$ . En la tabla 2.2 se muestran los polinomios de Zernike.

**Tabla 2.2. Polinomios de zernike.**

$Z_1$ Piston	1
$Z_2$ Tilt en z	2x
$Z_3$ Tilt en Y	2y
$Z_4$ Defoco	$\sqrt{3}(2\rho^2 - 1)$
$Z_5$ Astigmatismo primario a 0°	$2\sqrt{6}xy$
$Z_6$ Astigmatismo secundario a 45°	$\sqrt{6}(x^2 - y^2)$

$Z_7$	$\sqrt{8}y(3\rho^2 - 2)$
$Z_8$	$\sqrt{8}x(3\rho^2 - 2)$
$Z_9$	$\sqrt{8}y(3x^2 - y^2)$
$Z_{10}$	$\sqrt{8}x(x^2 - 3y^2)$
$Z_{11}$	$\sqrt{5}(6\rho^4 - 6\rho^2 + 1)$
$Z_{12}$	$\sqrt{10}(x^2 - y^2)(4\rho^2 - 3)$
$Z_{13}$	$2\sqrt{10}xy(4\rho^2 - 3)$
$Z_{14}$	$\sqrt{10}(\rho^4 - 8x^2y^2)$
$Z_{15}$	$4\sqrt{10}xy(x^2 - y^2)$
$Z_{16}$	$\sqrt{12}x(10\rho^4 - 12\rho^2 + 3)$
$Z_{17}$	$\sqrt{12}y(10\rho^4 - 12\rho^2 + 3)$
$Z_{18}$	$\sqrt{12}x(x^2 - 3y^2)(5\rho^2 - 4)$
$Z_{19}$	$\sqrt{12}y(x^2 - 3y^2)(5\rho^2 - 4)$
$Z_{20}$	$\sqrt{12}x(16x^4 - 20x^2\rho^2 + 5\rho^4)$
$Z_{21}$	$\sqrt{12}y(16y^4 - 20y^2\rho^2 + 5\rho^4)$

## 2.8. Conclusión.

En este capítulo se dio una introducción sobre el fenómeno de interferencia, se explicó en que consiste la Interferometría de desplazamiento de fase PSI, se dieron las ventajas de PSI, se comentó el mejor método para hacer PSI, se explicó brevemente la prueba de Ronchi. También se definieron los momentos geométricos los cuales usamos para detectar la fase válida de los patrones de franjas digitalizados. Se explican los métodos de desenvolvimiento de fase. Y se dieron los polinomios de Zernike usados en el ajuste de los datos del frente de onda.

## *Capítulo 3*

### *3. Ecuaciones de fondo.*

#### *3.1. Introducción.*

En este capítulo se explica para que se usan las ecuaciones de fondo, se explica brevemente la ecuación de modulación de Schreiber[15], se describe como se obtuvieron las ecuaciones de fondo y se muestran las ecuaciones de fondo que se encontraron para algunos algoritmos PSI.

Cuando se trabaja en el taller de óptica con patrones de franjas obtenidos con la técnica de PSI se tienen que encontrar los píxeles de fase válida para hacer un análisis lo más exacto posible, debido a que comúnmente los sistemas ópticos bajo prueba son de geometría circular, hexagonal u alguna otra forma, la fase válida de los patrones de franjas digitalizados tiene la forma del sistema bajo prueba, el cual se encuentra dentro de un arreglo rectangular debido a la colocación de los sensores en la cámara CCD, por lo que la fase que está dentro del arreglo rectangular pero fuera del círculo es fase no válida.

Proponemos un método para detectar los píxeles de fase válida cuando se usa cualquier técnica PSI y mostramos las ecuaciones a las que se llega para algunos algoritmos PSI.

### 3.2. Ecuación de modulación de Schreiber.

Si se desea, la intensidad de los patrones de franjas digitalizados puede ser evaluada para determinar la modulación de los datos  $\gamma(x, y)$  [15] a través del interferograma o ronchigrama:

$$\gamma(x, y) = \frac{I''(x, y)}{I'(x, y)} \quad (3.1)$$

El numerador es la intensidad de modulación, y el denominador es la intensidad promedio o la intensidad de fondo. Todas las características tales como sensibilidad y el fondo están implícitas incluidas en el análisis como parte de  $I'(x, y)$  y  $I''(x, y)$ . Esta información es usada para evaluar la calidad de los datos que han sido medidos. Un dato de modulación cercano a uno es bueno y una modulación más baja es mala. Puntos de datos con valores de modulación por debajo de algún umbral tendrán insuficiente señal a ruido, y el valor de la fase del frente de onda no puede ser calculado para este punto. Estos puntos llamados de fase no valida son excluidos del análisis y típicamente los umbrales de modulación de datos están en el rango de 5-10%. Este valor es por supuesto dependiente de la aplicación. En el libro de Malacara [15] se pueden encontrar las siguientes ecuaciones de modulación para algunos algoritmos PSI.

#### 3.2.1. Ecuación de modulación para algoritmo PSI de 3 pasos de 0, 90 y 180 grados.

La ec. 3.2 es la ecuación de modulación para el algoritmo PSI de 3 pasos de 0, 90 y 180 grados de la sección 2.3.4.2.

$$\gamma(x, y) = \frac{\sqrt{(I_1 - I_3)^2 + (2I_2 - I_1 - I_3)^2}}{I_1 + I_3} \quad (3.2)$$

**3.2.2. Ecuación de modulación para algoritmo PSI de 3 pasos de 0, 120 y 240 grados.**

La ec. 3.3 es la ecuación de modulación para el algoritmo PSI de 3 pasos de 0, 120 y 240 grados de la sección 2.3.4.3.

$$\gamma(x, y) = \frac{\sqrt{(2.25)(I_1 - I_3)^2 + (0.75)(2I_2 - I_1 - I_3)^2}}{0.866(I_1 + I_3 + I_2)} \quad (3.3)$$

**3.2.3. Ecuación de modulación para algoritmo PSI de 4 pasos de 0, 90, 180 y 270 grados.**

La ec. 3.4 es la ecuación de modulación para el algoritmo PSI de 4 pasos de 0, 90, 180 y 270 grados de la sección 2.3.4.4.

$$\gamma(x, y) = \frac{2\sqrt{(I_4 - I_2)^2 + (I_1 - I_3)^2}}{I_1 + I_2 + I_3 + I_4} \quad (3.4)$$

**3.2.4. Ecuación de modulación para algoritmo de 5 pasos de Hariharan 0, 90, 180, 270 y 360 grados.**

La ec. 3.5 es la ecuación de modulación para el algoritmo PSI de 5 pasos de 0, 90, 180, 270 y 360 grados de la sección 2.3.4.7.

$$\gamma(x, y) = \frac{3\sqrt{4(I_2 - I_4)^2 + (I_1 + I_5 - 2I_3)^2}}{2(I_1 + I_2 + 2I_3 + I_4 + I_5)} \quad (3.5)$$

**3.3. Método para la obtención de la ecuación de fondo.**

Simulando cinco patrones de franjas ideales con desplazamientos de fase de 90 grados entre ellos y haciendo algunas combinaciones entre sus

intensidades observamos que para una combinación se encuentra una constante ideal igual a 1785 para todos los píxeles de fase valida. La ec. 3.6 es la ecuación de fondo para el algoritmo Hariharan de PSI de la sección 2.4.4.7 y fue calculado a partir de patrones de franjas experimentales y fue comparado con el valor de la constante ideal, entonces seleccionamos un valor de umbral el cual va de 1% a 100% del valor ideal, si el valor experimental es menor que el valor de umbral, ese píxel es de fase no valida y si el valor es más grande que el umbral el píxel es de fase valida, haciendo esto podemos detectar de manera rápida los píxeles de fase valida con muy buena precisión.

$$2I_1 + 3I_2 + 4I_3 + 3I_4 + 2I_5 = 1785 \quad (3.6)$$

El problema aquí es el de seleccionar un valor de umbral para poder hacer la comparación de manera automática ya que el valor de umbral depende de factores dependientes de la prueba como: la irradiancia del LED, la reflectancia del espejo bajo prueba, las lentes objetivo de la CCD y la iluminación de fondo. Para resolver esto probamos diferentes espejos variando la irradiancia del LED y los patrones de franjas obtenidos los analizamos con diferentes valores de umbral con esto encontramos que un buen desenvolvimiento en los bordes de la fase valida depende de una buena selección del valor de umbral. Realizamos aproximadamente 1000 procesos de desenvolvimiento para diferentes espejos y de ellos solo 100 valores de umbral proporcionaban buenos resultados de desenvolvimiento de la fase. Graficamos los 100 valores de umbral contra las intensidades promedio del patrón de franjas y se realiza un ajuste polinomial para seleccionar automáticamente el valor de umbral para cualquier serie de patrones de franjas. Como ejemplo en la Fig. 3.1 se muestra la curva del ajuste tercer orden del promedio de intensidades de un patron de franjas digitalizado y la Eq. 3.7 es usada para obtener de manera automática el valor de umbral para cualquier prueba óptica que use el algoritmo Hariharan.

$$\text{Threshold}(x) = -8.45471 + 0.38168x - 0.0025x^2 + 0.000010055x^3 \quad (3.7)$$

donde el eje x es el promedio de intensidades del patron de franjas y el eje y es el valor deseado del umbral que se usara en la detección de píxeles de fase valida, con la práctica hemos observado que un valor umbral de 20 es óptimo para la mayoría de los casos.

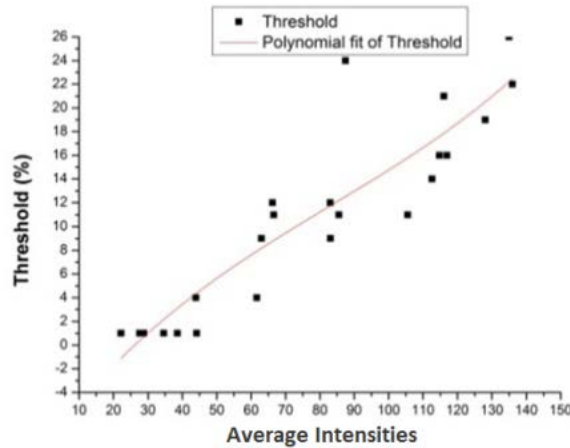


Fig. 3.1. Computando automáticamente el valor de umbral con ajuste polinomial de tercer orden

### 3.4. Ecuaciones de fondo para varios algoritmos

#### PSI.

Siguiendo el mismo método de la sección anterior encontramos las ecuaciones de fondo para los algoritmos PSI de la sección 2.4.3.

#### 3.4.1. Ecuación de fondo para algoritmo PSI de 3 pasos de 0, 90 y 180 grados.

La eq. 3.8 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.2.

$$I_1 + I_3 = 255 \quad (3.8)$$

**3.4.2. Ecuación de fondo para algoritmo PSI de 3 pasos de 0, 120 y 240 grados.**

La eq. 3.9 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.3.

$$I_1 + I_2 + I_3 = 382 \quad (3.9)$$

**3.4.3. Ecuación de fondo para algoritmo PSI de 4 pasos de 0, 90, 180 y 270 grados.**

La eq. 3.10 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.4.

$$S_1 + S_2 + S_3 + S_4 = 510 \quad (3.10)$$

**3.4.4. Ecuación de fondo para algoritmo PSI de 4 pasos de 0, 120, 240 y 360 grados.**

La eq. 3.11 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.5.

$$I_1 + 2I_2 + 2I_3 + I_4 = 766 \quad (3.11)$$

**3.4.5. Ecuación de fondo para algoritmo PSI de 5 pasos 0, 45, 90, 135 y 180 grados.**

La eq. 3.12 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.6.

$$I_1 + I_5 = 255 \quad (3.12)$$

**3.4.6. Ecuación de fondo para algoritmo PSI Hariharan de 5 pasos de 0, 90, 180, 270 y 360.**

La eq. 3.13 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.7.

$$2I_1 + 3I_2 + 4I_3 + 3I_4 + 2I_5 = 1785 \quad (3.13)$$



**3.4.7. Ecuación de fondo para algoritmo PSI de 6 pasos de 0, 90, 180, 240, 360 y 450 grados.**

La eq. 3.14 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.8.

$$I_1 + 2I_3 + I_5 = 510 \quad (3.14)$$

**3.4.8. Ecuación de fondo para algoritmo PSI de 7 pasos de 0, 60, 120, 180, 240, 300 y 360 grados.**

La eq. 3.15 es la ecuación de fondo encontrada para el algoritmo PSI de la sección 2.3.4.9.

$$I_1 + I_2 + I_3 + 2I_4 + I_5 + I_6 + I_7 = 1020 \quad (3.15)$$

### **3.5. Conclusión.**

En este capítulo se explicó que son los píxeles de fase válida y la importancia de saber diferenciarlos de los de fase no válida, por lo que se explicó la ecuación de Schreiber, y se describió como se obtuvieron las ecuaciones de fondo y se muestran las ecuaciones de fondo que se encontraron para algunos algoritmos PSI. Estas ecuaciones de fondo junto con los momentos geométricos son una herramienta poderosa en la detección de píxeles de fase válida cuando se analizan patrones de franjas obtenidos experimentalmente, dando muy buenos resultados.

## *Capítulo 4*

# *4. Pantalla de cristal líquido (LCD).*

### *4.1. Introducción.*

En este capítulo se da una introducción del funcionamiento básico de una LCD, se describen las características de la LCD usada en las pruebas ópticas, se explica la resolución de la LCD para diferentes pruebas ópticas con PSI y finalmente se explican las pruebas ópticas que pueden ser implementadas usando una LCD.

Una pantalla de cristal líquido o *lcd* (sigla del inglés *liquid crystal display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.

### *4.2. Características de la LCD.*

Cada píxel de un LCD típicamente consiste de una capa de moléculas alineadas entre dos electrodos transparentes, y dos filtros de polarización, los ejes de transmisión de cada uno que están (en la mayoría de los casos) perpendiculares entre sí. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada al pasar por el segundo polarizador.

La superficie de los electrodos que están en contacto con los materiales de cristal líquido es tratada a fin de ajustar las moléculas de cristal líquido en una dirección en particular. Este tratamiento suele ser normalmente aplicable en una fina capa de polímero que es unidireccionalmente frotada utilizando, por ejemplo, un paño. La dirección de la alineación de cristal líquido se define por la dirección de frotación.

Antes de la aplicación de un campo eléctrico, la orientación de las moléculas de cristal líquido está determinada por la adaptación a las superficies. En un dispositivo twisted nematic, TN (uno de los dispositivos más comunes entre los de cristal líquido), las direcciones de alineación de la superficie de los dos electrodos son perpendiculares entre sí, y así se organizan las moléculas en una estructura helicoidal, o retorcida. Debido a que el material es de cristal líquido birrefringente, la luz que pasa a través de un filtro polarizante se gira por la hélice de cristal líquido que pasa a través de la capa de cristal líquido, lo que le permite pasar por el segundo filtro polarizado. La mitad de la luz incidente es absorbida por el primer filtro polarizante, pero por lo demás todo el montaje es transparente.

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal (esto se puede resistir gracias a las fuerzas elásticas desde que las moléculas están limitadas a las superficies). Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el píxel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal

líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

El efecto óptico de un dispositivo twisted nematic (TN) en el estado del voltaje es mucho menos dependiente de las variaciones de espesor del dispositivo que en el estado del voltaje de compensación. Debido a esto, estos dispositivos suelen usarse entre polarizadores cruzados de tal manera que parecen brillantes sin tensión (el ojo es mucho más sensible a las variaciones en el estado oscuro que en el brillante). Estos dispositivos también pueden funcionar en paralelo entre polarizadores, en cuyo caso la luz y la oscuridad son estados invertidos. La tensión de compensación en el estado oscuro de esta configuración aparece enrojecida debido a las pequeñas variaciones de espesor en todo el dispositivo. Tanto el material del cristal líquido como el de la capa de alineación contienen compuestos iónicos. Si un campo eléctrico de una determinada polaridad se aplica durante un período prolongado, este material iónico es atraído hacia la superficie y se degrada el rendimiento del dispositivo. Esto se intenta evitar, ya sea mediante la aplicación de una corriente alterna o por inversión de la polaridad del campo eléctrico que está dirigida al dispositivo (la respuesta de la capa de cristal líquido es idéntica, independientemente de la polaridad de los campos aplicados)

Cuando un dispositivo requiere un gran número de píxeles, no es viable conducir cada dispositivo directamente, así cada píxel requiere un número de electrodos independiente. En cambio, la pantalla es multiplexada. En una pantalla multiplexada, los electrodos de la parte lateral de la pantalla se agrupan junto con los cables (normalmente en columnas), y cada grupo tiene su propia fuente de voltaje. Por otro lado, los electrodos también se agrupan (normalmente en filas), en donde cada grupo obtiene una tensión de sumidero. Los grupos se han diseñado de manera que cada píxel tiene una combinación única y dedicada de fuentes y sumideros. Los circuitos

electrónicos o el software que los controla, activa los sumideros en secuencia y controla las fuentes de los píxeles de cada sumidero.

### ***4.3. Pruebas implementadas con la LCD.***

Con la LCD hemos implementado un conjunto de pruebas ópticas usando la pantalla de cristal líquido (LCD). Realizamos con la LCD la prueba de Foucault, la prueba del alambre, la prueba de Hartmann, la prueba de Ronchi y la prueba de Ronchi nula. Aunque estas pruebas han sido estudiadas y desarrolladas desde hace muchos años, estas necesitan diferentes pantallas para que cada prueba sea implementada. Una ventaja de usar una LCD es que con esta, es posible desplegar demasiadas pantallas en la LCD y por lo tanto se pueden desarrollar todas las pruebas mencionadas anteriormente. La otra ventaja es que digitalizando los patrones observados con una cámara CCD es posible analizarlos y obtener resultados cuantitativos en lugar de resultados cualitativos. También se utiliza la prueba de Ronchi como interferómetro de desplazamiento lateral para desarrollar la interferometría de desplazamiento de fase (PSI).

### ***4.4. Resolución de la LCD.***

Las dimensiones horizontal y vertical son expresadas en píxeles. Las pantallas HD tienen una resolución nativa desde 1280x720 píxeles (720p), hasta 1920x1080 píxeles (1080p).

El ancho de punto es la distancia entre los centros de dos píxeles adyacentes. Cuanto menor sea el ancho de punto, tanto menor granularidad

tendrá la imagen. El ancho de punto suele ser el mismo en sentido vertical y horizontal, pero puede ser diferente en algunos casos

Nosotros usamos una LCD con una resolución de 1024X768 píxeles de acuerdo a los datos del fabricante cada píxel mide 13X15  $\mu\text{m}$ , físicamente la LCD mide 3.6X2.5 cm, físicamente el ancho de punto es mayor al ancho de cada píxel dejando un espacio entre píxeles adyacentes, si consideramos que la LCD es ideal no se consideran estos espacios.

Cuando se despliega una rejilla de Ronchi con un periodo de 2 píxeles por franja (1 píxel línea oscura y 1 píxel línea brillante), físicamente tenemos una rejilla con un periodo de 361.2 franjas por pulgada, el cual es considerado como la rejilla de máxima resolución que podemos desplegar en la LCD.

#### **4.4.1. Resolución de la LCD para pruebas con PSI.**

Cuando se despliegan rejillas de Ronchi en la LCD para hacer pruebas ópticas con la técnica de PSI el ancho de franjas no puede ser de 2 porque no se notaría el cambio de fase al hacer el incremento de fase, por ello lo más importante a considerar es el tamaño de paso, puesto que a partir de este se calcula el periodo mínimo que podemos desplegar en la LCD el cual es calculado con la ec. 4.1.

$$PF = \frac{360}{dfr} \quad (4.1)$$

donde PF es el número de píxeles por franja de la rejilla de Ronchi y dfr es el tamaño de paso entre rejillas de Ronchi desplegadas.

Para pruebas donde el tamaño de paso es de  $90^\circ$  el valor del periodo  $PF=4$  el cual es equivalente a 180.6 franjas por pulgada, este nos indica la máxima resolución que podemos desplegar en la LCD, para pruebas de PSI con este incremento de fase se pueden desplegar rejillas de diferentes periodos siempre y cuando los valores de PF sean múltiplos de 4.

Para pruebas donde el tamaño de paso es de 120° el valor del periodo PF=3 el cual es equivalente a 240.8 franjas por pulgada, este nos indica la máxima resolución que podemos desplegar en la LCD, para pruebas de PSI con este incremento de fase se pueden desplegar rejillas de diferentes periodos siempre y cuando los valores de PF sean múltiplos de 3.

Para pruebas donde el tamaño de paso es de 45° el valor del periodo PF=8 el cual es equivalente a 90.3 franjas por pulgada, este nos indica la máxima resolución que podemos desplegar en la LCD, para pruebas de PSI con este incremento de fase se pueden desplegar rejillas de diferentes periodos siempre y cuando los valores de PF sean múltiplos de 8.

Para pruebas donde el tamaño de paso es de 60° el valor del periodo PF=6 el cual es equivalente a 120.4 franjas por pulgada, este nos indica la máxima resolución que podemos desplegar en la LCD, para pruebas de PSI con este incremento de fase se pueden desplegar rejillas de diferentes periodos siempre y cuando los valores de PF sean múltiplos de 6.

También hay que tomar en cuenta que para obtener un buen contraste en las franjas el periodo de la rejilla desplegada depende de parámetros físicos de la prueba, para evitar el fenómeno de difracción [33] el periodo de la rejilla se calcula con la ec. 4.2.

$$d \leq 10\lambda \frac{R}{D} \left[ 1 + \left( 1 + \frac{\Delta R}{25\lambda \left( \frac{R}{D} \right)^2} \right)^{\frac{1}{2}} \right] \quad (4.2)$$

Donde R es el radio de curvatura de la superficie bajo prueba, D es el diámetro,  $\Delta R$  es la diferencia entre el radio de curvatura máximo y mínimo y  $\lambda$  es la longitud de onda.

## ***4.5. Conclusión.***

En este capítulo se explicó brevemente el funcionamiento básico de la LCD, se dieron sus principales características, se dieron las pruebas que se pueden implementar con la LCD, se explicó la resolución de la LCD y como obtener los periodos por franja mínimos para diferentes incrementos de paso cuando se usa la técnica PSI. Con la LCD se evita el uso de varias rejillas de Ronchi costosas y vuelve la prueba más versátil ya que se pueden desplegar rejillas de Ronchi con diferente paso de fase.



## *Capítulo 5*

# *5. Arreglos experimentales.*

### *5.1. Introducción.*

En este capítulo se describen los arreglos experimentales usados para probar superficies ópticas, se describe la prueba de Ronchi implementada con la técnica de PSI con motores a pasos y se describe el arreglo experimental de la configuración de la prueba de Ronchi con la LCD para implementar la prueba de Foucault, la prueba del alambre, la prueba de Hartmann, la prueba de Ronchi con rejilla bironchi, con rejilla circular e incluso con rejilla de Ronchi nula.

### *5.2. Prueba de Ronchi con motores a pasos.*

El arreglo experimental de la Fig. 5.1 consiste de un espejo o lente bajo prueba, un LED pulido hacia el ánodo y cátodo para tener un ángulo de divergencia más grande de la fuente de iluminación, una rejilla de Ronchi un colocada en una montura que tiene 3 motores a pasos, los cuales tienen una resolución angular de 200 pasos/revolución y son controlados desde la PC.

La montura tiene 3 movimientos, uno en el eje X y otro en el eje Y para obtener de manera independiente la aberración transversal y de ahí el frente de onda de la superficie bajo prueba. El eje Z es usado para el defoco del sistema óptico. Los motores a pasos mueven unos micrómetros que tienen

una longitud de desplazamiento de 4-20 mm, con capacidades de carga axial de 40-450 N y una resolución de 1  $\mu\text{m}$ .

Usamos una montura para poder intercambiar los LEDs que se usan como fuente de iluminación los cuales tienen longitudes de onda de 452.98nm, 478.21nm, 564.98nm, 589.31nm, 625.89nm, 689.77nm y 933.21nm, todo esto se hace usando una fuente de poder pequeña y un resistor variable, para incrementar o reducir la intensidad del LED usado.

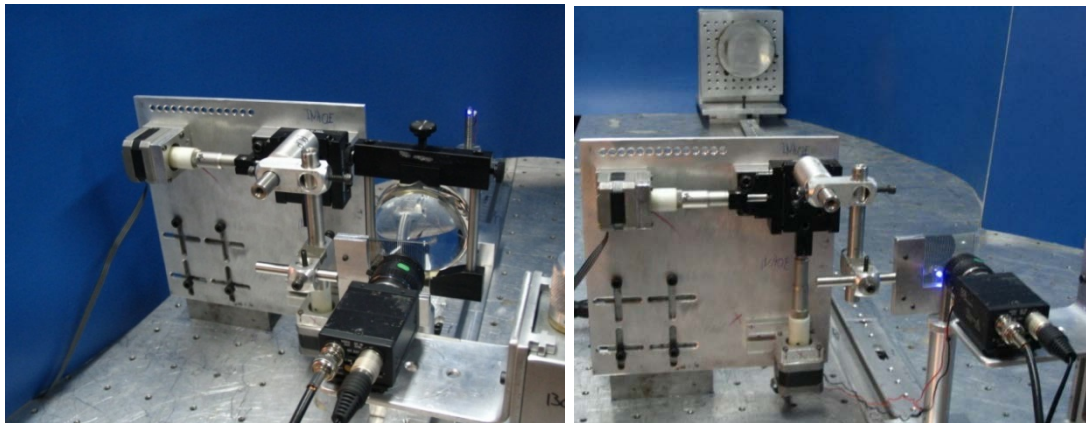


Fig. 5.1. Arreglos experimentales de la prueba de Ronchi para probar: a) lentes, b) espejos.

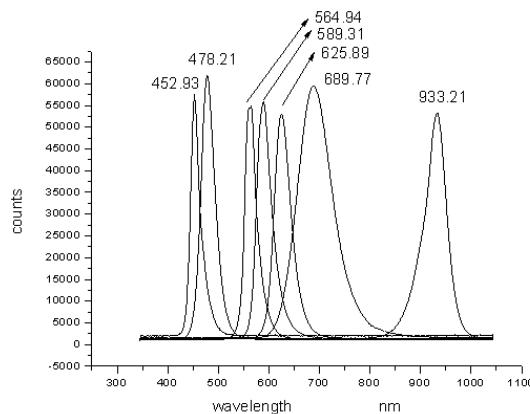


Fig. 5.2. Picos espectrales de los LEDs usados como fuente de iluminación.

### 5.2.1. Fuente de poder para la fuente de luz.

Cuando se usan LEDs a los que se les puede controlar su intensidad se tiene la ventaja de que se pueden probar superficies rugosas, superficies

aluminadas, superficies en proceso de pulido o incluso superficies líquidas [31] la fuente de poder tiene la ventaja porque fue diseñada para corregir muy bien la corriente que alimenta al LED evitando variaciones en la intensidad. Cuando se prueban espejos cóncavos la manera más adecuada de poner la fuente de iluminación y el detector es colocarlos lo más cerca posible en el radio de curvatura, haciendo esto observamos que para focos cortos el ángulo de iluminación debe ser grande, esto significa que debemos tener ángulos de divergencia grande para poder iluminar focos cortos.

Con el objetivo de eliminar factores que introducen errores en el resultado final, mejoramos nuestro dispositivo. En una superficie pulida no siempre da buenos resultados trabajando con la luz visible, pero en algunas ocasiones cuando uno prueba superficies rugosas se debe usar una fuente de iluminación en el infrarrojo. Un factor analizado fue la iluminación, usamos varios LEDs de diferentes longitudes de onda los cuales fueron medidos experimentalmente. Otro factor analizado fue el  $f/\#$  en este caso es importante la divergencia y la intensidad de la fuente de luz, antes que nada calculamos la divergencia la cual tiene un rango entre  $20^\circ$  y  $140^\circ$  más grande que un lecho común. Cada le fue pulido hasta llegar cerca de su cátodo para obtener un mejor ángulo de divergencia, lo cual también nos asegura que la iluminación es uniforme y cerca de 1W. Con estas modificaciones iluminamos totalmente el área de sistemas ópticos convergentes rápidos y lentos. Para evitar las variaciones en la irradiancia se diseñó y se construyó una fuente de poder la cual alimenta al LED manteniendo constantes la corriente eléctrica esta fuente usa corriente alterna (AC) y consiste de un transformador de voltaje (T1), un rectificador completo (D3), y un regulador ajustable (U1) el cual provee de voltaje variable al diodo (D4). El nivel de irradiancia se regula con (R4) y le da estabilidad en voltaje a la fuente de poder. La Fig. 5.3 muestra el Layout de la fuente de poder.

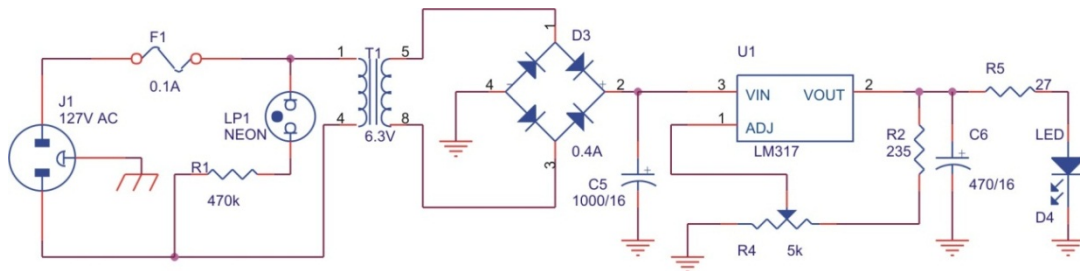


Fig. 5.3. Layout de la fuente de poder

### 5.3. Arreglo experimental con la LCD.

El arreglo experimental para poner en práctica la prueba del filo de la navaja, la prueba del alambre, la prueba de Ronchi, la prueba bironchi, la prueba de Ronchi nula y la prueba de Hartmann con ayuda de la pantalla LCD se muestra en la Fig. 5.4 este arreglo consiste de un espejo o lente a prueba, una pantalla LCD que nos permite cambiar instantáneamente entre diferentes rejillas y de forma automática el periodo de la rejilla Ronchi, se usa el mismo conjunto de LEDs intercambiables de la sección anterior como fuente de iluminación ya que no utilizan ningún componente óptico adicional evitando así aberraciones adicionales durante la prueba. Los LEDs son pulidos y se usa la misma fuente de poder mencionada en la sección 5.2.1. Los Ronchigramas se digitalizaron con una cámara monocromática CCD de 640x480 píxeles.

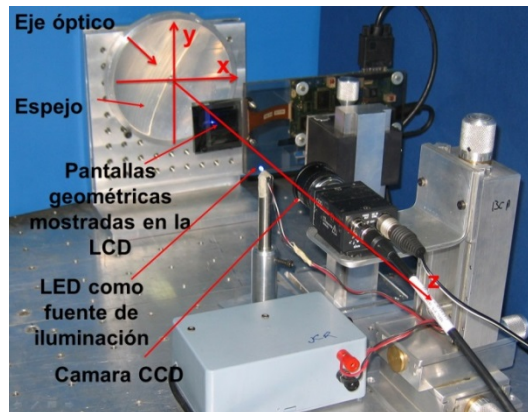


Fig. 5.4. Arreglo experimental de la prueba de Ronchi con la LCD.

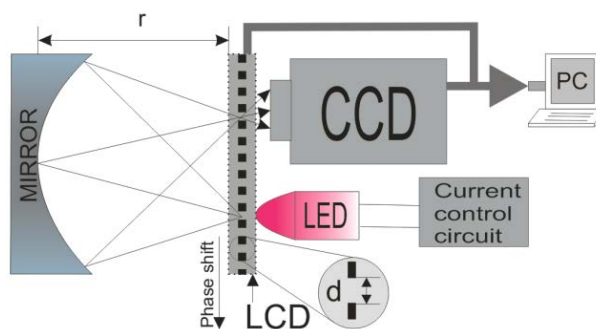


Fig. 5.5. Diagrama del dispositivo implementado con la LCD

#### 5.4. Prueba de Foucault.

El método filo de la navaja, introducido por Foucault [4] ha demostrado ser muy útil para probar las superficies ópticas. Los rayos pueden ser movidos de sus trayectorias esperadas por las aberraciones ópticas, por difracción o por una deformación, superficie sin terminar. La idea básica es la detección de desplazamientos laterales de los rayos mediante el bloqueo de los rayos o la modificación de estos desplazados. El bloqueo o la modificación se puede lograr mediante la colocación de pantallas en cualquiera de los planos de convergencia de la luz que pasa a través de, o siendo reflejada desde la superficie óptica bajo prueba. Las principales ventajas son su alta sensibilidad y su simplicidad tanto en el aparato y en la interpretación cualitativa. En general, este examen se utiliza para detectar la presencia de aberraciones transversales, esto se hace mostrando una amplia línea de negro en la pantalla LCD. Simulando el filo de la cuchilla que está bloqueando una parte de un plano transversal por los rayos o luz difractada para que aparezca una sombra sobre la región aberrada.

De acuerdo a la teoría geométrica para esta prueba el borde de la navaja en esta prueba es colocado a una distancia  $r_1$  de la intersección del rayo principal (el origen del plano  $X_1$ - $Y_1$ ), y el ángulo  $\Phi_1$  esta subtendido entre el eje  $Y_1$  y el filo de la navaja como se muestra en la Fig. 5.6. El ángulo  $\Phi_1$  se

define como positivo si la inclinación del filo de la navaja es positiva. La siguiente ecuación define el borde como:

$$x_1 \cos \phi_1 - y_1 \operatorname{sen} \phi_1 = r_1 \quad (5.1)$$

La transmitancia sobre este plano se expresa como:

$$T(x_1, y_1) = \begin{cases} 1 & \text{if } x_1 \cos \phi_1 - y_1 \operatorname{sen} \phi_1 < r_1 \\ 0 & \text{if } x_1 \cos \phi_1 - y_1 \operatorname{sen} \phi_1 > r_1 \end{cases} \quad (5.2)$$

$X_1$ - $Y_1$  definen el plano paraxial de convergencia del frente de onda ideal  $W$ , cualquier punto  $(x_1, y_1)$  sobre este plano satisface aproximadamente la siguiente propiedad.

$$(x_1, y_1) = \left( -R \frac{\partial W}{\partial x}, -R \frac{\partial W}{\partial y} \right) \quad (5.3)$$

donde  $R$  es la distancia entre el plano  $X=Y$  y el plano  $X_1=Y_1$ . Por lo tanto la función de transmitancia queda de la siguiente forma

$$T\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right) = \begin{cases} 1 & \text{if } -\frac{\partial W}{\partial x} \cos \phi_1 + \frac{\partial W}{\partial y} \operatorname{sen} \phi_1 < \frac{r_1}{R} \\ 0 & \text{if } -\frac{\partial W}{\partial x} \cos \phi_1 + \frac{\partial W}{\partial y} \operatorname{sen} \phi_1 > \frac{r_1}{R} \end{cases} \quad (5.4)$$

Para el caso particular de aberraciones con simetría rotacional, cualquier eje sobre el plano  $X_1$ - $Y_1$  puede ser usado como referencia para especificar la posición del filo de la navaja. Si se escoge el eje  $X$  para que  $\Phi_1=90^\circ$  La función de transmitancia se reduce a:

$$T\left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}\right) = \begin{cases} 1 & \text{if } \frac{\partial W}{\partial y} < \frac{r_1}{R} \\ 0 & \text{if } \frac{\partial W}{\partial y} > \frac{r_1}{R} \end{cases} \quad (5.5)$$

Esta prueba se puede implementar con el arreglo experimental de la prueba de Ronchi sin utilizar ningún elemento óptico adicional, desplegando el filo de la navaja desde la pantalla LCD.

### 5.5. Prueba del alambre.

Prueba de Foucault es muy sensible para la detección de errores zonales, sin embargo la prueba de Foucault es menos sensible en la medición de los errores. Ritchey [28] sugiere el uso de una pantalla con ranuras zonales sobre cualquier superficie esférica, al comprobar el espejo junto al método de Foucault. De esta manera es posible calcular la cantidad de aberración rayos transversal de las dos ranuras de la zona, que se localizan en lados opuestos del eje óptico, para comprobar si las regiones detrás de las ranuras son regulares y para que coincida entre ellos. Esos espacios se muestran en la pantalla LCD de la figura.

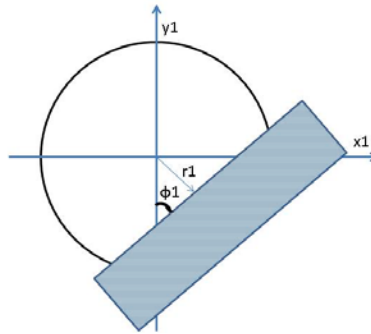


Fig. 5.6. Posición del filo de navaja proyectado en el plano de la pupila de entrada.

Si la posición de la parte central del alambre es denotada por  $r_1$  y el ancho del alambre por  $2\delta r$ , la amplitud de la transmitancia en el plano paraxial está dado por

$$T(x_1, y_1) = \begin{cases} 1 & \text{if } |x_1 \cos \phi_1 - y_1 \sin \phi_1| < \delta r \\ 0 & \text{if } |x_1 \cos \phi_1 - y_1 \sin \phi_1| > \delta r \end{cases} \quad (5.6)$$

El centro del contorno oscuro obtenido con la prueba del alambre coincide con el contorno entre las regiones oscuras y brillantes del patrón de sombras obtenido con la prueba de Foucault. El alambre no cubre la misma extensión de la navaja por lo que las áreas oscuras son reducidas a líneas oscuras a lo largo de las posiciones del borde.

### ***5.6. Prueba de Hartmann.***

La prueba de Hartmann [29] es una prueba de pantalla para la examinación de la calidad de un sistema óptico. Esta prueba necesita una pantalla, placa de Hartmann, que tiene una serie de agujeros en ella en la pupila de salida del sistema óptico bajo prueba. Localizamos el LCD justo en frente del sistema óptico. Las manchas correspondientes a estos agujeros se registran en un CCD, antes o después de foco. Las posiciones de estos puntos se asignan de nuevo en el elemento óptico, y la diferencia entre las posiciones de estos puntos y los agujeros de la pantalla se utilizan para calcular las aberraciones.

La prueba de Hartmann, la prueba de Ronchi y la interferometría de desplazamiento lateral miden la derivada del frente de onda en lugar de la diferencia de camino óptico OPD como los interferómetros Twyman-Green y Fizeau. Una deformación grande del frente de onda produce cambios en la derivada del frente de onda si la extensión de esta deformación es larga. De manera análoga pequeñas deformaciones en el frente de onda producen grandes cambios en la derivada del frente de onda si la extensión de la deformación es corta.

### ***5.7. Prueba de Ronchi con rejilla bironchi, circular y nula.***

Cuando una cuadrícula [29,30] se visualiza en la pantalla LCD en la configuración de la figura. La rejilla permite procesar la aberración transversal tanto de la direcciones X e Y con una sola serie de ronchigramas. También en lugar de líneas rectas o rejillas cuadradas, es posible utilizar curvas



circulares. Patrones con este tipo de decisión se diferencian de los patrones normales de Ronchi.

También rejillas de Ronchi nulas [31] se pueden desplegar en la pantalla LCD para la prueba de calidad de los sistemas ópticos. Esta prueba necesita una pantalla de Ronchi nula de que debe ser calculada antes de que se realice la prueba.

## **5.8. Conclusión.**

En este capítulo se describieron los dispositivos usados para probar sistemas ópticos convergentes, usando los métodos mencionados en capítulos anteriores para el análisis cualitativo y cuantitativo de los sistemas ópticos bajo prueba.

Se explicó la prueba de Ronchi implementada con la técnica de PSI con motores a pasos y se describe el arreglo experimental de la configuración de la prueba de Ronchi con la LCD para implementar la prueba de Foucault, la prueba del alambre, la prueba de Hartmann, la prueba de Ronchi con rejilla bironchi, con rejilla circular e incluso con rejilla de Ronchi nula, todas estas pruebas se pueden implementar sin usar ningún elemento óptico adicional.

# Capítulo 6

## 6. Software desarrollado.

### 6.1. Introducción.

En este capítulo se describen los programas desarrollados para la obtención de patrones de franjas, el primero es el de control del movimiento de los motores a pasos, el segundo es para la implementación de la técnica PSI con la LCD y el tercero para el análisis de los patrones obtenidos.

### 6.2. Software para mover los motores a pasos.

Se implementó un software en Visual Basic para controlar el movimiento de los motores a pasos en las direcciones vertical y horizontal para obtener el desplazamiento de fase deseado. La ventana principal del software se muestra en la Fig. 6.1.

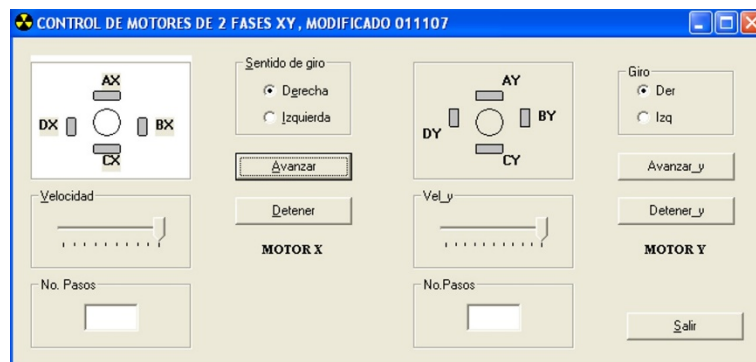


Fig. 6.1. Ventana principal del software usado para mover los motores a pasos.

### 6.3. Software para desplegar rejillas en la LCD.

En la Fig. 6.2 Se muestra la ventana principal del programa “LCD GRATINGS” el cual desarrollamos y usamos para desplegar cualquier tipo de rejilla en la LCD de manera automática e instantánea y para dar los desplazamientos de fase deseados sin mover un solo componente óptico.

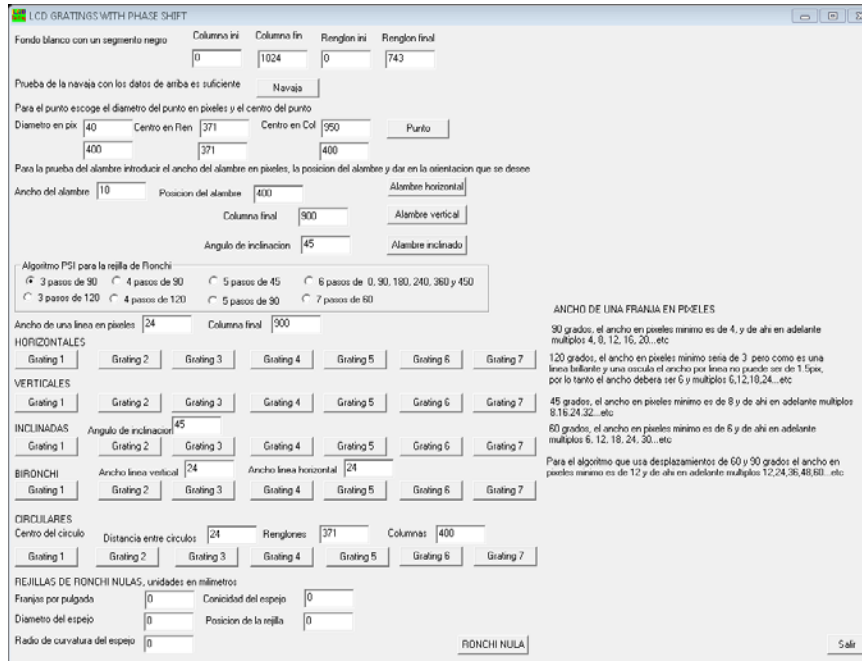


Fig. 6.2. Ventana principal del software usado para la creación de rejillas.

La parte superior del programa donde ponemos los valores para simular el filo de la navaja desplegado en la LCD se ve en la Fig. 6.3. en esta parte podemos poner una columna negra de ancho variable colocando un valor inicial de columna y un valor final de columna, para simular el filo de la navaja con orientación vertical de la misma manera si queremos colocar renglones negros ingresamos el valor del renglón inicial y del renglón final.

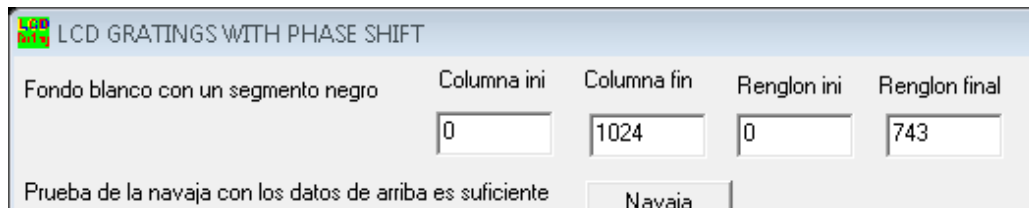


Fig. 6.3. Ingreso de datos para la creación de la rejilla de la prueba de la navaja.

El código en visual c++ 6.0 para esta parte se muestra a continuación.

```
void CLCDGRATINGSDlg::OnButton1()
{
    int i,j,aa;
    CClientDC dc(this);
    for (i = 0; i < 743; i++)
    {
        for (j = 0; j < 1024; j++)
        {
            aa = 255;
            if (i >= m_renini && i <= m_renfin && j >= m_colini && j <= m_colfin)
            {
                aa = 0;
            }
            dc.SetPixel(j, i, RGB(aa,aa,aa));
        }
    }
}
```

En la Fig. 6.4. Se muestra que con esta parte del programa se pueden simular navajas verticales en cualquier posición de la LCD y del ancho que uno prefiera, evitándose el tener que estar moviendo una navaja manualmente. En la Fig. 6.4. a) se muestra una navaja vertical que inicia en el píxel 50 y termina en el píxel 100, en la Fig. 6.4. b) se muestra una navaja vertical que inicia en el píxel 100 y termina en el píxel 200, en la Fig. 6.4. c) se muestra una navaja vertical que inicia en el píxel 200 y termina en el píxel 400 y en la Fig. 6.4. d) se muestra una navaja vertical que inicia en el píxel 400 y termina en el píxel 800.

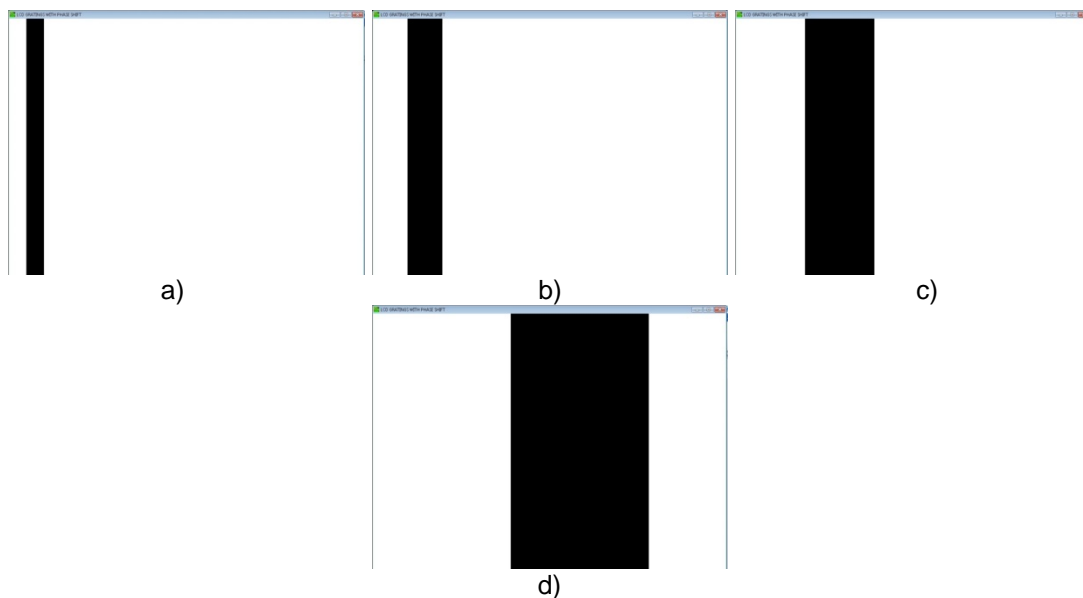


Fig. 6.4. a, b, c y d) Filos de navaja verticales desplegados en la LCD con posiciones y anchos variables.

En la Fig. 6.5. Se muestra que con la misma parte del programa se pueden simular navajas horizontales en cualquier posición de la LCD y del ancho que uno prefiera. En la Fig. 6.5. a) se muestra una navaja horizontal que inicia en el píxel 50 y termina en el píxel 100, en la Fig. 6.5. b) se muestra una navaja horizontal que inicia en el píxel 100 y termina en el píxel 200, en la Fig. 6.5. c) se muestra una navaja horizontal que inicia en el píxel 200 y termina en el píxel 400 y en la Fig. 6.5. b) se muestra una navaja horizontal que inicia en el píxel 400 y terminaría en el píxel 800, pero como la LCD solo es de 768 y 25 píxeles los utiliza el título de la ventana entonces termina en el píxel 743.



**Fig. 6.5. a, b, c y d) Filos de navaja desplegados en la LCD con posiciones y anchos variables horizontalmente.**

La parte del programa que se muestra en la Fig. 6.6. es donde ponemos los valores para simular dos círculos desplegados en la LCD. En esta parte podemos poner círculos con diametro de hasta un píxel para simular una prueba de difracción por punto, el otro círculo de diametro más grande es para que por ahí pase la luz que va hacia la CCD, se da la opción de cambiar el diametro de los círculos y el centro de ellos en reglones y columnas ya que cuando se trabaja en el laboratorio es más práctico mover las aberturas

desde la PC utilizando este programa en lugar de mover las monturas mecánicas.

Fig. 6.6. Ingreso de datos para dos círculos.

El código en visual c++ 6.0 para los dos círculos se muestra a continuación

```
void CLCDGRATINGSDlg::Onpunto()
{
    int i,j,aa;
    double radpix, radpix2;
    CClientDC dc(this);
    m_colfin = 1024;
    UpdateData(TRUE);

    for (i = 0; i < 743; i++)
        {
            for (j = 0; j < 1024; j++)
                {
                    aa = 255;

                    if (i >= m_renini && i <= m_renfin && j >= m_colini && j <= m_colfin)
                        {
                            aa = 0;
                        }

                    radpix = (i-m_cenren)*(i-m_cenren) + (j-m_cencol)*(j-m_cencol);
                    if (radpix <= (m_radpix/2)*(m_radpix/2))
                        {
                            aa = 255;
                        }

                    radpix2 = (i-m_cenren2)*(i-m_cenren2) + (j-m_cencol2)*(j-m_cencol2);
                    if (radpix2 <= (m_radpix2/2)*(m_radpix2/2))
                        {
                            aa = 255;
                        }

                    dc.SetPixel(j, i, RGB(aa,aa,aa));
                }
        }
}
```

En la Fig. 6.7. a) Se muestran dos círculos uno con un diámetro de 40 píxeles y centrado en el renglón 371 y en la columna 900 el otro círculo tiene un diámetro de 400 píxeles y está centrado en el renglón 371 y en la columna 400. En la Fig. 6.7. b) Se muestran dos círculos uno con un diámetro de 300 píxeles y centrado en el renglón 371 y en la columna 700 el otro círculo tiene un diámetro de 20 píxeles y está centrado en el renglón 371 y en la columna 100.



Fig. 6.7. a, b) Círculos desplegados en la LCD con diámetro y posición variable.

La parte del programa que se muestra en la Fig. 6.8. es donde ponemos los valores para simular un alambre desplegado en la LCD. En esta parte podemos poner alambre de hasta un píxel para simular la prueba del alambre, pudiendo simular alambres horizontales, verticales y con cualquier ángulo de inclinación del ancho que uno prefiera o se necesite para la prueba. Para simular esta prueba solo es necesario ingresar el ancho del alambre en píxeles y poner el valor donde iniciara el alambre. La parte que dice columna final se colocó solo por si se tiene que dejar pasar la luz de la fuente de iluminación (LED) ese valor nos dejara píxeles en blanco a partir de el.

Para la prueba del alambre introducir el ancho del alambre en pixeles, la posición del alambre y dar en la orientación que se desee

Ancho del alambre  Posición del alambre

Columna final

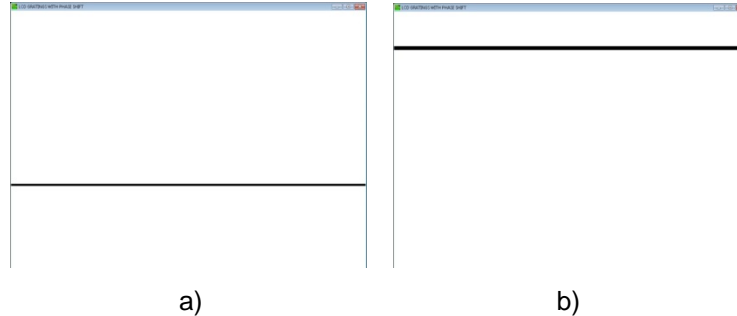
Angulo de inclinacion

Fig. 6.8. Ingreso de datos para la prueba del alambre.

Para simular el alambre horizontal se usa el botón que dice alambre horizontal el cual su código en visual c++ 6.0 es el siguiente.

```
void CLCDGRATINGSDLg::OnButton3()
{
int i,j,aa;
CClientDC dc(this);
for (i = 0; i < 743; i++)
    {
        for (j = 0; j < 1024; j++)
            {
                aa = 255;
                if ( j >= m_posala && j <= m_posala+m_ancala)
                    {
                        aa = 0;
                    }
                dc.SetPixel(j, i, RGB(aa,aa,aa));
            }
    }
}
```

En la Fig. 6.9. a) Se muestran un alambre horizontal con 5 píxeles de grosor que inicial en el renglón 500. En la Fig. 6.9. b) Se muestran un alambre horizontal con 10 píxeles de grosor que inicial en el renglón 100.

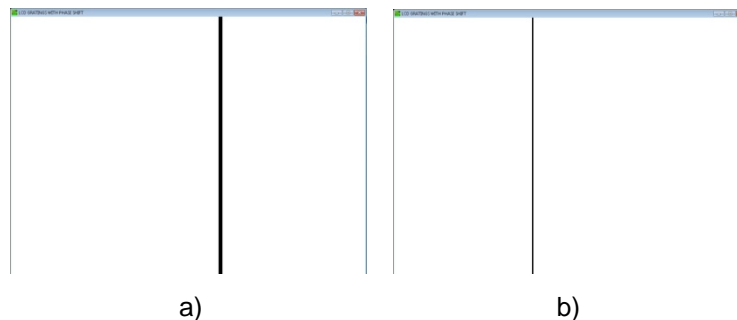


**Fig. 6.9. a, b) Alambres horizontales desplegados en la LCD.**

Para simular el alambre vertical se usa el botón que dice alambre vertical del cual su código en visual c++ 6.0 es el siguiente.

```
void CLCDGRATINGSDlg::OnButton4()
{
    int i,j,aa;
    CClientDC dc(this);
    for (i = 0; i < 743; i++)
    {
        for (j = 0; j < 1024; j++)
        {
            aa = 255;
            if ( i >= m_posala && i <= m_posala+m_ancala && j<=m_colfin1)
            {
                aa = 0;
            }
            dc.SetPixel(j, i, RGB(aa,aa,aa));
        }
    }
}
```

En la Fig. 6.10. a) Se muestra un alambre vertical con 10 píxeles de grosor que inicial en la columna 600. En la Fig. 6.10. b) Se muestra un alambre vertical con 3 píxeles de grosor que inicial en el renglón 400.



**Fig. 6.10. Alambres verticales desplegados en la LCD.**



Para simular el alambre inclinado se usa el botón que dice alambre inclinado del cual su código en visual c++ 6.0 es el siguiente.

```
void CLCDGRATINGSDlg::OnButton5()
{
int i,j,k,aa;
double teta1, tt, ah, cons1;
CClientDC dc(this);
teta1 = m_anginc * pi / 180;
tt = tan(teta1);
ah = m_ancala / cos(teta1);
for (j = 0; j < 1024; j++)
    {
        for (i = 0; i < 743; i++)
            {
                aa = 255;
                cons1 = ceil((i - (j-m_posala)*tt)/ah);
                if(j < m_colfin1)
                    {
                        if(cons1 == -1)
                            {aa = 0;}
                    }
                dc.SetPixel(j, i, RGB(aa,aa,aa));
            }
    }
}
```

En la Fig. 6.11. a) Se muestra un alambre inclinado con 10 píxeles de grosor que inicial en la columna 0 y tiene un ángulo de inclinación de 45 grados. En la Fig. 6.11. b) Se muestra un alambre inclinado con 15 píxeles de grosor que inicial en la columna 200 y tiene un ángulo de inclinación de 60 grados. En la Fig. 6.11. c) Se muestra un alambre inclinado con 5 píxeles de grosor que inicial en la columna 200 y tiene un ángulo de inclinación de 30 grados.

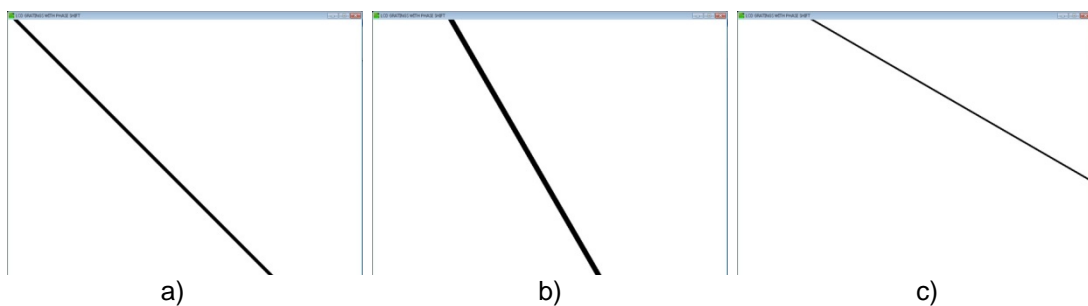


Fig. 6.11. Alambres inclinados desplegados en la LCD.

La parte del programa que se muestra en la Fig. 6.12. es donde seleccionamos el algoritmo PSI con el que se crearan las rejillas y es donde se pone el ancho de línea en píxeles que es igual que el periodo de la rejilla en la Tabla 6.1 se muestran algunas equivalencias de los periodos mas

usados en las rejillas, la columna final es para dejar una línea blanca para dejar pasar la luz de la fuente de iluminación.

Fig. 6.12. Selección de algoritmo PSI, e ingreso de ancho de línea en píxeles.

Tabla 6.1. Equivalencia de los periodos de las rejillas en anchos de línea en píxeles

píxeles / línea	píxeles / franja	franjas / pulgada
1	2	361.243931
2	4	180.621965
3	6	120.414644
4	8	90.3109827
6	12	60.2073218
8	16	45.1554913
12	24	30.1036609
16	32	22.5777457
20	40	18.0621965
24	48	15.0518304

La parte del programa que se muestra en la Fig. 6.13. es donde estan los botones para desplegar rejillas en la LCD. En esta parte podemos desplegar rejillas. Rejillas de Ronchi horizontales, rejillas de Ronchi verticales, Rejillas de Ronchi Inclinadas o rejillas bironchi para estas se ingresan los anchos de línea vertical y horizontal.

Fig. 6.13. Botones para desplegar las rejillas seleccionadas.

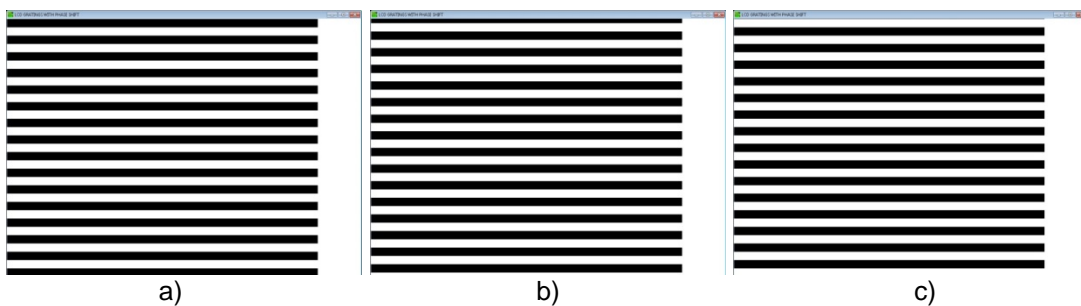
El código del primer botón para la generación de rejillas horizontales con el método seleccionado se muestra a continuación, cabe mencionar que cada botón de rejilla horizontal lleva un código similar, lo que cambia es la variable del desplazamiento de fase.

```

{
int i,j,k,aa;
double cons1, cons2, cons3;
CClientDC dc(this);
cons2 = 1024/(2*m_perepi);
if(m_iPsi == 0)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 1)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 2)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 3)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 4)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 5)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 6)
{cons1 = 0*m_perepi/180;}
if(m_iPsi == 7)
{cons1 = 0*m_perepi/180;}
for (i = 0; i < 743; i++)
    {
        for (j = 0; j < 1024; j++)
            {
                aa = 255;
                cons3 = ceil((i + cons1)/m_perepi);
                if (j < m_colfin2)
                    {
                        for (k = 0; k < cons2; k++)
                            {
                                if(cons3 == 2*k+1)
                                    {aa = 0;}
                            }
                    }
                dc.SetPixel(j, i, RGB(aa,aa,aa));
            }
    }
}

```

En la Fig. 6.14. a, b, c, d y e) Se muestran 5 rejillas de Ronchi horizontales con un periodo de 24 píxeles por franja y un desplazamiento de fase entre ellas de 90 grados.



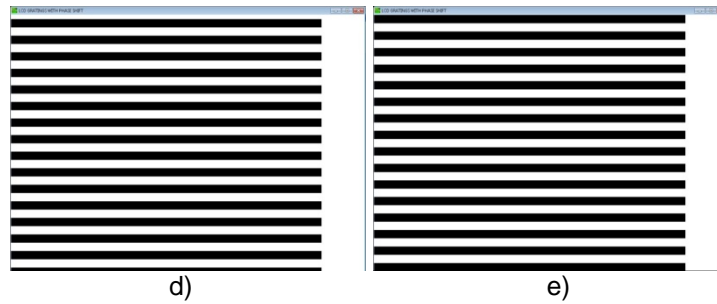


Fig. 6.14. Rejillas de Ronchi horizontales con desplazamientos de fase de 90 grados.

El código del primer botón para la generación de rejillas verticales con el método seleccionado se muestra a continuación, cabe mencionar que cada botón de rejilla vertical lleva un código similar, lo que cambia es la variable del desplazamiento de fase.

```
void CLCDGRATINGSdIg::OnButton13()
{
    int i,j,k,aa;
    double cons1, cons2, cons3;
    CClientDC dc(this);
    cons2 = 1024/(2*m_perepi);
    if(m_iPsi == 0)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 1)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 2)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 3)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 4)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 5)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 6)
    {cons1 = 0*m_perepi/180;}
    if(m_iPsi == 7)
    {cons1 = 0*m_perepi/180;}
    for (i = 0; i < 743; i++)
    {
        for (j = 0; j < 1024; j++)
        {
            aa = 255;
            cons3 = ceil((j + cons1)/m_perepi);
            if(j < m_colfin2)
            {
                for (k = 0; k < cons2; k++)
                {
                    if(cons3 == 2*k+1)
                    {aa = 0;}
                }
            }
            dc.SetPixel(j, i, RGB(aa,aa,aa));
        }
    }
}
```

En la Fig. 6.15. a, b, c y d ) Se muestran 4 rejillas de Ronchi horizontales con un periodo de 48 píxeles por franja y un desplazamiento de fase entre ellas de 120 grados.

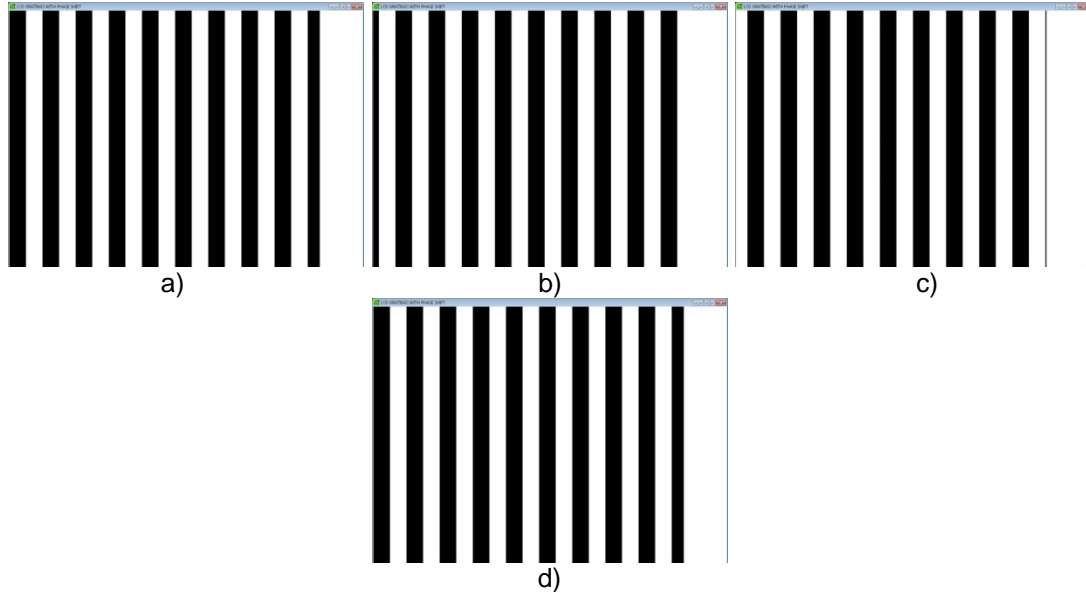


Fig. 6.15. Rejillas de Ronchi horizontales con desplazamientos de fase de 120 grados.

El código del primer botón para la generación de rejillas inclinadas con el método seleccionado se muestra a continuación, cabe mencionar que cada botón de rejilla inclinada lleva un código similar, lo que cambia es la variable del desplazamiento de fase.

```
void CLCDGRATINGSDLg::OnButton20()
{
int i,j,k,aa;
double cons1, cons2, cons3;
double teta1, tt, ah;
CClientDC dc(this);
teta1 = m_anginc2 * pi / 180;
tt = tan(teta1);
ah = m_perepi / cos(teta1);
cons2 = 1024/(m_perepi);
if(m_iPsi == 0)
{cons1 = 0*ah/180;}
if(m_iPsi == 1)
{cons1 = 0*ah/180;}
if(m_iPsi == 2)
{cons1 = 0*ah/180;}
if(m_iPsi == 3)
{cons1 = 0*ah/180;}
if(m_iPsi == 4)
{cons1 = 0*ah/180;}
if(m_iPsi == 5)
{cons1 = 0*ah/180;}
if(m_iPsi == 6)
{cons1 = 0*ah/180;}
}
```

```

if(m_iPsi == 7)
{cons1 = 0*ah/180;}
for (j = 0; j < 1024; j++)
{
    for (i = 0; i < 743; i++)
    {
        aa = 255;
        cons3 = ceil((i + cons1 + j*tt)/ah);
        if(j < m_colfin2)
        {
            for (k = 0; k < cons2; k++)
            {
                if(cons3 == 2*k+1)
                {aa = 0;}
            }
        }
        dc.SetPixel(j, i, RGB(aa,aa,aa));
    }
}
    
```

En la Fig. 6.1. a, b, c, d y e ) Se muestran 5 rejillas de Ronchi inclinadas con un periodo de 96 píxeles por franja y un desplazamiento de fase entre ellas de 120 grados.

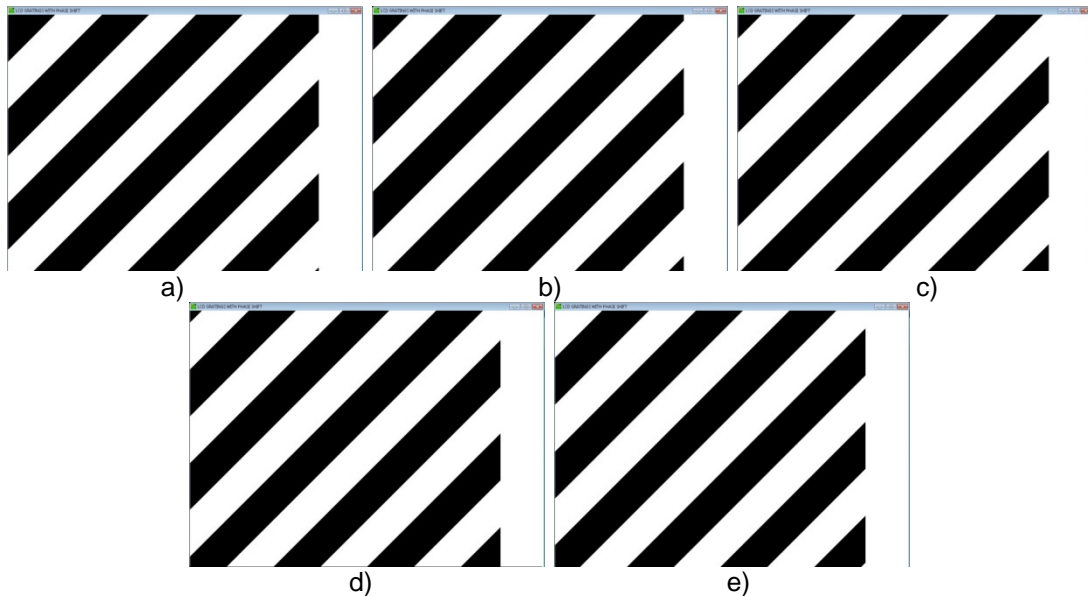


Fig. 6.16. Rejillas de Ronchi inclinadas con desplazamientos de fase de 45 grados.

El código del primer botón para la generación de rejillas bironchi con el método seleccionado se muestra a continuación, cabe mencionar que cada botón de rejilla bironchi lleva un código similar, lo que cambia es la variable del desplazamiento de fase.

```

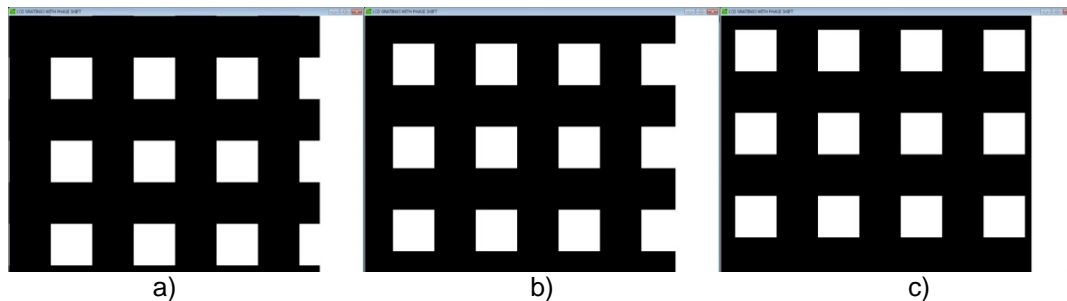
void CLCDGRATINGSDIg::OnButton27()
{
    int i,j,k,aa;
    
```

```

double cons1h, cons2, cons3, cons1v, cons4;
CCClientDC dc(this);
cons2 = 1024/(2*m_anclinh);
if(m_iPsi == 0)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 1)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 2)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 3)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 4)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 5)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 6)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
if(m_iPsi == 7)
{cons1h = 0*m_anclinh/180;
cons1v = 0*m_anclinv/180;}
for (j = 0; j < 1024; j++)
{
    for (i = 0; i < 743; i++)
    {
        aa = 255;
        cons3 = ceil((i + cons1h)/m_anclinh);
        cons4 = ceil((j + cons1v)/m_anclinv);
        if(j < m_colfin2)
        {
            for (k = 0; k < cons2; k++)
            {
                if(cons3 == 2*k+1 || cons4 == 2*k+1)
                {aa = 0;}
            }
        }
        dc.SetPixel(j, i, RGB(aa,aa,aa));
    }
}

```

En la Fig. 6.17. a, b, c, d, e, f, y g ) Se muestran 7 rejillas de Ronchi inclinadas con un periodo de 120 píxeles por franja y un desplazamiento de fase entre ellas de 60 grados.



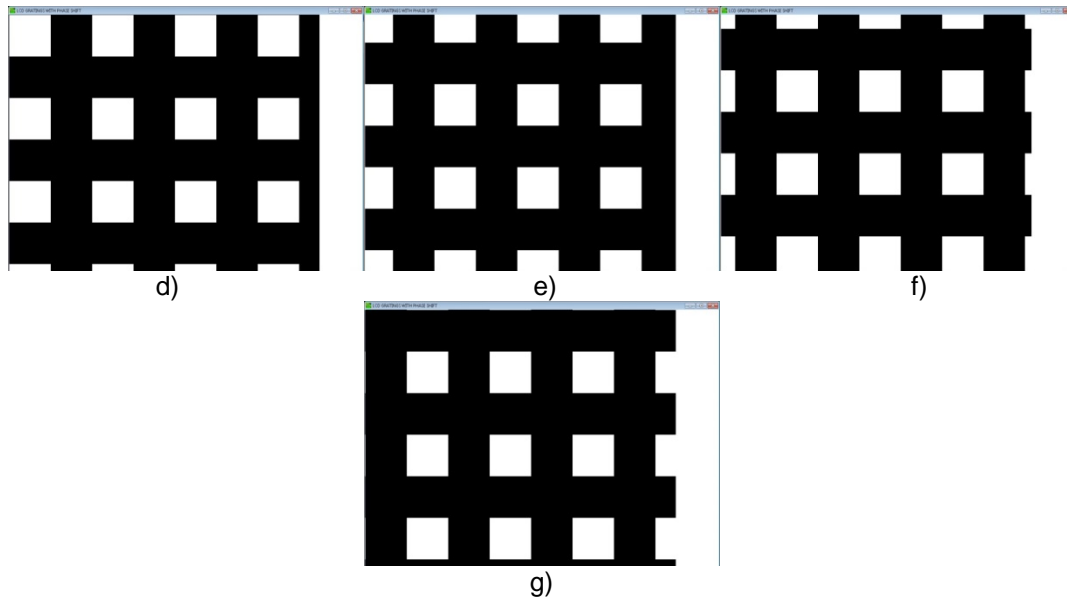


Fig. 6.17. Rejillas bironchi con un desplazamiento de fase de 120 píxeles por franja

La parte del programa que se muestra en la Fig. 6.18. es donde ingresamos los datos para crear rejillas circulares; se coloca la distancia entre los círculos en píxeles que es el equivalente al periodo de la rejilla y el centro de los círculos en renglones y columnas.

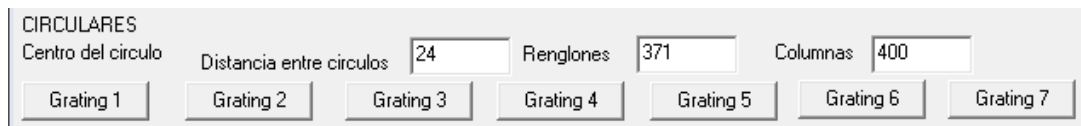


Fig. 6.18. Ingreso de datos de rejilla circular

El código del primer botón para la generación de rejillas circulares con el método seleccionado se muestra a continuación, cabe mencionar que cada botón de rejilla circular lleva un código similar, lo que cambia es la variable del desplazamiento de fase.

```
void CLCDGRATINGSDIg::OnButton34()
{
    int i,j,k,aa;
    double cons1, cons2, cons3, cons4;
    CClientDC dc(this);
    cons2 = 1024/(2*m_discir);
    if(m_iPsi == 0)
    {cons1 = 0*m_discir/180;}
    if(m_iPsi == 1)
    {cons1 = 0*m_discir/180;}
    if(m_iPsi == 2)
    {cons1 = 0*m_discir/180;}
    if(m_iPsi == 3)
```



```

{cons1 = 0*m_discir/180;}
if(m_iPsi == 4)
{cons1 = 0*m_discir/180;}
if(m_iPsi == 5)
{cons1 = 0*m_discir/180;}
if(m_iPsi == 6)
{cons1 = 0*m_discir/180;}
if(m_iPsi == 7)
{cons1 = 0*m_discir/180;}
for (j = 0; j < 1024; j++)
{
    for (i = 0; i < 743; i++)
    {
        aa = 255;
        cons3 = sqrt((i-m_cenren3)*(i-m_cenren3)+(j-m_cencol3)*(j-m_cencol3));
        cons4 = ceil((cons1+cons3)/m_discir);
        if(j < m_colfin2)
        {
            for (k = 0; k < cons2; k++)
            {
                if(cons4 == 2*k+1 )
                {aa = 0;}
            }
        }
        dc.SetPixel(j, i, RGB(aa,aa,aa));
    }
}
    
```

En la Fig. 6.19. a, b, c, d, e y f ) Se muestran 6 rejillas de circulares con una distancia entre círculos de 48 píxeles y un desplazamiento de fase entre ellas de 0,90,180, 240, 360 y 450 grados.

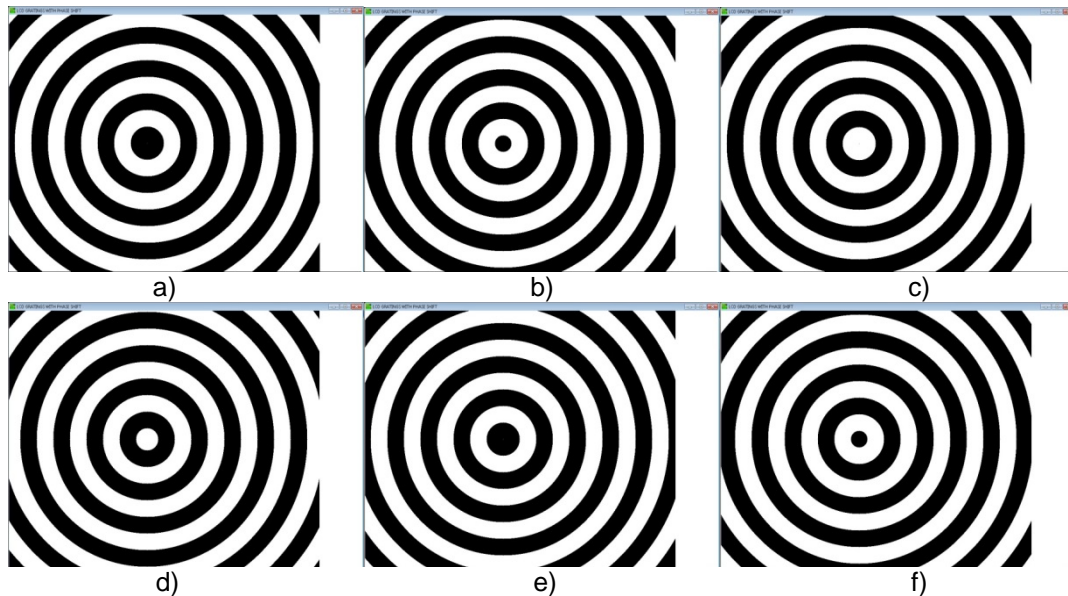


Fig. 6.19. Rejilla circulares con desplazamientos de fase de 0, 90, 180, 240, 360 y 450 grados

## 6.4. Software para analizar interferogramas y Ronchigramas.

En la Fig. 6.20 se muestra la ventana principal del software desarrollado para el análisis de patrones de franjas.

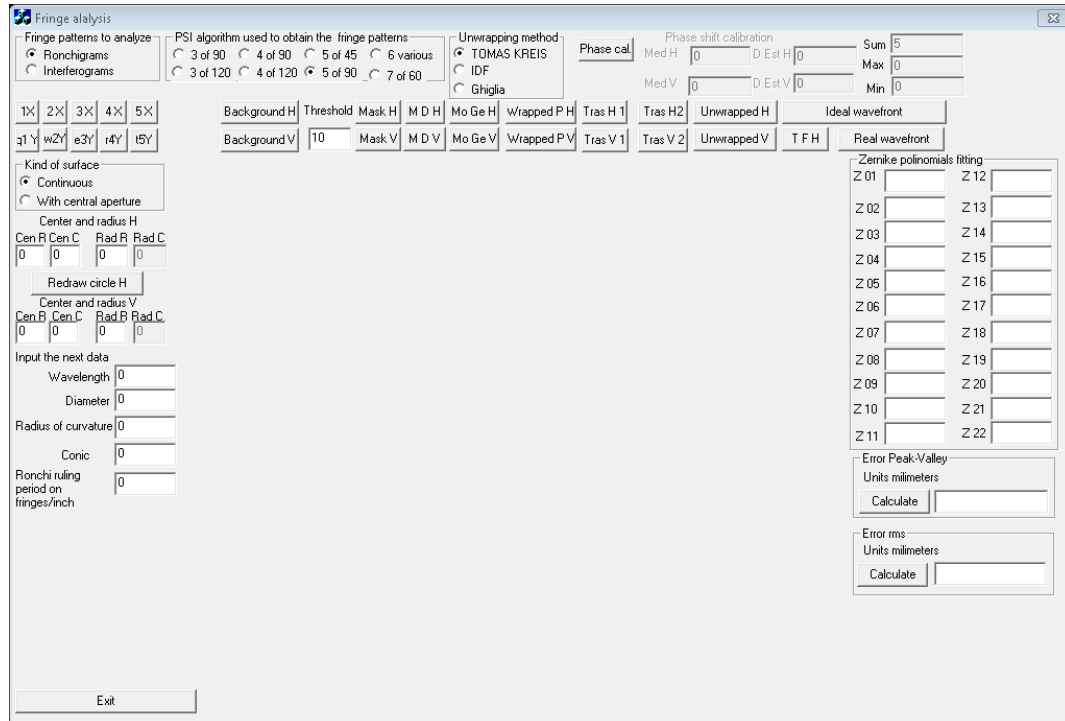


Fig. 6.20. Ventana principal del software desarrollado para el análisis de patrones de franjas.

### 6.4.1. Selección de tipo de patrones de franjas a analizar, algoritmo PSI y método de desenvolvimiento.

En la Fig. 6.21 aparece la parte del programa donde seleccionamos lo que vamos a analizar, ya que pueden ser patrones de franjas obtenidos con un interferómetro o patrones de franjas obtenidos con la prueba de Ronchi. También se selecciona el algoritmo PSI con el que se tomaron los patrones de franjas. Y finalmente en esta parte seleccionamos el método de desenvolvimiento con el que queremos se hará el análisis.

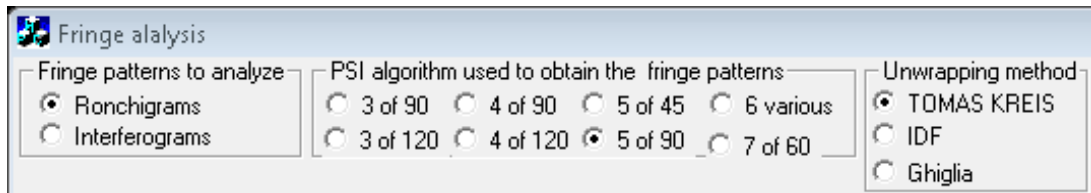


Fig. 6.21. Selección de tipo de franjas a analizar, método PSI y método de desenvolvimiento.

#### 6.4.2. Abrir patrones de franjas a analizar.

La lectura de patrones de franjas verticales se hace con los botones 1X, 2X, 3X, 4X, 5X, 6X Y 7X de la Fig. 6.22 los cuales abren patrones de franjas con desplazamientos de fase horizontal. Dichos archivos deberán estar guardados con formato \*.dat.

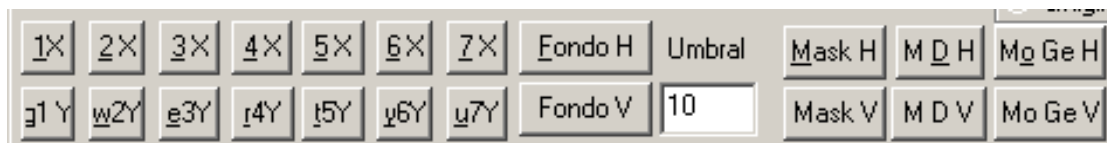


Fig. 6.22. Lectura de imágenes Fondo de las imágenes y umbral.

El código para abrir las imágenes verticales en el programa se muestra a continuación, solo se muestra el código de la función para un botón.

```
void CAnsetDlg::OnB1h()
// funcion abrir imagen 1h
CFileDialog m_IdFile(TRUE);
if (m_IdFile.DoModal() == IDOK)
{
    m_nombre = m_IdFile.GetPathName();
    UpdateData(FALSE);
}
CClientDC dc(this);
stream = fopen ( m_nombre, "r");
for (i = 0; i < 480; i++)
{
    for (j = 0; j < 640; j++)
    {
        fscanf(stream, "%ld \n", &aa);
        dc.SetPixel(j+160, i+120, RGB(aa,aa,aa));
        l1H[i][j] = aa;
    }
}
} //Cierra función
```

La lectura de patrones de franjas horizontales se hace con los botones q1Y, w2Y, e3Y, r4Y, t5Y, y6Y Y u7Y de la Fig. 6.21 los cuales abren patrones de franjas con desplazamientos de fase vertical. Dichos archivos deberán estar

guardados con formato \*.dat. El código para abrir las imágenes es similar al de los patrones de franjas verticales con desplazamiento de fase horizontal.

Cuando se analizan interferogramas no es necesario abrir imágenes con desplazamiento de fase horizontal, este solo es necesario cuando se analizan patrones de franjas obtenidos con desplazamiento de fase lateral en la prueba de Ronchi.

### 6.4.3. Cálculo de las ecuaciones de fondo.

Los botones FONDO H y FONDO V de la Fig. 6.22, sirven para calcular el fondo combinado de la serie de interferogramas o Ronchigramas según sea el caso, Para este cálculo se usan las ecuaciones de la sección 3.4. y el código para los diferentes algoritmos PSI implementados es el siguiente.

```
void CAnsetDlg::OnBfondoh()
{//Abrir funcion fondoh
CClientDC dc(this);
if (m_iPsi == 0)
{
for (i = 0; i < 480; i++)
{
for (j = 0; j < 640; j++)
{
fondoh[i][j] = (I1H[i][j] + I3H[i][j]);
aa = fondoh[i][j]*255/255;          dc.SetPixel(j+160, i+120, RGB(0,0,aa));
}
}
}
if (m_iPsi == 1)
{
for (i = 0; i < 480; i++)
{
for (j = 0; j < 640; j++)
{
fondoh[i][j] = (I1H[i][j] + I2H[i][j] + I3H[i][j]);
aa = fondoh[i][j]*255/382;          dc.SetPixel(j+160, i+120, RGB(0,0,aa));
}
}
}
if (m_iPsi == 2)
{
for (i = 0; i < 480; i++)
{
for (j = 0; j < 640; j++)
{
fondoh[i][j] = (I1H[i][j] + I2H[i][j] + I3H[i][j] + I4H[i][j]);
aa = fondoh[i][j]*255/511;          dc.SetPixel(j+160, i+120, RGB(0,0,aa));
}
}
}
if (m_iPsi == 3)
```

```

{
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            fondoh[i][j] = (11H[i][j] + 2*12H[i][j] + 2*13H[i][j] + 14H[i][j]);
            aa = fondoh[i][j]*255/767;           dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
        }
    }
}
if (m_iPsi == 4)
{
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            fondoh[i][j] = (11H[i][j] + 15H[i][j]);
            aa = fondoh[i][j]*255/255;           dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
        }
    }
}
if (m_iPsi == 5)
{
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            fondoh[i][j]= (2*11H[i][j]+3*12H[i][j]+4*13H[i][j]+3*14H[i][j]+2*15H[i][j]);
            aa = fondoh[i][j]*255/1785;           dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
        }
    }
}
if (m_iPsi == 6)
{
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            fondoh[i][j] = (11H[i][j] + 2*13H[i][j] + 15H[i][j]);
            aa = fondoh[i][j]*255/511;           dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
        }
    }
}
if (m_iPsi == 7)
{
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            fondoh[i][j] = (11H[i][j]+12H[i][j]+13H[i][j]+2*14H[i][j]+15H[i][j]+16H[i][j]+17H[i][j]);
            aa = fondoh[i][j]*255/1023;           dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
        }
    }
}
} //Cierra funcion fondoh

```

Una imagen de cómo queda el fondo combinado con el programa se muestra a continuación

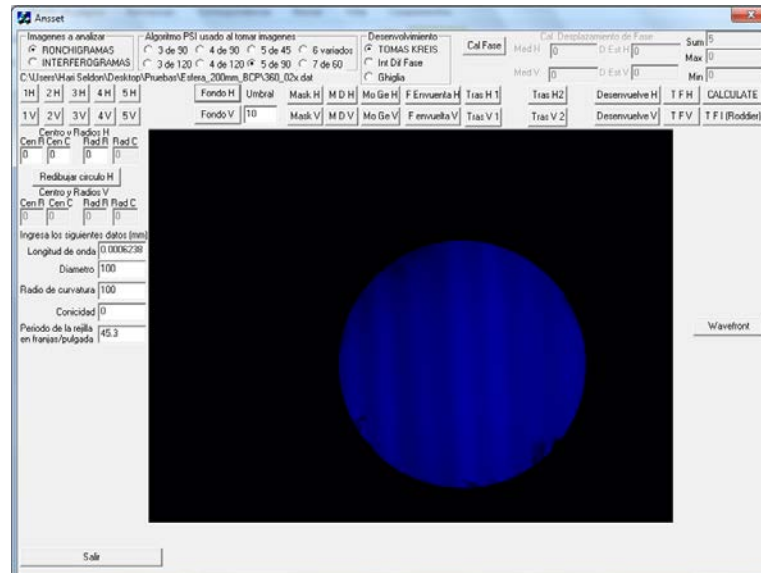


Fig. 6.23. Fondo combinado de una serie de Ronchigramas.

#### 6.4.4. Máscara de los píxeles de fase valida.

Después del cálculo del fondo se calcula la máscara de los píxeles de fase valida. El umbral por default se encuentra con un valor de 10, pero se puede cambiar de acuerdo a las necesidades, Este valor se modifica o se deja así antes de calcular la máscara de los interferogramas, la máscara nos indica que píxeles contienen fase valida, y nos sirven para comenzar a hacer el análisis.

El código para calcular la máscara se usan los botones MASK H y MASK V estos usan el fondo combinado de los interferogramas y en base a cálculos teóricos, se aplican diferentes condiciones para cada algoritmo PSI, el código para cada algoritmo se muestra a continuación.

```
void CAnsetDlg::OnBmascarah()
{
    //Abre funcion mascara h
    CClientDC dc(this);
    for (i = 0; i < 480; i++)
    {
        for (j = 0; j < 640; j++)
        {
            mascarah[i][j] = 0;
            if(m_iPsi == 0)
            {
                if (fondoh[i][j] > m_umbral*2.55)
                {
                    mascarah[i][j] = 1;
                }
            }
        }
    }
}
```

```
else if (m_iPsi == 1)
{
    if (fondoh[i][j] > m_umbral*3.82)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 2)
{
    if (fondoh[i][j] > m_umbral*5.11)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 3)
{
    if (fondoh[i][j] > m_umbral*7.67)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 4)
{
    if (fondoh[i][j] > m_umbral*2.55)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 5)
{
    if (fondoh[i][j] > m_umbral*17.85)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 6)
{
    if (fondoh[i][j] > m_umbral*5.11)
        { mascarah[i][j] = 1;}
}
else if (m_iPsi == 7)
{
    if (fondoh[i][j] > m_umbral*10.23)
        { mascarah[i][j] = 1;}
}
aa = mascarah[i][j] * 255;          dc.SetPixel(j+160, i+120, RGB(0,aa,aa));
}
}
} //Cierra funcion mascara h
```

En la Fig. 6.24 se muestra una imagen de cómo queda la máscara de píxeles de fase valida con un umbral de 10 para una serie de Ronchigramas que tienen muy buen contraste.

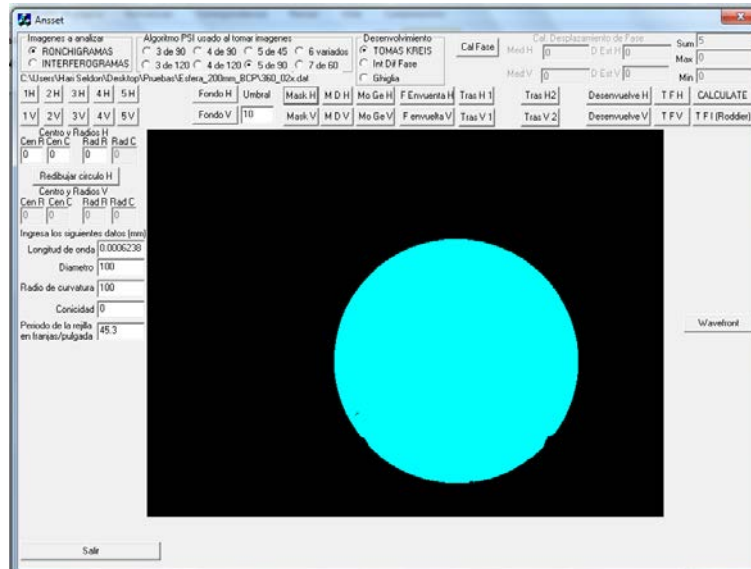


Fig. 6.24. Máscara de los píxeles de fase válida para una serie de Ronchigramas.

#### 6.4.5. Ecuación de modulación de datos y momentos geométricos.

Los botones M D H y M D V se calculan con la ecuación de modulación de datos de Schreiber [13] de la sección 3.2 y ofrecen resultados similares a los de la máscara calculada con las ecuaciones de fondo. En la fig. 6.25 se muestra la máscara obtenida con las ecuaciones de modulación de datos.

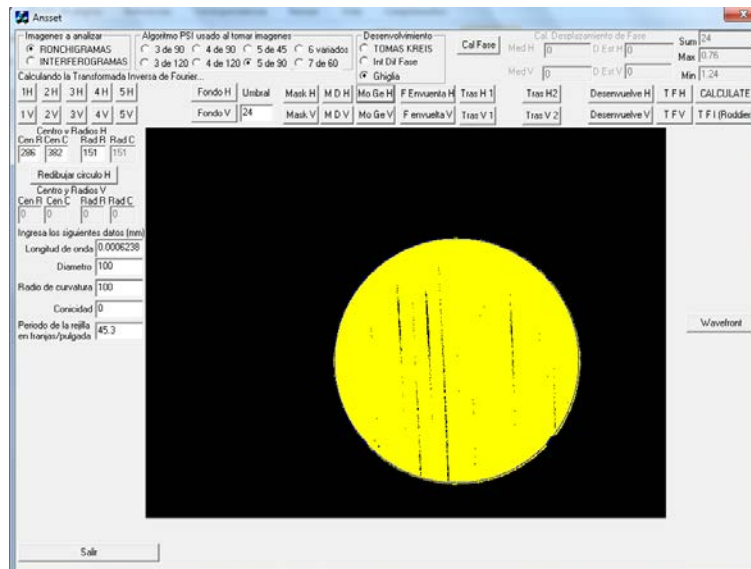


Fig. 6.25. Máscara de píxeles de fase válida obtenidos con la ecuación de modulación



Los momentos geométricos se calculan con los botones Mo Ge H Y Mo Ge V sirven para calcular los momentos geométricos de la máscara, con esto conocemos el centroide y radio de la máscara en píxeles. A continuación se muestra el código que se usa para calcular los momentos geométricos.

```

void CAnsetDlg::OnBmogeH()
{//Abre funcion momentos geometricos h
  for (p = 0; p < 3; p++)
  {
    for ( q = 0; q < 3; q++)
    {
      M[p][q] = 0;
      for (i = 0; i < 480; i++)
      {
        for (j = 0; j < 640; j++)
        {
          if ( mascararah[i][j] ==1)
          {
            poti = pow(i,p);          potj = pow(j,q);
            M[p][q] += poti * potj;
          }
          else
            continue;
        }
      }
    }
  }
  X_I[0][0] = M[1][0] / M[0][0];          X_I[0][1] = M[0][1] / M[0][0];
  for (p = 0; p < 3; p++)
  {
    for ( q = 0; q < 3; q++)
    {
      U[p][q] = 0;
      for (i = 0; i < 480; i++)
      {
        for (j = 0; j < 640; j++)
        {
          if ( mascararah[i][j] ==1)
          {
            difpoti = i - X_I[0][0];          difpotj = j - X_I[0][1];
            poti = pow(difpoti,p);          potj = pow(difpotj,q);
            U[p][q] += poti * potj;
          }
          else
            continue;
        }
      }
    }
  }
  a = U[2][0] - U[0][2]; b = pow(a,2);          c = b + 4 * U[1][1] * U[1][1];          d = sqrt(c);
  X_I[1][0] = ((U[2][0] + U[0][2]) + d)/2;          X_I[1][1] = ((U[2][0] + U[0][2]) - d)/2;
  e = 2 * (U[1][1] / a);          X_I[1][2] = (1/2) * atan(e);
  f = X_I[1][0] / U[0][0];          g = X_I[1][1] / U[0][0];
  a1 = pow(f,0.5);          b1 = pow(g,0.5);          X_I[2][0] = 2 * a1;          X_I[2][1] = 2 * b1;
  cereh = X_I[0][0];          cecoh = X_I[0][1];          rareh = X_I[2][1];          racoh = X_I[2][0];
  m_cereh = cereh;          m_cecoh = cecoh;          m_rareh = rareh;          m_racoh = racoh;
  UpdateData(FALSE);
  CClientDC dc(this);
  for (i = 0; i < 480; i++)
  {
    for (j = 0; j < 640; j++)
    {

```

```
        radio = sqrt((j-cecoh)*(j-cecoh)+(cereh-i)*(cereh-i));
        if (radio < rareh+0 )
            {
                mascaraff[i][j] = 1;
                if (radio > rareh-1)
                    {
                        aa = 255;          dc.SetPíxel(j+160, i+120, RGB(0,0,aa));
                    }
            }
    }
} //Cierra funcion momentos geometricos h
```

En la Fig. 6.26 se muestra se muestra una imagen de los momentos geométricos. De color azul fuerte se ve el círculo donde dentro de él se encuentra la fase valida del lado derecho de la imagen el programa nos muestra el centroide y el radio de la máscara.

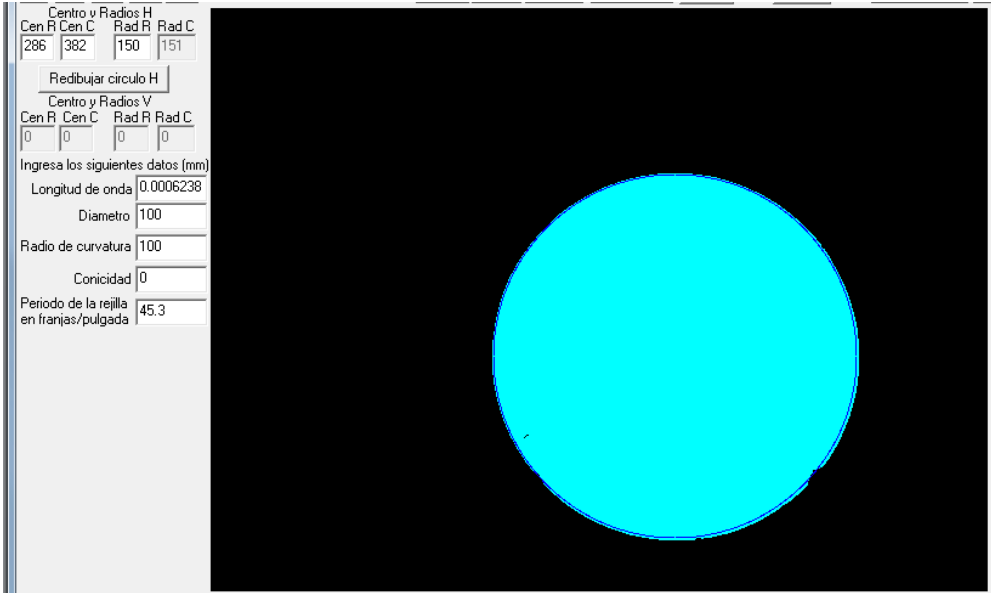


Fig. 6.26. Centroide y radio de la máscara de los píxeles de fase valida.

**6.4.6. Fase envuelta, traslación de la fase y desenvolvimiento de la fase.**

El botón F Envuelta H y F envuelta V, calculan la fase envuelta de lo que se encuentre dentro del círculo que nos dieron los momentos geométricos. A continuación se muestra el código usado para calcular la fase envuelta para cada algoritmo PSI.

```

void CAnsetDlg::OnBfenh()
{
    //Abre funcion fase envuelta h
    //stream1 = fopen ( "Calibracion de la fase.dat", "w");           iteraciones = jj/2;           piin = 0;
    CClientDC dc(this);
    for (i = 0; i < 480; i++)
        {
            //abre for i
            for (j = 0; j < 640; j++)
                {
                    //abre for j
                    Phienvuelta[i][j] = 0; mo_Phih[i][j] = 0;
                    if(mascaraf[i][j] == 1)
                        {
                            //abre if principal
                            if(m_iPsi == 0)
                                {
                                    //abre if
                                    Phi1[i][j] = - ( -11H[i][j] + 2*12H[i][j] - 13H[i][j] );
                                    Phi2[i][j] = 11H[i][j] - 13H[i][j];
                                    //Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]); //1 A
                                    Phienvuelta[i][j] = atan2( Phi2[i][j] , -Phi1[i][j]); //1 B
                                }
                                //cierra if
                            else if (m_iPsi == 1)
                                {
                                    //abre if
                                    Phi1[i][j] = -(sqrt(3)) * ( 11H[i][j] - 13H[i][j] );
                                    Phi2[i][j] = 11H[i][j] - 2*12H[i][j] + 13H[i][j];
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 2)
                                {
                                    //abre if
                                    //Phi1[i][j] = 14H[i][j] - 12H[i][j]; //A
                                    //Phi2[i][j] = 11H[i][j] - 13H[i][j]; //A
                                    Phi1[i][j] = 11H[i][j] + 12H[i][j] - 13H[i][j] - 14H[i][j]; //B
                                    Phi2[i][j] = 11H[i][j] - 12H[i][j] - 13H[i][j] + 14H[i][j]; //B
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 3)
                                {
                                    //abre if
                                    Phi1[i][j] = -(sqrt(3)) * ( 12H[i][j] - 13H[i][j] );
                                    Phi2[i][j] = 11H[i][j] - 12H[i][j] - 13H[i][j] + 14H[i][j];
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 4)
                                {
                                    //abre if
                                    Phi1[i][j] = - ( -11H[i][j] + 4*12H[i][j] - 4*14H[i][j] + 15H[i][j] );
                                    Phi2[i][j] = 11H[i][j] + 2*12H[i][j] - 6*13H[i][j] + 2*14H[i][j] + 15H[i][j];
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 5)
                                {
                                    //abre if
                                    Phi1[i][j] = 2 * ( 12H[i][j] - 14H[i][j] ); //A
                                    Phi2[i][j] = 2 * 13H[i][j] - 15H[i][j] - 11H[i][j]; //A
                                    //Phi1[i][j] = - ( 3*12H[i][j] - 3*13H[i][j] - 14H[i][j] + 15H[i][j] ); //B
                                    Phi2[i][j] = 11H[i][j] - 12H[i][j] - 3*13H[i][j] + 3*14H[i][j]; //B
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 6)
                                {
                                    //abre if
                                    Phi1[i][j] = 11H[i][j] + 3*12H[i][j] - 4*14H[i][j] - 15H[i][j] + 16H[i][j];
                                    Phi2[i][j] = 11H[i][j] - 12H[i][j] - 4*13H[i][j] + 3*15H[i][j] + 16H[i][j];
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                            else if (m_iPsi == 7)
                                {
                                    //abre if
                                    Phi1[i][j] = -(sqrt(3))*(12H[i][j] + 13H[i][j] - 15H[i][j] - 16H[i][j]);
                                    Phi2[i][j] = 11H[i][j] + 12H[i][j] - 13H[i][j] - 2*14H[i][j] - 15H[i][j] + 16H[i][j] + 17H[i][j];
                                    Phienvuelta[i][j] = atan2( Phi1[i][j] , Phi2[i][j]);
                                }
                                //cierra if
                        }
                    //cierra if principal

                    mo_Phih[i][j] = (Phienvuelta[i][j] + pi);
                }
        }
}

```

```

aa = mo_Phih[i][j]*255/(2*pi);
dc.SetPixel(j+160, i+120, RGB(aa,aa,aa));

if (m_iPsi == 5)
  {
  //abre if
  Phi1[i][j] = I5H[i][j] - I1H[i][j];
  Phi2[i][j] = 2*(I4H[i][j] - I2H[i][j]);
  Phi3[i][j] = Phi1[i][j] / Phi2[i][j];
  if (Phi2[i][j] != 0)
    {
    CPhiH[i][j] = (acos( Phi1[i][j])) * 180/pi;
    if (Phi1[i][j] <= -1)
      {
      CPhiH[i][j] = 180;
      }
    else if (Phi1[i][j] >= 1)
      {
      CPhiH[i][j] = 0;
      }
    else if (Phi1[i][j] == 0)
      {
      CPhiH[i][j] = 90;
      }
    }
  if (Phi2[i][j] == 0)
    {
    CPhiH[i][j] = 0;
    }
  }
//
fprintf( stream1, " %d %d %f %f \n", i, j, Phienvuelta[i][j], CPhiH[i][j]);
} //cierra if
} //cierra for j
} //cierra fot i
} //Cierra funcion fase envuelta h

```

En la Fig. 6.27 se muestra una imagen de la fase envuelta para una serie de Ronchigramas con desplazamientos de fase de 90 grados.

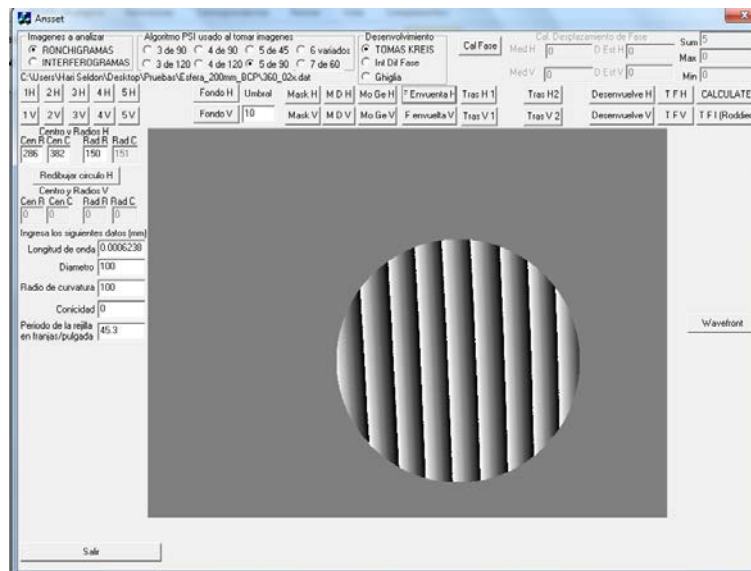
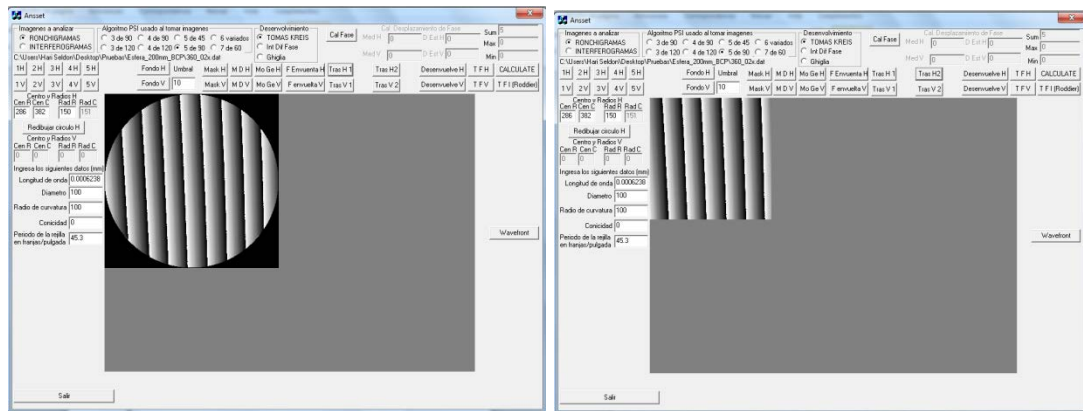


Fig. 6.27. Fase Envuelta.

Para facilitar el análisis una vez que se conoce la fase envuelta se hace una traslación. Los botones Tras H 1, Tras V 1, Tras H 2 y Tras V 2, nos trasladan la fase a la esquina superior izquierda para facilitar la programación del cálculo de la fase desenvuelta, los primeros trasladan la fase que en forma de cuadrado fuera del círculo, y los últimos trasladan la fase que está en un cuadrado dentro del círculo. La Fig 6.28 a, b) muestra las imágenes de ambas traslaciones.



a) b)  
Fig. 6.28. Traslaciones de la fase.

Para desenvolver la fase el botón Desenvuelve H y Desenvuelve V, desenvuelven la fase con el método indicado, y devuelven la fase desenvuelta. A continuación se muestra el código de los métodos de desenvolvimiento programados.

```

////////////////////////////////////
////////  FUNCION DONDE SE CALCULA LA FASE ENVUELTA CON 3 METODOS H  //////////
////////////////////////////////////
void CAnsetDlg::OnBdesh()
{//Inicia funcion desenvolvimiento h
CClientDC dc(this);
////////////////////////////////////
////////////////////////////////////  DESENVOLVIMIENTO CON EL METODO DE KREIS H  ///////////////////////////////////
if(m_iDesenvolvimiento == 0)
{
stream1 = fopen ( "FDH_KREIS.dat", "w");          iteraciones = jj/2;          piin = 0;
for (i = 0; i < ii; i++)
{
for (j = 0; j < jj; j++)
{
fa_en_0[i][j] = mod_Phih[i][j];
mascara2[i][j] = 0;          M4[i][j] = 0;          C3[i][j] = 0;
} //cierra for j
} //cierra for i
}
}

```

```

for ( o= 0; o < iteraciones; o++)
{
for (i = 0; i < ii; i++)
{
for (j = 0; j < jj; j++)
{
if (piin == 0 && mascara1[i][j] == 1 && mascara2[i][j] != 2)
{
mascara2[i][j] = 2;          piin = 1;          M4[i][j] = 0;
}
if (piin != 0 && mascara1[i][j] == 1 && mascara2[i][j] != 2)
{//abre if 1
if (mascara2[i][j-1] == 2 )
{//abre if 2
xx = fa_en_0[i][j] - fa_en_0[i][j-1];
if(xx < pi)
{ M4[i][j] = M4[i][j-1]; }
if(xx <= -pi)
{ M4[i][j] = M4[i][j-1] + 1; }
if(xx >= pi)
{ M4[i][j] = M4[i][j-1] - 1; }
mascara2[i][j] = 2;
} //cierra if 2
if (mascara2[i-1][j] == 2 )
{//abre if 3
xx = fa_en_0[i][j] - fa_en_0[i-1][j];
if(xx < pi)
{ M4[i][j] = M4[i-1][j]; }
if(xx <= -pi)
{ M4[i][j] = M4[i-1][j] + 1; }
if(xx >= pi)
{ M4[i][j] = M4[i-1][j] - 1; }
mascara2[i][j] = 2;
} //cierra if 3
if (mascara2[i-1][j+1] == 2 )
{//abre if 4
xx = fa_en_0[i][j] - fa_en_0[i-1][j+1];
if(xx < pi)
{ M4[i][j] = M4[i-1][j+1]; }
if(xx <= -pi)
{ M4[i][j] = M4[i-1][j+1] + 1; }
if(xx >= pi)
{ M4[i][j] = M4[i-1][j+1] - 1; }
mascara2[i][j] = 2;
} //cierra if 4
if (mascara2[i][j+1] == 2 )
{//abre if 5
xx = fa_en_0[i][j] - fa_en_0[i][j+1];
if(xx < pi)
{ M4[i][j] = M4[i][j+1]; }
if(xx <= -pi)
{ M4[i][j] = M4[i][j+1] + 1; }
if(xx >= pi)
{ M4[i][j] = M4[i][j+1] - 1; }
mascara2[i][j] = 2;
} //cierra if 5
} //cierra if 1
} //cierra for j
} //cierra for i
// cout << "k" << k << "\n";
} //cierra for k
suma=0; maximo=0;          minimo=0;
for (i = 0; i < ii; i++)
{
for (j = 0; j < jj; j++)
{
C3[i][j] = fa_en_0[i][j] + 2*pi*M4[i][j];
}
}
}
}
}

```

```

        suma = suma + 1;
        if (C3[i][j] > maximo )
            {
                maximo = C3[i][j];
            }
        if (C3[i][j] < minimo )
            {
                minimo = C3[i][j];
            }
        } //cierra for j
    } //cierra for i
m_suma = suma;          m_maximo = maximo;          m_minimo = minimo;
UpdateData(FALSE);
for (i = 0; i < ii; i++)
    {
        for (j = 0; j < jj; j++)
            {
                fprintf( stream1, "%d %d %f \n", i, j, C3[i][j]);
            }
    }
}
}
////////////////////////////////////////////////////
/////   DESENVOLVIMIENTO CON INTEGRACION DE DIFERENCIAS DE FASE   /////
else if (m_iDesenvolvimiento == 1)
    {
        stream1 = fopen ( "FDH_IDF.dat", "w");          iteraciones = jj/2;          piin = 0;
        for (i = 0; i < ii; i++)
            {
                for (j = 0; j < jj; j++)
                    {
                        fa_en_1[i][j] = mod_Phik[i][j];
                        fa_de_1[i][j] = 0;          mascara2[i][j] = 0;          C3[i][j] = 0;
                    }
                } //cierra for j
            } //cierra for i

        for ( o= 0; o < iteraciones; o++)
        {
            for (i = 0; i < ii; i++)
                {
                    for (j = 0; j < jj; j++)
                        {
                            if (piin == 0 && mascara1[i][j] == 1 && mascara2[i][j] != 2)
                                {
                                    mascara2[i][j] = 2;
                                    piin = 1;
                                    fa_de_1[i][j] = 0;
                                }
                            if (piin != 0 && mascara1[i][j] == 1 && mascara2[i][j] != 2)
                                { //abre if 1
                                    if (mascara2[i][j-1] == 2 )
                                        { //abre if 2
                                            X[i][j] = fa_en_1[i][j] - fa_de_1[i][j-1];
                                            V[i][j] = atan (sin(X[i][j]) / cos(X[i][j]));
                                            fa_de_1[i][j] = fa_de_1[i][j-1] + V[i][j];
                                            mascara2[i][j] = 2;
                                        } //cierra if 2

                                    if (mascara2[i-1][j] == 2 )
                                        { //abre if 3
                                            X[i][j] = fa_en_1[i][j] - fa_de_1[i-1][j];
                                            V[i][j] = atan (sin(X[i][j]) / cos(X[i][j]));
                                            fa_de_1[i][j] = fa_de_1[i-1][j] + V[i][j];
                                            mascara2[i][j] = 2;
                                        } //cierra if 3

                                    if (mascara2[i-1][j+1] == 2 )
                                        { //abre if 4
                                            X[i][j] = fa_en_1[i][j] - fa_de_1[i-1][j+1];
                                            V[i][j] = atan (sin(X[i][j]) / cos(X[i][j]));
                                            fa_de_1[i][j] = fa_de_1[i-1][j+1] + V[i][j];
                                            mascara2[i][j] = 2;
                                        }
                                }
                        }
                }
        }
    }
}
}

```

```

                                                                    //cierra if 4
                    if (mascara2[i][j+1] == 2 )
                        { //abre if 5
                            X[i][j] = fa_en_1[i][j] - fa_de_1[i][j+1];
                            V[i][j] = atan (sin(X[i][j]) / cos(X[i][j]));
                            fa_de_1[i][j] = fa_de_1[i][j+1] + V[i][j];
                            mascara2[i][j] = 2;
                        } //cierra if 5
                    } //cierra if 1
                } //cierra for j
            } //cierra for i
//      cout << "k" << k << "\n";
} //cierra for k
suma=0; maximo=0;          minimo=0;
for (i = 0; i < ii; i++)
    {
        for (j = 0; j < jj; j++)
            {
                C3[i][j] = fa_de_1[i][j];
                suma = suma + 1;
                if (C3[i][j] > maximo )
                    {          maximo = C3[i][j];          }
                if (C3[i][j] < minimo )
                    {          minimo = C3[i][j];          }
            } //cierra for j
        } //cierra for i
m_suma = suma;
m_maximo = maximo;
m_minimo = minimo;
UpdateData(FALSE);
for (i = 0; i <= ii; i++)
    {
        for (j = 0; j <= jj; j++)
            {
                fprintf( stream1, "%d %d %f \n", i, j, C3[i][j]);
            }
    }

/////////////////////////////////////////////////////////////////
//////////////////////      DESENVOLVIMIENTO CON EL METODO DE GHIGLIA      ////////////////////////
else if (m_iDesenvolvimiento == 2)
    {
        stream1 = fopen ( "FDH_GHIGLIA.dat", "w");
        for (i = 0; i < ii; i++)
            {
                for (j = 0; j < jj; j++)
                    {
                        fa_en_2[i][j] = mod_Phih[i][j];
                        fa_de_2[i][j] = 0;          C3[i][j] = 0;          mascara2[i][j] = 0;
                        delta_X[i][j] = 0;          delta_Y[i][j] = 0;          delta_X_Y[i][j] = 0;
                        C[i][j] = 0;          C1[i][j] = 0;          C2[i][j] = 0;
                    }
            }
        for (i = 0; i < ii; i++)
            {
                for (j = 0; j < jj; j++)
                    {
                        delta_X_Y[i][j] = 0;
                        if (mascara1[i][j] == 1)
                            {
                                if ( j == 0 && mascara1[i][j+1] == 1)
                                    {
                                        delta_X_Y[i][j] = fa_en_2[i][j+1] - fa_en_2[i][j];
                                    }
                                if ( j == jj-1 && mascara1[i][j-1] == 1)

```



```

        {
            delta_X_Y[i][j] = fa_en_2[i][j-1] - fa_en_2[i][j] ;
        }
    if ( 0<=j && j<jj-1 )
        {
            if (mascara1[i][j+1] == 0 && mascara1[i][j-1] == 1)
                {
                    delta_X_Y[i][j] = fa_en_2[i][j-1] - fa_en_2[i][j] ;
                }
            if (mascara1[i][j+1] == 1 && mascara1[i][j-1] == 0)
                {
                    delta_X_Y[i][j] = fa_en_2[i][j+1] - fa_en_2[i][j] ;
                }
            if (mascara1[i][j+1] == 1 && mascara1[i][j-1] == 1)
                {
                    delta_X_Y[i][j] = fa_en_2[i][j+1] + fa_en_2[i][j-1] - 2*fa_en_2[i][j] ;
                }
        }
    }
}

////////////////////////////////////
//TRANSFORMADA COSENO DE FOURIER
//*
m_nombre = "Calculando la Transformada de Fourier...";
UpdateData(FALSE);
for (m = 0; m < ii; m++)
    {
        for (n = 0; n < jj; n++)
            {
                C[m][n] = 0;
                if (mascara1[m][n] == 0)
                    { C[m][n] = 0; }
                if (mascara1[m][n] == 1)
                    {
                        for (j = 0; j < jj; j++)
                            {
                                co = cos(pi*(n)*(2*(j)+1)/(2*jj));
                                C[m][n] = C[m][n] + 2*delta_X_Y[m][j]*co;
                            }
                    }
            }
    }
m_nombre = "Calculando el filtro de la Transformada de Fourier...";
UpdateData(FALSE);
for (i = 0; i < ii; i++)
    {
        for (j = 0; j < jj; j++)
            {
                if (mascara1[i][j] == 1)
                    {
                        cos1 = cos(pi*(j)/jj);
                        if (cos1 == 1)
                            {
                                C1[i][j]=0;
                            }
                        if (cos1 != 1)
                            {
                                C1[i][j] = C[i][j] / (2 * (cos1-1));
                            }
                    }
            }
    }
//TRANSFORMADA COSENO INVERSA DE FOURIER
m_nombre = "Calculando la Transformada Inversa de Fourier...";

```

```

UpdateData(FALSE);
for (m = 0; m < ii; m++)
    {
    for (n = 0; n < jj; n++)
        {
        C2[m][n] = 0;
        if (mascara1[m][n] == 1)
            {
            for (j = 0; j < jj; j++)
                {
                if (j==0)
                    {co = 0.5*cos(pi*(n)*(2*(j)+1)/(2*jj));}
                if (j!=0)
                    {co = cos(pi*(n)*(2*(j)+1)/(2*jj));}
                C2[m][n] = C2[m][n] + C1[m][j]*co;
                }
            }
        C2[m][n] = C2[m][n] / (jj);
        if (mascara1[m][n] == 0)
            {
            C2[m][n] = -1000;
            }
        }
    }
suma=0; maximo=0;           minimo=0;
for (i = 0; i < ii; i++)
    {
    for (j = 0; j < jj; j++)
        {
        if (mascara1[i][j] == 1)
            {
            mascara2[i][j] = 2; C3[i][j] = C2[i][j];   suma = suma + 1;
            if (C3[i][j] > maximo )
                { maximo = C3[i][j]; }
            if (C3[i][j] < minimo )
                { minimo = C3[i][j]; }
            }
        }
    } //cierra for j
} //cierra for i
m_suma = suma; m_maximo = maximo; m_minimo = minimo; UpdateData(FALSE);
for (i = 0; i < ii; i++)
    {
    for (j = 0; j < jj; j++)
        {
        fprintf( stream1, " %d %d %d %f %f %f %f %f %f \n", i, j, mascara1[i][j], fa_en_2[i][j], delta_X_Y[i][j], C[i][j], C1[i][j],
        C2[i][j]);
        }
    }
//for (i = 0; i < ii; i++)
// {
// fprintf( stream1, " %d %d \n", i, COI[i][0]);
// }
} //cierra if desenvolvimiento 2
for (i = 0; i < ii; i++)
    {
    for (j = 0; j < jj; j++)
        {
        dc.SetPixel(j+160, i+120, RGB(0,0,0));
        if (mascara2[i][j] == 2)
            {
            aa = (C3[i][j]-minimo)*255/(maximo-minimo); dc.SetPixel(j+160, i+120, RGB(aa,aa,0));
            }
        }
    } //cierra for j
} //cierra for i
} //Termina funcion desenvolvimiento h

```

En la Fig. 6.29 se muestra la fase desenvuelta con el método de Kreis para una serie de Ronchigramas.

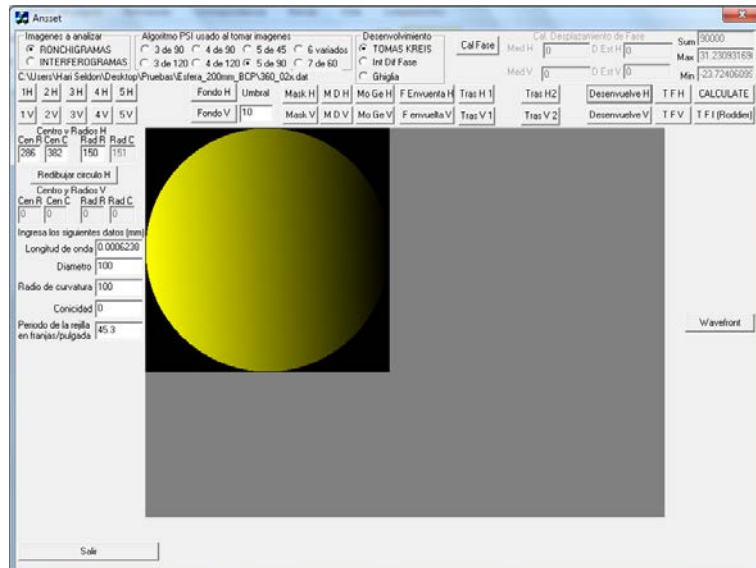


Fig. 6.29. Desenvolvimiento de la fase.

## 6.5. Conclusión.

Se explicó brevemente la forma de utilizar los programas desarrollados para desplegar rejillas en la LCD así como para hacer el análisis de patrones de franjas obtenidos con la técnica PSI, También se puso el código por funciones en C++.

# Capítulo 7

## 7. Resultados.

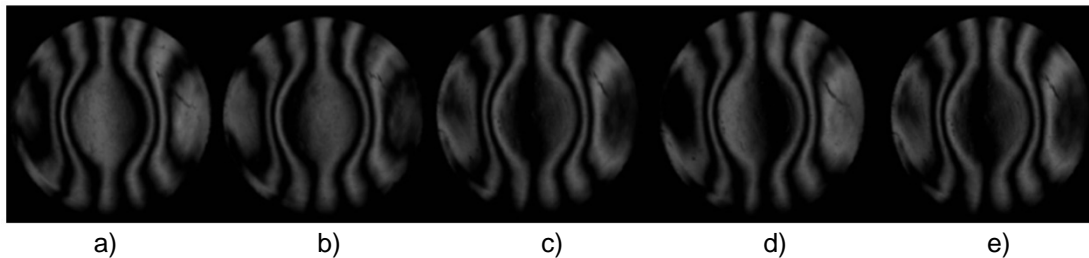
### 7.1. Introducción.

En este capítulo se muestran los patrones de franjas obtenidos, una comparación con Zemax de los resultados de la prueba con motores a pasos y una comparación con Durango ® de la prueba con la LCD. Además se explica la versatilidad de usar la LCD para implementar varias pruebas ópticas.

### 7.2. Prueba de Ronchi con los motores a pasos.

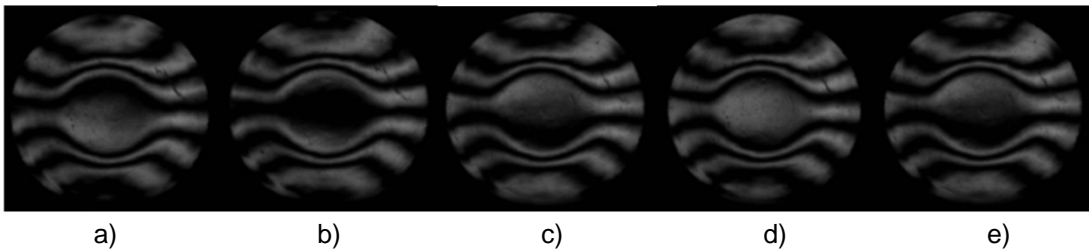
Se probó una lente biconvexa obtenida de una lupa escolar. Probamos esta lente que tiene un  $f/\# = 1.771$  y un diámetro de 27.56 mm empleando el arreglo experimental que usa los motores a pasos. Los motores tienen una resolución de 200 pasos por revolución, estos mueven un tornillo micrométrico que desplaza la rejilla, este tornillo en una revolución avanza 500 micras por lo que la distancia de desplazamiento por paso es de  $2.5\mu\text{m}$ . Cuando se compran las rejillas comerciales el vendedor nos da la resolución de estas en franjas/pulgada, para la rejilla que usamos la cual es de 50 franjas/pulgada haciendo una regla de tres, una franja físicamente mide  $508\mu\text{m}$ , para hacer un desplazamiento de fase de  $360^\circ$  es necesario desplazar la rejilla una franja, El algoritmo Hariharan usado en la prueba necesita 4 desplazamientos de fase de  $90^\circ$  que es equivalente a  $508\mu\text{m} / 4$  desplazamientos =  $127\mu\text{m} /$  desplazamiento, con el motor a pasos se

necesitan dar  $127\mu\text{m}/2.5\mu\text{m} = 50.8$ , pero como el motor a pasos solo da pasos completos se redondea a 51 pasos para obtener un desplazamiento aproximadamente igual a  $90^\circ$ . El CCD es colocado en la montura XYZ, el LED usado como fuente de luz es alimentado por la fuente de corriente constante, como se puede ver en la Figura 5.1. Los Ronchigramas verticales de la Fig. 7.1. se obtuvieron con el algoritmo PSI Hariharan de la sección 2.3.4.7.



**Fig. 7.1. Ronchigramas verticales.**

Los Ronchigramas horizontales de la Fig. 7.2. se obtuvieron con el algoritmo PSI Hariharan de la sección 2.3.4.7.



**Fig. 7.2. Ronchigramas horizontales.**

La aberración transversal en el eje X y la aberración del frente de onda obtenido se muestran en la Fig. 7.3. El error pico-valle de la aberración transversal en el eje X fue de 5.9938mm. El RMS es 0.0923mm, el error de desplazamiento de fase es de  $3.5795^\circ$ . El error pico-valle de la diferencia de camino óptico OPD es 3.5795mm y finalmente el valor de los coeficientes de los polinomios de Zernike se muestran en la Tabla 7.1.

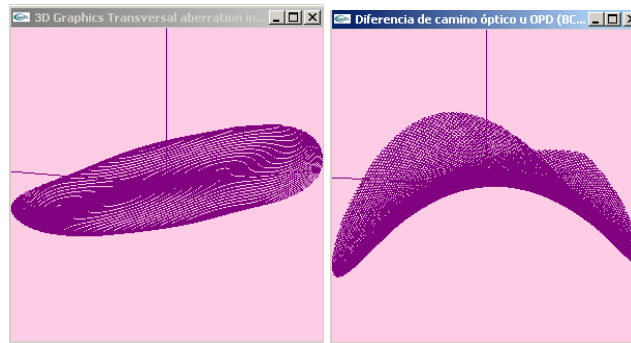


Fig. 7.3. Aberración transversal y aberración del frente de onda.

Tabla 7.1. Valores de los coeficientes de Zernike del espejo bajo prueba.

Transversal aberration coefficients in X		Transversal aberration coefficients in Y		Wave front coefficients in X,Y	
a <sub>3</sub> =	-0.01434	a <sub>3</sub> =	-0.01171	a <sub>3</sub> =	-0.01302
a <sub>5</sub> =	1.52862	a <sub>5</sub> =	1.56314	a <sub>5</sub> =	1.54058
a <sub>6</sub> =	0.00528	a <sub>6</sub> =	7.47255E-4	a <sub>6</sub> =	0.0048
a <sub>7</sub> =	-0.01919	a <sub>7</sub> =	0.00188	a <sub>7</sub> =	-0.00102
a <sub>8</sub> =	0.01184	a <sub>8</sub> =	0.00465	a <sub>8</sub> =	0.00326
a <sub>11</sub> =	-0.01761	a <sub>11</sub> =	0.00235	a <sub>11</sub> =	-4.00769E-4
a <sub>12</sub> =	0.0692	a <sub>12</sub> =	0.07783	a <sub>12</sub> =	0.07993
a <sub>13</sub> =	-0.0072	a <sub>13</sub> =	-0.01965	a <sub>13</sub> =	-0.00766
a <sub>14</sub> =	0.02615	a <sub>14</sub> =	3.20712E-4	a <sub>14</sub> =	7.52675E-4
a <sub>15</sub> =	0.00156	a <sub>15</sub> =	-0.00314	a <sub>15</sub> =	-0.00116
a <sub>16</sub> =	0.01444	a <sub>16</sub> =	-4.21816E-4	a <sub>16</sub> =	-0.00373
a <sub>17</sub> =	0.01741	a <sub>17</sub> =	-0.00872	a <sub>17</sub> =	-4.13692E-4
a <sub>18</sub> =	0.0072	a <sub>18</sub> =	-0.00341	a <sub>18</sub> =	0.00134
a <sub>19</sub> =	-7.32069E-4	a <sub>19</sub> =	0.00788	a <sub>19</sub> =	0.00436
a <sub>20</sub> =	-0.00402	a <sub>20</sub> =	-0.00255	a <sub>20</sub> =	-8.02781E-4
a <sub>21</sub> =	-0.06691	a <sub>21</sub> =	-0.06572	a <sub>21</sub> =	-0.06626
a <sub>22</sub> =	0.00219	a <sub>22</sub> =	0.00611	a <sub>22</sub> =	0.00315
a <sub>23</sub> =	-0.0183	a <sub>23</sub> =	-0.00149	a <sub>23</sub> =	-0.00407
a <sub>24</sub> =	0.00132	a <sub>24</sub> =	0.0011	a <sub>24</sub> =	0.00122
a <sub>25</sub> =	-0.01241	a <sub>25</sub> =	0.00725	a <sub>25</sub> =	0.00278

### 7.2.1. Comparación entre Zemax y nuestros resultados.

Para comprobar nuestros resultados con este dispositivo realizamos una comparación con el software de diseño óptico Zemax. El parámetro a comparar es el porcentaje de error relativo que tiene el sistema. Para hacer la comparación utilizamos los resultados de la aberración transversal. Zemax nos da el pico-valle el cual es usado para determinar el error absoluto y el error relativo generado. La lente simulada en Zemax tiene una superficie plana y otra con un radio de curvatura de 16.48mm ±0.05mm, la distancia desde la fuente de luz a la superficie de la lente es de 137.48mm ±0.05mm. El espesor en el centro de la lente es de 4.03mm, la distancia de la segunda superficie de la lente a la rejilla fue de 17.72mm. Con estos parámetros se calcula la pupila de entrada n 19.39mm que coincide con el diámetro de la

lente, el índice de refracción de la lente es 1.52 y la longitud de onda usada es de 478.21nm. La gráfica de la aberración transversal obtenida con estos parámetros en Zemax y experimentalmente se muestra en la Fig. 7.4.

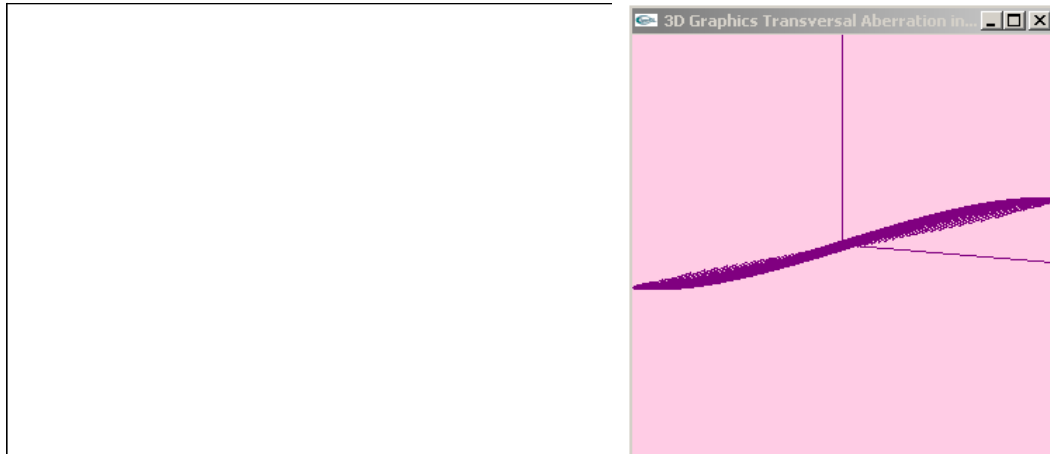


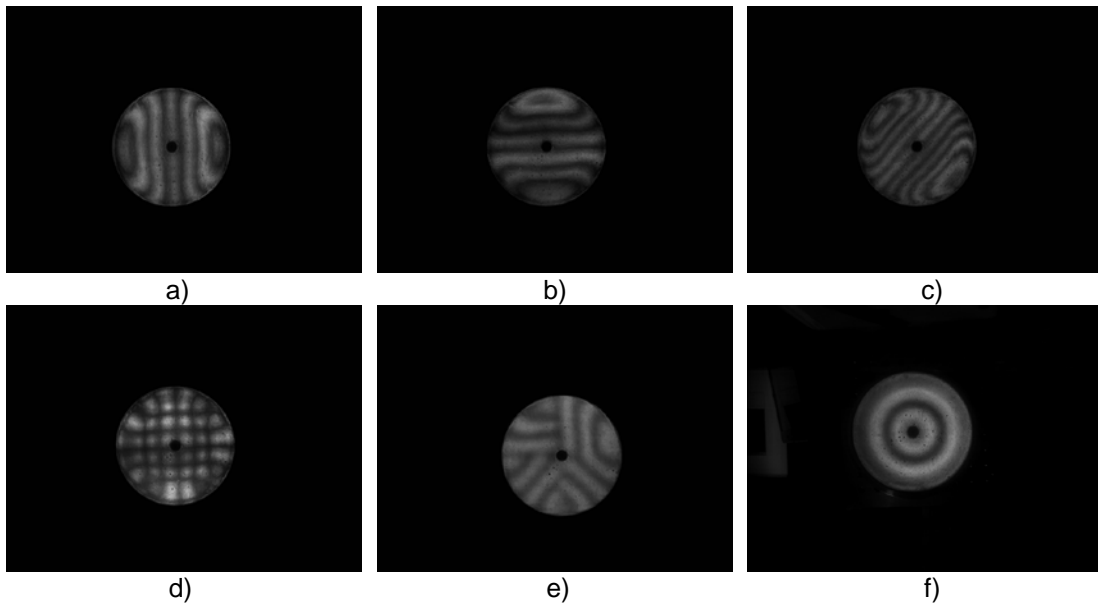
Fig. 7.4. Gráficas para aberración transversal en Zemax y aberración transversal con el software desarrollado.

Comparamos punto a punto las gráficas y podemos decir que la aberración transversal obtenida con nuestro programa y con Zemax son similares, tomamos el valor dado por Zemax como exacto y el valor del pico-valle obtenido por nuestro programa es el valor medido. El pico-valle dado por Zemax es 5.925mm y el pico-valle obtenido con nuestro sistema es de 5.987mm, computamos la diferencia para obtener el error absoluto de 0.062mm por lo tanto el error relativo es  $0.062\text{mm}/5.925\text{mm}=0.0105\text{mm}$ , aquí el porcentaje de error es 1.05% el cual podemos considerar que es aceptable ya que se probó una lente que no tienen alta precisión.

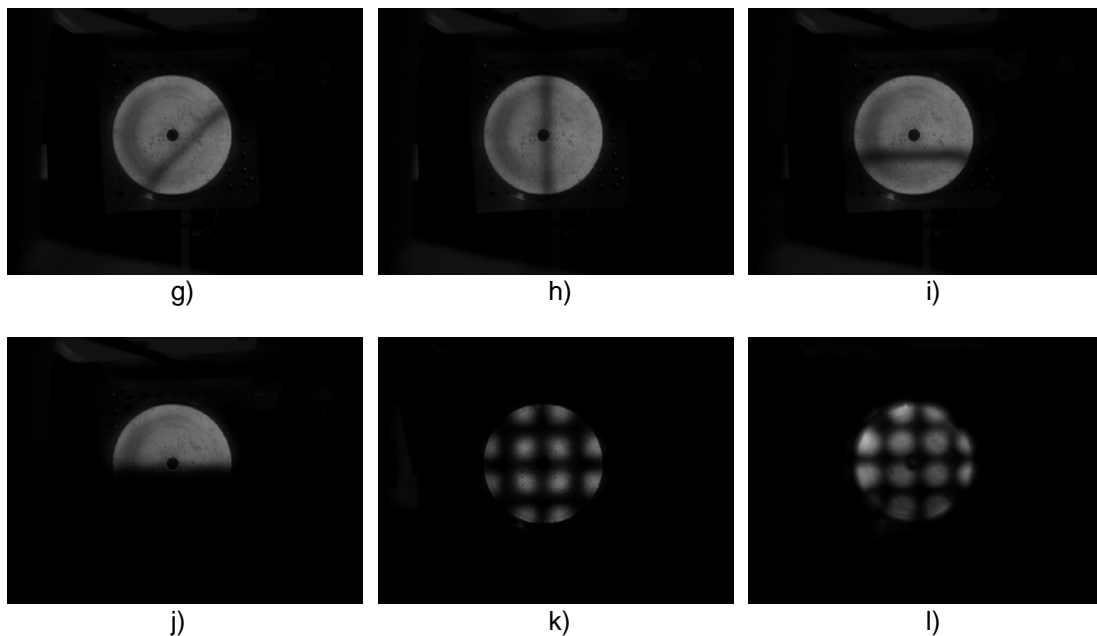
### 7.3. Prueba de Ronchi con LCD.

En la fig. 7.5 se muestran patrones de franjas obtenidos con el arreglo experimental de la sección 5.3 desplegando en la LCD rejillas creadas con el software de la sección 6.4 en la Fig. 7.5. a) se muestra un patrón de franjas

verticales, en la fig. 7.5. b) se muestra un patrón de franjas horizontales, en la fig. 7.5. c) se muestra un patrón de franjas inclinadas, en la fig. 7.5. d) se muestra un patrón de franjas bironchi y en la fig. 7.5. e) se muestra un patrón de franjas múltiple que sirve para demostrar que es posible desplegar cualquier tipo de rejilla en la LCD y obtener un buen contraste en los patrones de franjas obtenidos. En la Fig. 7.5. f) se muestra el patrón de franjas obtenido con una rejilla circular, en la Fig. 7.5. g-i) se observa el patrón de franjas de un alambre en las posiciones inclinado, vertical y horizontal, en la Fig. 7.5. j) se observa el patrón de franjas de la navaja desplegada en la LCD, en la la Fig. 7.5. k) se observa el patrón de franjas de una rejilla bironchi desplegada en la LCD y finalmente en la Fig. 7.5. se muestra el patrón de franjas con una rejilla de Hartmann desplegado en la LCD. Como se observa una sola configuración de Ronchi sirve para implementar varias pruebas ópticas sin la necesidad de utilizar componentes ópticos costosos adicionales, con lo cual la prueba se vuelve muy versátil de tal manera que será posible incluso desplegar rejillas de Ronchi nulas en la LCD.

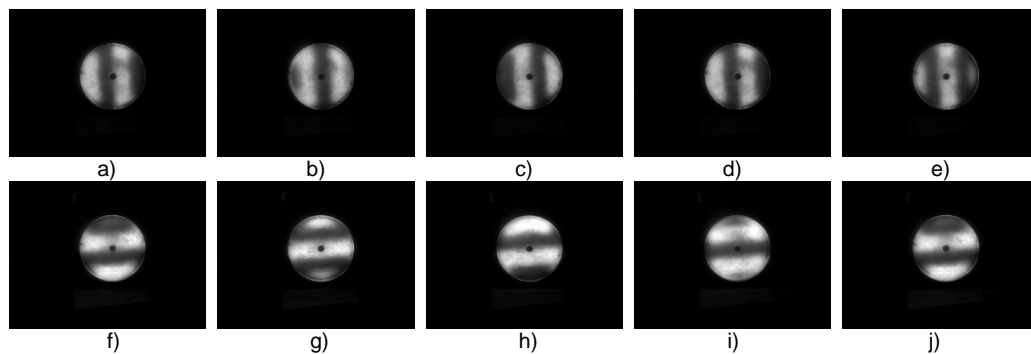






**Fig. 7.5. Franjas obtenidas desplazando diferentes rejillas en la LCD.**

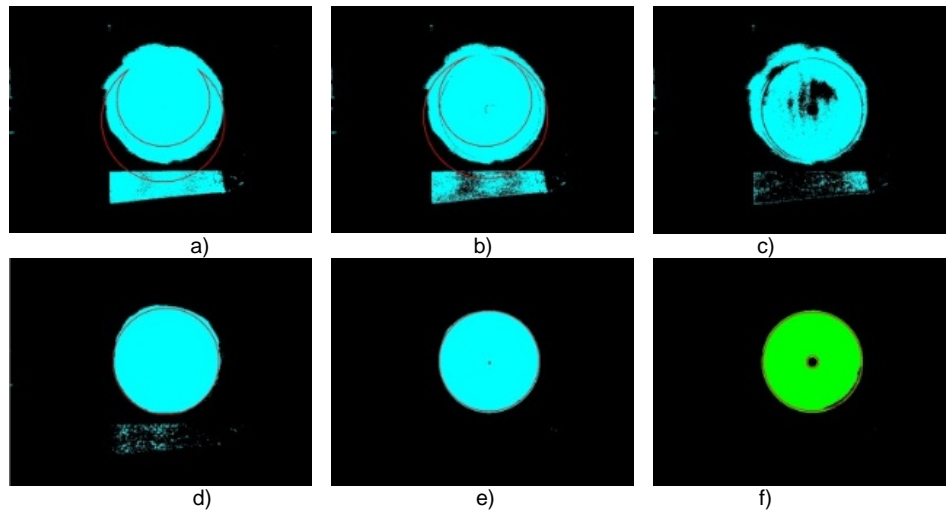
Se usó un espejo de aluminio 6065 como superficie de prueba en el arreglo experimental de la sección 5.3, el espejo tiene un radio de curvatura de 150mm con un orificio central. En la Fig. 7.6 a-e) se muestran 5 ronchigramas verticales obtenidos con el método PSI de Hariharan. En la Fig. 7.6 f-j) se muestran 5 ronchigramas horizontales obtenidos con el método PSI de Hariharan.



**Fig. 7.6. Ronchigramas verticales con desplazamiento de fase de  $90^\circ$ .**

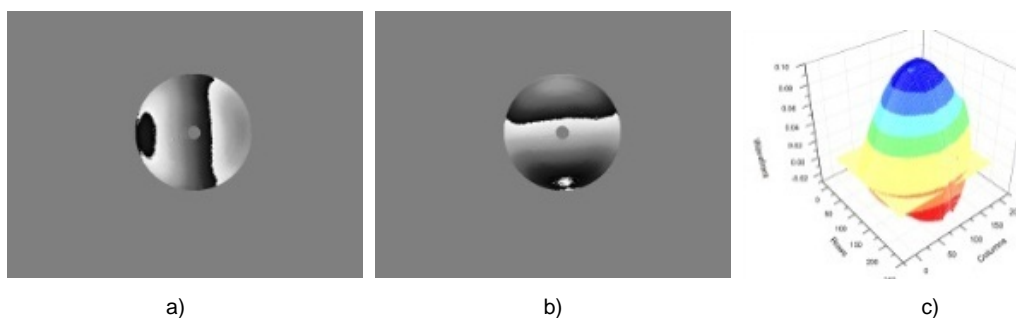
Para fines ilustrativos en la Fig. 7.7 a-f) se muestra en azul la máscara calculada a partir de la ecuación de fondo 3.13 con diferentes niveles de

umbral, como se observa en la Fig. 7.7 e) se seleccionaron bien los píxeles de fase valida, una vez que se tienen bien seleccionados los píxeles de fase valida se calculan los momentos geométricos de la máscara.



**Fig. 7.7. Máscaras de los píxeles de fase valida obtenidas con las ecuaciones de modulación y fondo aplicándoles los momentos geométricos.**

En la Fig. 7.8 a) se muestra la aberración transversal horizontal, en la Fig. 7.8 b) se muestra la aberración transversal vertical y en la Fig. 7.8 c) se muestra el frente de onda.



**Fig. 7.8. Aberraciones transversales horizontal y vertical; y frente de onda**

## 7.4. Comparación de los resultados con Durango®.

Se puede observar que cuando se analizan patrones de franjas obtenidos experimentalmente es muy importante realizar una buena selección de los píxeles de fase válida de los de fase no válida para hacer un análisis correcto.

Para verificar que los datos que da nuestro programa son correctos, hacemos una comparación con el programa Durango®. Probamos un espejo hiperbólico en el interferómetro Zygo y los datos fueron analizados con el programa Durango® usando el algoritmo Hariharan para tomar los interferogramas y realizar el análisis.

En la Fig. 7.9 se muestran 5 interferogramas digitalizados con el programa Durango®, los interferogramas digitalizados con el interferómetro Zygo tienen una resolución de 452x452 píxeles, con el programa Corel Photopaint® le cambiamos el tamaño a 640x480 píxeles llenando los píxeles agregados con el color negro, esto se hace debido a que este tamaño es el admitido por nuestro programa para el análisis.

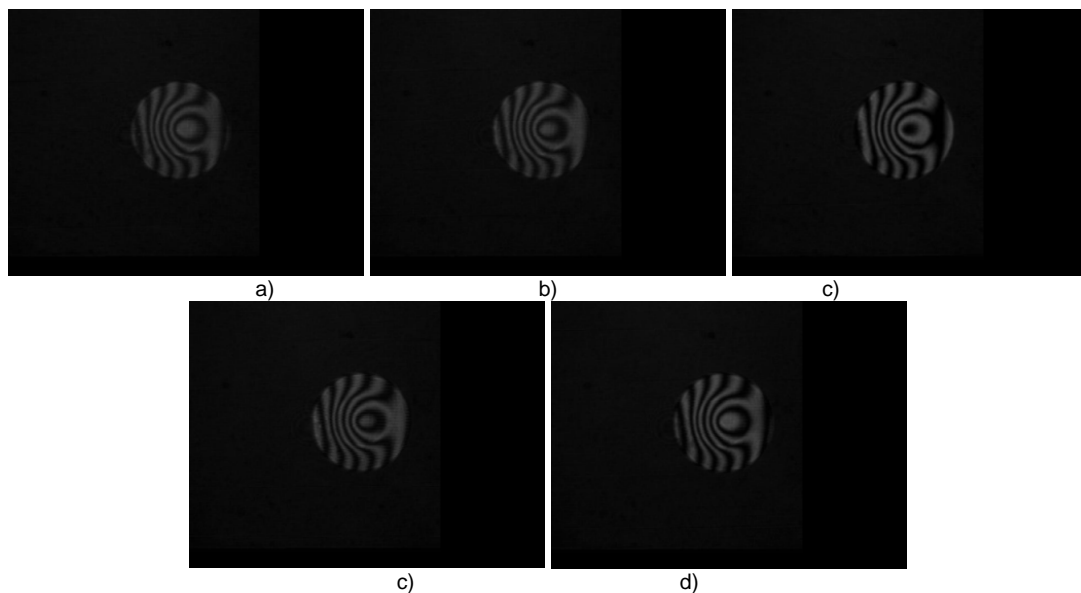


Fig. 7.9. Interferogramas digitalizados con el programa Durango® desde el Zygo.

En la Fig. 7.10 a) se muestra el resultado de la ecuación de fondo, b) se muestra la máscara de los píxeles de fase válida esta máscara se compara con el valor de la ec. 3.13 usando un valor umbral del 25% y en c) se muestra la intensidad de la imagen calculada con Durango®. Las imágenes a) y c) muestran resultados equivalentes. Durango® es un programa comercial en el cual se necesita seleccionar de manera manual la apertura que se analizará con la ecuación de fondo, nuestro programa realiza esta selección de forma automática.

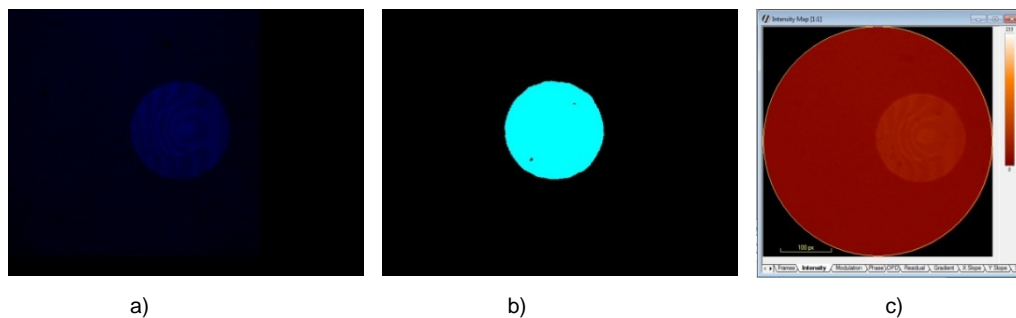


Fig. 7.10. a) Valor de la ecuación de fondo, b) Máscara de los píxeles de fase válida y c) Intensidad de la imagen con Durango®.

En la Fig. 7.11 a) se muestra el resultado obtenido con la ecuación de modulación con nuestro programa con un valor de umbral del 15% de la ec. 3.13 y en b) se muestra el resultado de la ecuación de modulación con el programa Durango®, se puede observar que en la fase válida el resultado es el mismo pero en el fondo en ambos casos se marcan muchos píxeles como fase válida los cuales en el momento de análisis introducirían errores en la recuperación del frente de onda.

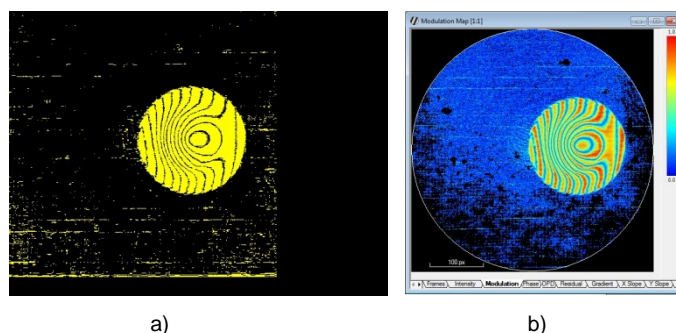
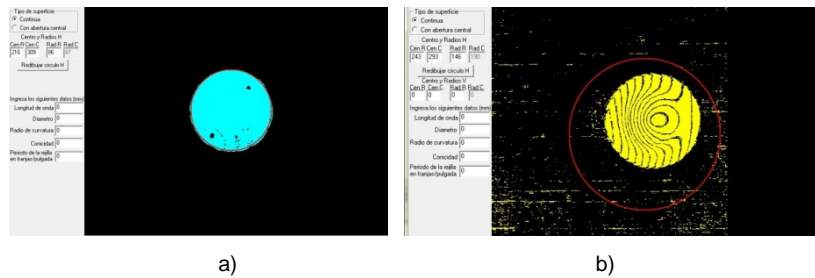


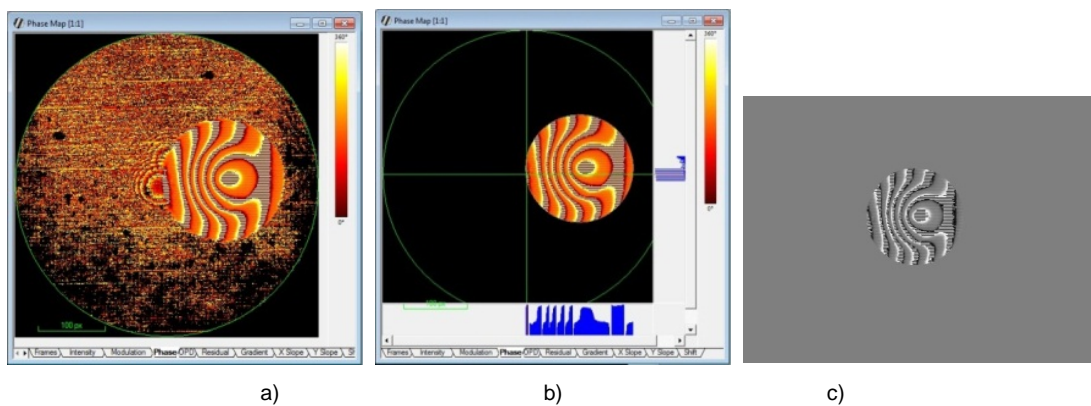
Fig. 7.11 Ecuación de modulación a) Con nuestro programa y b) con Durango®.

En la fig. 7.12 se muestra el círculo en el contorno de la fase valida calculado con los momentos geométricos a partir de la máscara de la ecuación de fondo 3.13, b) se muestra el círculo para el contorno de la fase valida calculada con los momentos geométricos a partir de los resultados de la ecuación de modulación 3.5 se observa que la selección del contorno se realiza con mayor precisión aplicando los momentos geométricos a partir de la ecuación de fondo.



**Fig. 7.12. Círculo en el contorno de la fase valida a) a partir de la ecuación de fondo y b) a partir de la ecuación de modulación.**

En la Fig. 7.13 a) se muestra la fase envuelta calculada con Durango® antes de seleccionar la apertura para el análisis, b) se muestra la fase envuelta calculada con Durango® después de seleccionar manualmente la apertura para el análisis y c) se muestra la fase envuelta calculada con nuestro programa seleccionando la apertura para el análisis con la ecuación de fondo y los momentos geométricos.



**Fig. 7.13. Fase envuelta a) calculada con Durango® antes de seleccionar la apertura b) calculada con Durango® antes de seleccionar la apertura y c) con nuestro programa.**

En la Fig. 7.14 a) se muestra la gráfica del frente de onda realizada con Origin 8.0® obtenido después de desenvolver la fase con el método de Kreis con nuestro programa y b) se muestra el frente de onda obtenido con Durango®. Como podemos ver en las imágenes hasta este punto los resultados cualitativos son prácticamente iguales con lo cual comprobamos que nuestro programa da buenos resultados cualitativos.

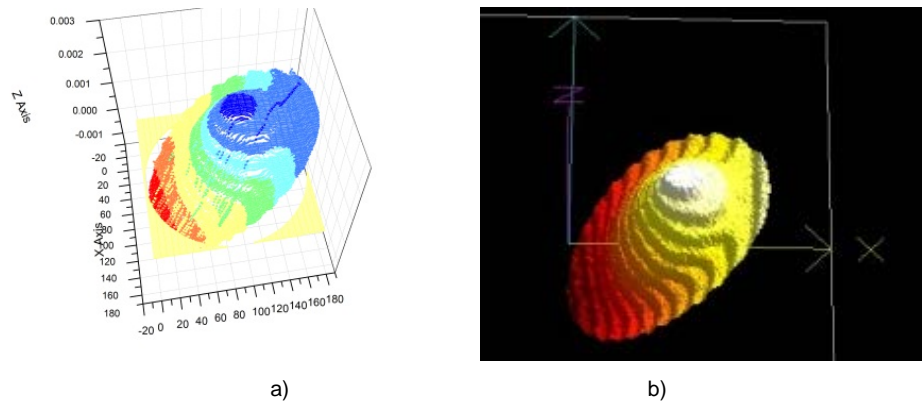


Fig. 7.14. Frente de onda a) con nuestro programa y b) con Durango®.

## 7.5. Error Pico-Valle, rms y ajuste con los polinomios de Zernike.

Para el frente de onda de la Fig. 7.14. a, se realizó un ajuste de grado 5 con los polinomios de Zernike descritos en la sección 2.7. y los datos se muestran en la Tabla 7.2.

Tabla 7.2. Coeficientes de Zernike ajustados

Polinomio		Coefficiente recuperado
Z <sub>1</sub> Piston	1	0.000212933400436882
Z <sub>2</sub> Tilt en z	2x	0.000788371527277764
Z <sub>3</sub> Tilt en Y	2y	0.000114074124597345
Z <sub>4</sub> Defoco	$\sqrt{3}(2\rho^2 - 1)$	-6.03435805278897e-005
Z <sub>5</sub> Astigmatismo primario a 0°	$2\sqrt{6}xy$	-0.00038229658516835
Z <sub>6</sub> Astigmatismo secundario a 45°	$\sqrt{6}(x^2 - y^2)$	-3.36741209253411e-005
Z <sub>7</sub>	$\sqrt{8}y(3\rho^2 - 2)$	1.1673112179161e-005

$Z_8$	$\sqrt{8}x(3\rho^2 - 2)$	1.07538248070745e-006
$Z_9$	$\sqrt{8}y(3x^2 - y^2)$	-2.97572958655069e-005
$Z_{10}$	$\sqrt{8}x(x^2 - 3y^2)$	-2.29581210040378e-006
$Z_{11}$	$\sqrt{5}(6\rho^4 - 6\rho^2 + 1)$	-2.79148059229673e-006
$Z_{12}$	$\sqrt{10}(x^2 - y^2)(4\rho^2 - 3)$	1.88982361417763e-006
$Z_{13}$	$2\sqrt{10}xy(4\rho^2 - 3)$	0.000236207257807025
$Z_{14}$	$\sqrt{10}(\rho^4 - 8x^2y^2)$	-4.96216031892583e-006
$Z_{15}$	$4\sqrt{10}xy(x^2 - y^2)$	-1.65408162708902e-006
$Z_{16}$	$\sqrt{12}x(10\rho^4 - 12\rho^2 + 3)$	-5.23882754872814e-006
$Z_{17}$	$\sqrt{12}y(10\rho^4 - 12\rho^2 + 3)$	1.75711896259023e-006
$Z_{18}$	$\sqrt{12}x(x^2 - 3y^2)(5\rho^2 - 4)$	1.79216549107159e-006
$Z_{19}$	$\sqrt{12}y(x^2 - 3y^2)(5\rho^2 - 4)$	-7.04481601686465e-006
$Z_{20}$	$\sqrt{12}x(16x^4 - 20x^2\rho^2 + 5\rho^4)$	-4.55718621961389e-006
$Z_{21}$	$\sqrt{12}y(16y^4 - 20y^2\rho^2 + 5\rho^4)$	5.29348541869668e-007

De estos coeficientes se realiza su gráfica respectiva, en Fig. 7.15 se observa dicha gráfica las unidades están dadas en milímetros, como se observa la forma del frente de onda obtenida con nuestro programa a partir de los coeficientes de Zernike ajustados es igual a la de la Fig. 7.14 a).

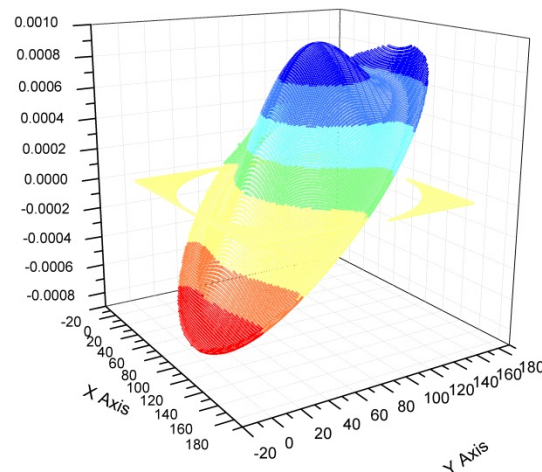


Fig. 7.15. Frente de onda obtenido a partir de los coeficientes ajustados de Zernike

Se obtiene el error Pico-Valle y el error rms del frente de onda, se obtiene antes de realizar el ajuste con los polinomios de Zernike y después de realizar el ajuste con los polinomios, también se comparan con los datos encontrados con Durango®.

**Tabla 7.3. Comparación de los errores Pico-Valle y rms.**

Error	Durango®	Frente de onda real antes de ajustar.	Frente de onda después de ajustar
Pico-Valle	<b>1.936 <math>\mu\text{m}</math></b>	<b>2.079 <math>\mu\text{m}</math></b> Diferencia con Durango® (-0.143 $\mu\text{m} \approx \lambda / 4.4$ )	<b>1.774 <math>\mu\text{m}</math></b> Diferencia con Durango® (0.162 $\mu\text{m} \approx \lambda / 3.9$ )
rms	<b>0.5058 <math>\mu\text{m}</math></b>	<b>0.5717 <math>\mu\text{m}</math></b> Diferencia con Durango® (-0.0659 $\mu\text{m} \approx \lambda / 9.6$ )	<b>0.5036 <math>\mu\text{m}</math></b> Diferencia con Durango® (0.0022 $\mu\text{m} \approx \lambda / 287$ )

Con los datos del error Pico-Valle, del error rms y de la gráfica de los coeficientes de Zernike podemos decir que cualitativamente los resultados cuantitativos obtenidos con nuestro programa son aceptables.

## 7.6. Conclusión.

Se muestran los resultados de como con la LCD se pueden implementar varias pruebas con la configuración de la prueba de Ronchi sin necesidad de agregar más elementos ópticos.

También se muestran los resultados al probar elementos ópticos con la prueba de Ronchi con motores a pasos, sus polinomios de Zernike.

Se muestran los resultados al probar elementos ópticos con la LCD en la prueba de Ronchi y analizar los patrones con nuestro software desarrollado a los cuales se les hace su comparación con el software comercial Durango®, obteniendo muy buenos resultados cualitativos y cuantitativos.



Se muestran los coeficientes de Zernike obtenidos después de realizar el ajuste con nuestro programa y la gráfica obtenida con los coeficientes para comprobar que los coeficientes obtenidos son correctos.

Se obtienen los errores Pico-valle y rms, del frente de onda real obtenido con nuestro programa antes y después de realizar el ajuste de los polinomios de Zernike y se comparan con los del programa Durango® para poder comprobar nuestros resultados cuantitativos.

## *Capítulo 8*

### *8. Conclusiones.*

Se dio una introducción del fenómeno de interferencia, de la Interferometría de desplazamiento de fase PSI y sus ventajas y métodos, se explicó brevemente la prueba de Ronchi. También se definieron los momentos geométricos los cuales usamos para detectar la fase válida de los patrones de franjas digitalizados. Se explican los métodos de desenvolvimiento de fase. Y dan los polinomios de Zernike usados en el ajuste de los datos del frente de onda.

También se explicó que son los píxeles de fase válida y la importancia de saber diferenciarlos de los de fase no válida, por lo que se explicó la ecuación de Schreiber, y se describió como se obtuvieron las ecuaciones de fondo y se muestran las ecuaciones de fondo que se encontraron para algunos algoritmos PSI. Estas ecuaciones de fondo junto con los momentos geométricos son una herramienta poderosa en la detección de píxeles de fase válida cuando se analizan patrones de franjas obtenidos experimentalmente.

Se describió el funcionamiento básico de la LCD, se dieron sus principales características, se dieron las pruebas que se pueden implementar con la LCD, se explicó la resolución de la LCD y como obtener los periodos por franja mínimos para diferentes incrementos de paso cuando se usa la técnica PSI. Con la LCD se evita el uso de varias rejillas de Ronchi costosas y vuelve la prueba más versátil ya que se pueden desplegar rejillas de Ronchi con diferente paso de fase.

Se mostraron los dispositivos usados para probar sistemas ópticos convergentes, usando los métodos mencionados para el análisis cualitativo y cuantitativo de los sistemas ópticos bajo prueba.

Se explicó la prueba de Ronchi implementada con la técnica de PSI con motores a pasos y se describe el arreglo experimental de la configuración de la prueba de Ronchi con la LCD para implementar la prueba de Foucault, la prueba del alambre, la prueba de Hartmann, la prueba de Ronchi con rejilla bironchi, con rejilla circular e incluso con rejilla de Ronchi nula, todas estas pruebas se pueden implementar sin usar ningún elemento óptico adicional.

También se explicó brevemente la forma de utilizar los programas desarrollados para desplegar rejillas en la LCD así como para hacer el análisis de patrones de franjas obtenidos con la técnica PSI.

Se implementó un montaje experimental con varias pruebas de interferometría. Se demuestra que un LED como fuente de iluminación puede ser utilizado sin problemas en la prueba de Ronchi obteniendo un buen contraste en las franjas. Tenemos la ventaja de generar y mostrar muchas rejillas Ronchi con el software desarrollado en la pantalla LCD con diferente periodo haciendo la prueba más versátil. Con la pantalla LCD es incluso posible implementar el interferómetro de difracción punto, la prueba de filo de la navaja, la prueba de Ronchi, la prueba bironchi y la prueba de Ronchi nula. Se muestran los coeficientes de Zernike obtenidos después de realizar el ajuste con nuestro programa y la gráfica obtenida con los coeficientes para comprobar que los coeficientes obtenidos son correctos.

Se obtienen los errores Pico-valle y rms, del frente de onda real obtenido con nuestro programa antes y después de realizar el ajuste de los polinomios de Zernike y se comparan con los del programa Durango® para poder comprobar nuestros resultados cuantitativos.

## Referencias.

1. J. Sánchez-Paredes, G. Silva-Ortigoza, J. Castro Ramos and José Sasian, “ $1\lambda$  Ronchi tester to obtain the wave front aberration in converging optical systems using phase shifting interferometry”, *Opt. Pura Apl.*, Vol.45 No. (4), pp 461-473 (2012).
2. J. Sánchez-Paredes, G. Silva-Ortigoza, J. Castro Ramos and S. Vázquez y Montiel “A Set of Lateral Shear Optical Tests with a Liquid Crystal Display” *Opt. Pura Apl.*, Vol. 46 No. (3), pp 205-214 (2013).
3. J. Sánchez-Paredes, J. Castro-Ramos and S. Vázquez-Montiel, “A comparison between 2D phase unwrapping techniques”, in *Interferometry XIII: Techniques and Analysis*, edited by Katherine Creath, Joanna Schmit, Proc. of SPIE Vol. 6292, 2006.
4. A. Cornejo-Rodríguez “Ronchi Test” in *Optical Shop Testing*, 3rd ed., pp.317-360, Chap. 9, Ed. D. Malacara (John Wiley, 2007).
5. Vasco Ronchi, “Forty Years of History of a Grating Interferometer”, *Applied Optics*, Vol. 3 Issue 4, pp.437-451 (1964).
6. J. M. Núñez-Alfonso, A. Cordero-Dávila, S. Vergara-Limon and J. Cuautle-Cortés, “Improved Ronchi Tester”, *Appl. Opt.*, Vol. 40, No. 4, pp. 501-505, Feb. 2001.
7. M. Mora González and N. Alcalá Ochoa, “The Ronchi test with LCD grating”, *Opt. Commun.* 191 (4-6), 203-207 (2001).
8. J. Castro Ramos, S. Vázquez-Montiel and A. Padilla-Vivanco, “Phase shifting interferometry by using a LCD and bironchi test”, *Proc. of SPIE* Vol. 5662, 639-645 (2004).
9. Chao-Wen Liang and Chien-Fu Ou, “Phase Shifting Grating-Slit Test Utilizing a Liquid Crystal Display”, *Optical Review*. Vol. 16, No. 4, 476-478.

10. William S. Meyers and H. Philip Stahl, "Contouring of a free oil surface", *Proc. SPIE* Vol. 1755, pp. 84-94, Interferometry: Techniques and Analysis, February 5, 1993.
11. K. Omura and T. Yatagai, "Phase measuring Ronchi test", *Appl. Opt.* Vol. 27, 523-528 (1988).
12. P. Hariharan, B. F. Oreb and T. Eiju, "Digital phase shifting interferometry: a simple error-compensating phase calculation algorithm", *Appl. Opt.* Vol. 26, No. 13, pp. 2504, July 1987.
13. Cheng Y. and J.C. Wyant, "Phase Shifter Calibration in Phase Shifting Interferometry", *Appl. Opt.*, 24, 3049, 1985.
14. P. Hariharan, B. F. Oreb and T. Eiju, "Digital Phase shifting interferometry, a simple error compensating phase calculation", *Appl. Opt.* 26, 2504, 1987.
15. Horst Schreiber and John H. Bruning, "Phase shifting interferometry", in *Optical Shop Testing*, Ch. 14, pp. 547-666, 3rd ed., Ed. D. Malacara (John Wiley, 2007).
16. Mukundam R. and Ramakrishnan K., "Moment functions in image analysis", World Scientific Publishing Co. Pte. Ltd, Singapore, 1998, pp. 9-37, 127-128.
17. Kreis T., "Digital holographic interference-phase measurement using the Fourier-transform method" *J. Opt. Soc. Am. A*, 3, 847-855 (1986).
18. Dennis C. Ghiglia and Louis A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods", Vol. 11, No.1, pp. 107-117, *J. Opt. Soc. Am. A.*, (1994).
19. Durango software is a trademark of diffraction international® in: <http://www.diffraction.com/>, last visit on June 2013.
20. Max Born and Emil Wolf, *Principles of optics*, Ch. 7, 7<sup>th</sup> edition, Cambridge.
21. Hecht, "Óptica", Ch. 2, 5, 9, Tercera edición, Addison Wesley.
22. M. V. Mantravadi and D. Malacara, "Newton, Fizeau, and Haidinger Interferometers", in *Optical Shop Testing* Ch. 1, 3rd ed., Ed. D. Malacara (John Wiley, 2007).

23. Paul F. Forman, Zygo Corporation, "*The Zygo interferometer System*", Proc. of SPIE, Vol. 0192, Interferometry, ed. George W. Hopkins (Aug 1979).
24. Yeou-Yen and James C. Wyant, "*Phase shifter calibration in phase shifting interferometry*", Applied Optics, Vol. 24, No. 18, 15 September 1985.
25. J. H. Bruning, D. R. Herriott, J. E. Gallagher, D. P. Rosenfeld, A. D. White, and D. J. Brangaccio, "*Digital wavefront measuring interferometer for testing optical surfaces and lenses*", Applied Optics, Vol. 13, No. 11, November 1974.
26. Daniel Malacara, Manuel Servin, "*Interferogram analysis for optical testing*", Marcell Decker, USA, 1998.
27. Wyant James , Zernikes eqs. in [http://www.optics.arizona.edu/jcwyant/Zernikes/ZernikeEquations .htm](http://www.optics.arizona.edu/jcwyant/Zernikes/ZernikeEquations.htm).
28. Ritchey G. W., "On the Modern Reflecting Telescope and the Making and Testing of Optical Mirrors," *Smithson. Contrib. Knowl.*, 34, 3 (1904)
29. J. Castro-Ramos and Jose Sasian, "*Automatic phase shifting Ronchi tester with a square Ronchi ruling*", Interferometry XII: Applications, ed. Wolfgang Osten, Erik Novak, Proc. SPIE Vol. 5532, 2004.
30. A. Cordero-Dávila, E. Luna-Aguilar, S. Vázquez-Montiel, S. Zarate-Vázquez, and M. E. Percino-Zacarias "*Ronchi test with a square grid*", Appl. Opt. Vol. **37**, 672-675 (1998).
31. D. Malacara and A. Cornejo, "*Null Ronchi Test for Aspherical Surfaces*", Applied Optics Vol. 13, No. 8, August, 1974.
32. W. S. Meyers, H. P. Stahl, "Contouring of a free oil surface", *Proc. SPIE* 1755, 84-94 (1993).
33. A. Cornejo, D. Malacara, "*Ronchi Test of Aspherical Surfaces, Analysis, and Accuracy*", Appl. Opt. Vol. 9(8),1987-1901, (1970).