

**DESARROLLO DE UNA ESTRATEGIA PARA LA CORRECCIÓN DE
FASE DISCONTINUA DE OBJETOS CON TOPOGRAFÍA COMPLEJA
BASADA EN ALGORITMOS DE FASE TEMPORAL**

Andrea Paola Rodríguez Hernández
código:2092765

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS
ESCUELA DE FÍSICA
BUCARAMANGA
2016

***DESARROLLO DE UNA ESTRATEGIA PARA LA CORRECCIÓN DE
FASE DISCONTINUA DE OBJETOS CON TOPOGRAFÍA COMPLEJA
BASADA EN ALGORITMOS DE FASE TEMPORAL***

Andrea Paola Rodríguez Hernández
código: 2092765

Trabajo de grado para optar al título de:
Físico

Director:
Jaime Enrique Meneses Fonseca, Ph.D.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS
ESCUELA DE FÍSICA
BUCARAMANGA
2016

Agradecimientos

Al Doctor Jaime Enrique Meneses Fonseca por su paciencia, orientación, apoyo y tiempo durante la realización de este trabajo, por transmitirme su pasión por la investigación y por la Física. Sin él nada de esto sería posible.

Al la Universidad Industrial de Santander por la formación y todos los conocimientos que pude adquirir en ella.

A los docentes de la escuela de física con los cuales tuve la oportunidad de compartir, y por los conocimientos transmitidos.

Al grupo de óptica y tratamiento de señales por acogerme y prestarme sus instalaciones para la realización de este trabajo.

Dedicatoria

A mis padres Elizabeth Hernández y Ludwing Rodriguez en especial a mi madre por animarme a siempre seguir adelante y apoyar mis decisiones, a mis hermanos Adriana y Pablo por siempre estar ahí, a toda mi familia. Y a lucho solo me queda agradecerle por considerarme una hija más y por su apoyo.

*A **mi cosita Andrés González** por orientarme en este proceso, por transmitirme sus conocimientos, por su compañía, sus estímulos, por sus regañitos de vez en cuando y por ser tan especial conmigo.*

A la Karima por sus ideas de diseñadora que fueron muy útiles en el desarrollo de la parte experimental, por su compañía hasta largas horas de la noche y claro por acompañarme los sábados en el laboratorio con Camila y ruper.

A las personas que a lo largo de estos años conocí en el camino de las cuales pude aprender algo, a Manu, Joseph y Oskis por su amistad, compañía en toda la carrera y los trasnochos para estar donde estamos hoy.

Índice general

INTRODUCCIÓN	16
1. MARCO DE REFERENCIA	18
1.1. CLASIFICACIÓN DE LAS TÉCNICAS DE RECONSTRUCCIÓN TRI-DIMENSIONAL	18
1.1.1. Técnicas de contacto	18
1.1.2. Técnicas de no contacto	19
1.2. PROYECCIÓN DE FRANJAS	21
1.2.1. Etapas	22
1.2.2. Método de corrimiento de fase	23
1.3. CORRECCIÓN DE FASE	25
1.3.1. Desarrollo de fase espacial	25
1.3.2. Desarrollo de fase temporal	26
1.4. PROCEDIMIENTO EXPERIMENTAL DE R3D	28
1.5. PLANTEAMIENTO DEL PROBLEMA	29
2. ALGORITMOS DE DESENVOLVIMIENTO DE FASE TEMPORAL TPU	34
2.1. DESENVOLVIMIENTO DE FASE TEMPORAL LINEAL (LTPU)	35
2.1.1. Simulación con objeto plano	40
2.1.2. Simulación con objeto discontinuo	45
2.1.3. Análisis de resultados	47
2.2. DESENVOLVIMIENTO DE FASE TEMPORAL EXPONENCIAL (ETPU)	48
2.2.1. Simulación con un objeto plano	53
2.2.2. Simulación con objeto discontinuo	58
2.2.3. Análisis de resultados	60
2.3. DESENVOLVIMIENTO DE FASE TEMPORAL EXPONENCIAL INVER-SO (RETPU)	61
2.3.1. Simulación con un objeto plano	66
2.3.2. Simulación con un objeto discontinuo	70
2.3.3. Análisis de resultados	72
2.4. DESENVOLVIMIENTO DE FASE TEMPORAL GENERALIZADO (GT-PU)	73
2.4.1. Simulación con objeto plano	76

2.4.2.	simulación de un objeto discontinuo	80
2.4.3.	Análisis de resultados	82
2.5.	ANÁLISIS DEL COMPORTAMIENTO DE LOS ALGORITMOS	82
3.	ANÁLISIS EXPERIMENTAL DE LOS ALGORITMOS PROPUESTOS	89
3.1.	MONTAJE EXPERIMENTAL	89
3.2.	ANÁLISIS PRELIMINAR	90
3.3.	ALGORITMO LINEAL	93
3.4.	ALGORITMO EXPONENCIAL BASE 2	95
3.5.	ALGORITMO INVERSO BASE 2	96
3.6.	ALGORITMO GENERALIZADO	97
3.7.	ANÁLISIS DE RESULTADOS	98
3.8.	SELECCIÓN DEL ALGORITMO TEMPORAL EN FUNCIÓN DEL OBJETO	100
4.	CONCLUSIONES	104
	BIBLIOGRAFIA	104

Indice de tablas

1.1. Técnicas Pasivas y algunos ejemplos.	19
1.2. Técnicas activas y algunos ejemplos.	21

Índice de figuras

1.1. Clasificación de las técnicas de medición tridimensional.	18
1.2. Montaje experimental de la técnica de proyección de franjas.	21
1.3. Patrón de franjas con perfil sinusoidal.	22
1.4. Etapas de la técnica de proyección de franjas.	23
1.5. Patrón de franjas desfasados $\pi/2$ radianes.	24
1.6. Fase envuelta y su respectivo perfil.	25
1.7. (a) Distribución fase continua, (b) Corte transversal de la fase.	26
1.8. Desenvolvimiento de fase temporal.	26
1.9. Fase discontinua para cada paso de proyección.	28
1.10. Fase continua para cada paso de proyección.	28
1.11. Discontinuidad de las franjas sobre un objeto.	30
1.12. Perdida de información en el cálculo de la fase.	30
1.13. Propagación de error en la fase continúa.	31
1.14. Perfil fase continua para puntos de partida diferente. Fuente: [3]	32
1.15. Línea central	32
1.16. Problema de objetos aislados	33
2.1. Mapa de fase discontinua $t = 1$	36
2.2. Mapa de fase discontinua $t = 2$	36
2.3. Resta de los mapas de fase ($t = 2$) - ($t = 1$).	36
2.4. valores de las 2π discontinuidades	36
2.5. Fase continua del mapa $t=2$	37
2.6. Algoritmo lineal	37
2.7. Gráfica del perfil de las fases y su diferencia	38
2.8. Gráfica del perfil de las fases y su diferencia	39
2.9. Primeros cuatro patrones de franjas.	41
2.10. Fases discontinuas	41
2.11. Diferencia $t=2$ y $t=1$	41
2.12. Valores "d" para $t=2$	42
2.13. Fase continua para $s=32$, algoritmo lineal.	42
2.14. Tasa de éxito para ruido bajo lineal.	42
2.15. Primeros cuatro patrones de franjas.	43
2.16. Fases discontinuas	43
2.17. Diferencia $t=2$ y $t=1$	44

2.18. Valores "d" para $t=2$	44
2.19. Fase continua para $s=32$	44
2.20. Tasa de éxito para ruido alto lineal.	45
2.21. Primeras 4 franjas proyectadas	45
2.22. Primeras 4 fases discontinuas	45
2.23. Diferencia entre el mapa $t=2$ y $t=1$	46
2.24. Valores "d" para $t=2$	46
2.25. Fase continua para $s=32$	46
2.26. Forma de los objetos simulados	47
2.27. Valores de fase empleados	48
2.28. Resta de los mapas de fase $(t = 2) - 2 \cdot (t = 1)$	50
2.29. Valores de las discontinuidades "d"	50
2.30. Fase continua $t=2$	50
2.31. Fase continua $t = 32$	51
2.32. Funcionamiento del algoritmo exponencial según pendientes	51
2.33. φ vs t	52
2.34. Algoritmo Exponencial base 2	52
2.35. Primeros cuatro patrones de franjas exponencial base 2.	53
2.36. Primeras fases discontinuas exponencial base 2	53
2.37. Fase continua 1 multiplicada por la base "2"	53
2.38. Fase continua 1 multiplicada por la base "2"	54
2.39. Valor de las discontinuidades "d"	54
2.40. Fase continua para $s=2$	54
2.41. Fase continua para $s=32$	55
2.42. Tasa de éxito para ruido bajo exponencial.	55
2.43. Primeros cuatro patrones de franjas exponencial base 2.	55
2.44. Primeros cuatro fases discontinuas exponencial base 2.	56
2.45. Fase continua 1 multiplicada por la base	56
2.46. Diferencia entre $t = 2 - 2(t = 1)$	56
2.47. Valores "d"	57
2.48. Fase continua para el mapa $s=2$	57
2.49. Fase continua para el mapa $s=32$	57
2.50. Tasa de éxito para ruido alto en algoritmo exponencial.	58
2.51. Primeras 4 franjas proyectadas exponencial	58
2.52. Primeras 4 fases discontinuas	58
2.53. 2 veces la fase continua 1	59
2.54. Diferencia mapa - 2 veces mapa 1	59
2.55. Valor de las "d"	59
2.56. Fase continua mapa 2	60
2.57. Fase absoluta con algoritmo exponencial.	60
2.58. resta del mapa s y $s-1$	62
2.59. Valor de las 2π discontinuidades	62
2.60. fase continua de la diferencia $\varphi_d(32) - \varphi_d(31)$	62
2.61. Proceso para obtener fase continua 32,30	63

2.62. fase continua de 32 por algoritmo inverso	63
2.63. secuencia de proyección algoritmo inverso.	64
2.64. Algoritmo inverso	65
2.65. Franjas con error introducido, algoritmo inverso.	66
2.66. Fases discontinuas algoritmo inverso ruido bajo.	66
2.67. Diferencia entre el mapa 32 y 31.	67
2.68. Fase continua de la diferencia entre el mapa 32 y 31.	67
2.69. Diferencia entre el mapa 31 y 30.	67
2.70. Fase continua 32-30	67
2.71. Fase continua 32-0	68
2.72. Tasa de éxito con ruido bajo algoritmo inverso	68
2.73. Franjas con error alto, algoritmo inverso.	68
2.74. Fases discontinuas algoritmo inverso ruido alto.	69
2.75. Diferencia entre el mapa 32 y 31.	69
2.76. Fase continua de diferencia entre mapa 32 y 31 error alto.	69
2.77. Fase continua de diferencia entre mapa 31 y 30 error alto.	69
2.78. Fase continua 32-30 con error alto.	70
2.79. Tasa de éxito ruido alto inverso	70
2.80. Primeras 4 franjas proyectadas inverso	71
2.81. Primeras 4 fases discontinuas inverso	71
2.82. Diferencia mapa 32-31	71
2.83. Fase continua mapa 32-31	71
2.84. Fase continua objeto mapa 32 inverso	72
2.85. diferencia de fase discontinua consecutiva para t y $2t$, $\varphi_d(t, 2t)$	72
2.86. Numero de franjas utilizado algoritmo generalizado	74
2.87. Fases discontinuas algoritmo generalizado	74
2.88. k veces fase continua 1	74
2.89. Fase continua mapa 7	74
2.90. Fase continua en 32 algoritmo generalizado	75
2.91. Algoritmo Generalizado	75
2.92. Franjas con error introducido, algoritmo generalizado.	76
2.93. Fases discontinuas, algoritmo generalizado.	76
2.94. 7 veces fase continua 1.	76
2.95. Fases continua del mapa 7.	77
2.96. Fases continua del mapa 13.	77
2.97. Fases discontinuas, algoritmo generalizado.	77
2.98. Tasa de éxito ruido bajo, algoritmo generalizado.	78
2.99. Primeros cuatro patrones de franjas generalizado.	78
2.100 Fases discontinuas	78
2.101 Diferencia mapa 7 -1	79
2.102 Fase continua mapa 7.	79
2.103 Fase continua mapa 32 con algoritmo generalizado.	79
2.104 Tasa de éxito ruido alto generalizado	80
2.105 Primeras 4 franjas proyectadas exponencial	80

2.106	Primeras 4 fases discontinuas	80
2.107	Diferencia mapa 7-1 objeto discontinuo	81
2.108	Fase continua del mapa 7	81
2.109	Fase continua objeto mapa 32 generalizado	81
2.110	Tasa de éxito ruido gaussiano con $\sigma = 5$ NG.	83
2.111	Tasa de éxito ruido gaussiano con $\sigma = 30$ NG.	83
2.112	Gráficas φ vs t para cada algoritmo	85
2.113	Comportamiento de los algoritmos variando ω	86
2.114	Comparación de los algoritmos exponenciales	87
2.115	Comparación de los algoritmos inversos	88
3.1.	Configuración experimental del sistema	89
3.2.	Configuración de la CCD.	90
3.3.	fase continua obtenida experimentalmente	91
3.4.	Nivel de ruido de la diferencia experimental e interpolada	91
3.5.	Amplitud de la onda según la intensidad	92
3.6.	Nivel de ruido de $I=0.7$	92
3.7.	Fase interpolada	93
3.8.	Errores experimentales del algoritmo lineal.	94
3.9.	Gráficas para el análisis del error del algoritmo lineal	94
3.10.	Desviación estándar versus modulación.	95
3.11.	Errores experimentales del algoritmo exponencial base 2.	95
3.12.	Gráficas para el análisis del error del algoritmo exponencial	96
3.13.	Errores experimentales del algoritmo inverso base 2.	96
3.14.	Gráficas para el análisis del error del algoritmo inverso	97
3.15.	Errores experimentales del algoritmo generalizado.	97
3.16.	Gráficas del para el análisis del error del algoritmo generalizado.	98
3.17.	Tasa de éxito vs modulación de todos los algoritmos.	98
3.18.	Tasa de éxito vs σ de todos los algoritmos.	99
3.19.	Tasa de éxito vs SNR de todos los algoritmos.	99
3.20.	Objetos utilizados para la obtención de la fase	100
3.21.	Franjas sobre los objetos de estudio	101
3.22.	Mascara de los objetos en estudio	101
3.23.	Modulaciones de los objetos en estudio	102
3.24.	Fase continua del maniquí con algoritmo lineal	102
3.25.	Fase continua del cráneo con algoritmo exponencial	103

RESUMEN

Título: Desarrollo de una estrategia para la corrección de la fase discontinua de objetos con topografía compleja basada en algoritmos de fase temporal¹

Autor: Andrea Paola Rodríguez Hernández²

Palabras claves: Proyección de franjas, Fase, Desenvolvimiento de fase temporal, Reconstrucción tridimensional.

La técnica de proyección de franjas es ampliamente utilizada en metrología óptica ya que permite realizar la reconstrucción tridimensional de objetos proyectando franjas paralelas sobre éstos, las cuales se deforman debido a la topografía de los mismos (los objetos) modificando la fase del patrón proyectado. La técnica utilizada proporciona fases discontinuas, requiriendo así un procedimiento para la corrección de estas; al aplicar el algoritmo de desenvolvimiento de fase espacial para corregir la fase discontinua en objetos que presentan topografía compleja han surgido ciertos problemas impidiendo el desarrollo de procesos automáticos: los objetos generan sombras que producen pérdida de información, se hace necesaria la intervención de un usuario para lograr adquirir la fase continua. Como la obtención de la fase es un paso fundamental para llegar a una reconstrucción tridimensional, surgió la necesidad de encontrar un procedimiento que permitiese corregir la fase discontinua de objetos complejos a partir de algoritmos de desenvolvimiento de fase temporal. Para ello se hizo una revisión de los principales algoritmos de tipo temporal reportados bibliográficamente, realizando un estudio del comportamiento de los mismos fue posible hacer una evaluación de las estrategias de corrección temporal de fase discontinua y su comportamiento asociado a la influencia del ruido. De esta manera se pudo concluir que el algoritmo exponencial creciente a base 2, es el que genera mejores resultados en la obtención de la fase, ya que es menos sensible al ruido.

¹Trabajo de grado

²Facultad de ciencias. Escuela de Física. Director: Dr. Jaime Enrique Meneses Fonseca

ABSTRACT

Title: Developing a strategy for correcting the wrapped phase of complex objects using temporal phase unwrapping algorithm¹

Author: Andrea Paola Rodríguez Hernández²

Keywords: Fringe projection, Phase, Temporal phase unwrapping, Three-dimensional reconstruction.

The fringe projection technique is widely used by optical metrology because it allows the reconstruction of three-dimensional objects by projecting on them parallel strips, which are deformed due to the topography of them, changing the phase of the projected pattern. The technique provides wrapped phases, requiring a method for correcting these. Applying the algorithm of spatial phase unwrapping to correct the discontinuous phase in objects which have complex topography they have emerged certain problems preventing the automated processes of unwrapping: objects generates shadows that produce loss of data, so a user intervention is necessary to acquire the continuous phase. Obtaining the phase is an essential step to get an accurate three-dimensional reconstruction, it became necessary to find a method that would allow correcting the wrapped phase of complex objects from algorithms of temporal phase unwrapping. For this, a research of the main temporal algorithms reported bibliographically was done; doing a study of the behavior of these algorithms was possible to evaluate strategies of temporal phase unwrapping correction and behavior associated with the noise influence. Thus, it was concluded that the growing base 2 exponential algorithm is which generate better results in the continuous phase, because it is less sensitive to noise.

¹Bachelor Thesis

²Facultad de ciencias. Escuela de Física. Director: Dr. Jaime Enrique Meneses Fonseca

INTRODUCCIÓN

La reconstrucción tridimensional de objetos es un elemento tecnológico muy importante en áreas como las ciencias computacionales, la medicina, entretenimiento, entre otras. Por esta razón se han desarrollado diferentes métodos que permiten obtener la forma 3D de los objetos; uno de estos es la técnica de proyección de franjas, la cual ha despertado gran interés debido a sus ventajas de operación de no contacto, rápida medición y procesamiento automático. La técnica consiste en generar un patrón de franjas paralelas que se proyectan con ayuda de un vídeo proyector sobre el objeto a reconstruir, cuando el objeto es plano las franjas poseen una fase definida, pero cuando este presenta diferente topografía las franjas son deformadas según la misma, generando así un cambio en la fase de las franjas. La diferencia entre estos valores de fase contiene información de la forma tridimensional del objeto. Para obtener la fase de los objetos se emplea una función arcotangente, ya que se asume un perfil sinusoidal en las franjas, generando valores acotados entre $-\pi$ a π en la fase. Comúnmente esta fase es llamada fase discontinua ("*wrapped phase*").

Para lograr la reconstrucción del objeto primero es necesario convertir la fase discontinua en continua mediante la adición de valores enteros de 2π , procedimiento llamado desenvolvimiento ("*unwrapping method*"). Al interior del *grupo de óptica y tratamiento de señales de la Universidad Industrial de Santander* se han desarrollado diferentes trabajos en la reconstrucción de objetos como: reconstrucción tridimensional del torso humano a 360° [11], análisis de grietas en tuberías no metálicas para la transportación de hidrocarburos [1], estudio de las características topográficas de la piel [3], entre otros. En los trabajos mencionados anteriormente se ha realizado el desenvolvimiento de la fase implementando el algoritmo de desenvolvimiento espacial (*spatial phase-unwrapping*), el cual consiste en comparar la fase entre píxeles vecinos haciendo una diferencia entre estas, cuando la diferencia toma un valor cercano a 2π el algoritmo asume que se encuentra presente una discontinuidad y le adiciona o sustrae un múltiplo entero de 2π a la fase del píxel.

El problema con el algoritmo surge cuando se tienen objetos con una topografía compleja, debido a que se generan sombras causando pérdida en la continuidad de las franjas. Al obtener la fase discontinua del patrón de franjas en la zona donde se generó la sombra se obtiene pérdida de información y aplicando el algoritmo espacial en la corrección de la fase cuando se llega a una región con pérdida de información, el algoritmo asume esa zona como discontinuidad adicionando valores incorrectos de 2π que se propagan en la dirección del desenvolvimiento, generando una fase continua incorrectamente calculada y

afectando la reconstrucción tridimensional del objeto. Para evitar el problema de propagación del error en la fase, en los trabajos realizados anteriormente en el grupo GOTS se implementó una máscara la cual validaba e invalidaba regiones en donde llevar a cabo el desenvolvimiento, limitando así la posibilidad de un cálculo automático de la fase.

A partir de los problemas encontrados con el algoritmo espacial se dedujo que al implementar un desenvolvimiento de fase temporal (*temporal phase-unwrapping*) estos problemas podrían ser corregidos. El desenvolvimiento temporal de fase consiste en proyectar patrones de franjas a diferentes frecuencias sobre el objeto en estudio. La frecuencia con que se proyectan las franjas varía dependiendo del algoritmo que se emplee para realizar el desenvolvimiento. Existen varios tipos de algoritmos temporales, entre los que se encuentran el lineal, exponencial, inverso y generalizado, donde la fase de cada punto se desenvuelve de forma independiente, sin tener en cuenta información de los vecinos, previniendo así la propagación del error espacialmente. Algunas de las ventajas encontradas al emplear el algoritmo de desenvolvimiento temporal es que permite obtener fase absoluta, además de un cálculo automático de la corrección de fase.

Este trabajo se divide en cinco partes: primero se hace una presentación formal de cada uno de los algoritmos, segundo se muestran gráficamente los pasos que deben llevarse a cabo para obtener la fase continua, tercero se hace una simulación sobre objetos planos y discontinuos usando los algoritmos temporales, en la cuarta parte se analiza el error que presentan los algoritmos, realizando gráficas comparativas por medio del criterio de tasa de éxito, lo que sirve de base para seleccionar el algoritmo que presente un mejor comportamiento; para saber en qué punto de las gráficas se encuentra el sistema implementando en el laboratorio para un objeto específico, se hace un análisis de como la modulación de las franjas afecta el ruido en la fase final. En la parte final del trabajo se selecciona el mejor algoritmo para ciertos objetos.

1. MARCO DE REFERENCIA

La reconstrucción tridimensional (R3D) es el proceso mediante el cual se genera un modelo informático de la apariencia tridimensional 3D (3 dimensiones) de un objeto, desde un conjunto de imágenes bidimensionales (2 dimensiones). La R3D se conforma de ciertas etapas: 1) adquisición de las imágenes 2D, 2) cálculo de la fase del objeto a estudiar con alguno de los métodos existentes, 4) conversión fase en altura, 6) conversión de píxeles a milímetros lo que genera la R3D del objeto.

1.1. CLASIFICACIÓN DE LAS TÉCNICAS DE RECONSTRUCCIÓN TRIDIMENSIONAL

Debido a la gran cantidad de estudios realizados hasta ahora respecto a las técnicas que permiten la medición de la forma 3D ha sido posible la clasificación de la técnica de medición de la forma 3D en dos grandes grupos, los cuales son: métodos de contacto y métodos de no contacto; este último a su vez se divide en dos subgrupos: técnicas pasivas y técnicas activas [5]. (ver figura 1.1).

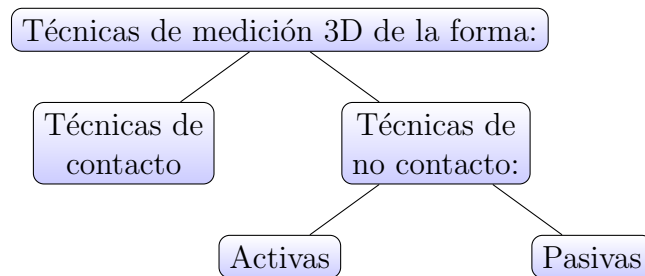


Figura 1.1: Clasificación de las técnicas de medición tridimensional.

1.1.1. Técnicas de contacto

En las técnicas de medición de contacto, el sistema sondea el objeto a través de contacto físico, estando el objeto en contacto o descansando sobre una superficie en el plano. Si el objeto medido no está sobre una superficie plana o no puede descansar de forma

estable sobre una superficie debe ser sostenido por un accesorio.

Un ejemplo de esta técnica es la máquina de medición de coordenadas (*CMM coordinate measuring machine*). Las CMMs han sido utilizadas en el área de fabricación ya que generan mediciones de alta precisión; Sin embargo se hace necesario el contacto con el objeto lo cual podría modificar o dañar su superficie. Otra desventaja de la técnica es que es relativamente lenta debido a que solo mide un punto a la vez.

1.1.2. Técnicas de no contacto

Debido a las limitaciones presentadas con las técnicas de contacto, se desarrollaron diferentes métodos de no contacto. Ya que no es necesario el contacto físico, estas técnicas pueden ser utilizadas para medir objetos blandos o deformables, ampliando así sus áreas de aplicación. Las técnicas de no contacto a su vez han sido clasificadas en dos grupos: técnicas pasivas y activas.

Técnicas pasivas	
Se basan en la iluminación ambiental para la recuperación de la información de profundidad.	
Profundidad de desenfoque	Algunas áreas de la imagen son nítidas y enfocadas mientras otras son borrosas. Los puntos de la escena del objeto crean imágenes con diferente cantidad de desenfoque o borrosidad. Si lo borroso debido al desenfoque puede ser cuantificado, esta información se usa para recuperar la profundidad de la forma [25]. Para obtener una R3D de la escena, generalmente se utilizan dos o más imágenes desenfocadas con diferentes ajustes de enfoque [7]. Desventaja: requiere fuerte información de textura para analizar los efectos de borrosidad, resolución no muy alta [22].
Visión estéreo	Analiza dos o más imágenes de la misma escena desde diferentes perspectivas. Cuando dos cámaras observan una escena 3D desde dos posiciones distintas hay una cantidad de relaciones geométricas entre los puntos 3D y sus proyecciones sobre las imágenes 2D. Estas relaciones se derivan aproximando las cámaras al modelo de cámara de pinhole. Ventajas: bajo costo del hardware, puede medir a gran velocidad, y requiere solo dos cámaras. Desventaja: costo de cálculo de los puntos, debido a regiones con luminancia constante que generan problemas para hacer coincidir únicamente dos puntos.

Tabla 1.1: Técnicas Pasivas y algunos ejemplos.

Técnicas activas
Se desarrollan por la emisión de algún tipo de iluminación codificada sobre objetos para el establecimiento de la correspondencia. Mediante la captura y análisis de la interacción entre el objeto y la iluminación pueden ser obtenidas las coordenadas de los puntos o la información 3D.

<p>Tiempo de vuelo (<i>TOF time of flight</i>)</p>	<p>Las cámaras TOF involucran iluminación activa y rango de entrega (profundidad), midiendo el tiempo necesario para que una señal de luz viaje desde la fuente de la cámara a la escena y vuelva al sensor de la cámara. Estas cámaras se basan en la emisión continua de una señal periódica. Modulando la fase en una frecuencia constante, y grabando la fase con el detector se determina el tiempo usando la diferencia de fase. Las unidades de proyección y grabado son vistas desde el mismo ángulo evitando problemas de obstrucciones. Desventaja resolución del detector baja porque los sensores deben medir información de intensidad y distancia.</p>	
<p>Triangulación láser</p>	<p>El sistema contiene un láser, un detector y un lente para enfocar el haz en el detector. Cuando se realiza la medición el láser emite una luz sobre la superficie del objeto, el cual se refleja desde la superficie y cae sobre el detector a través del lente. Según la posición del haz en el detector y su posición de emisión, la profundidad Z del sensor a la superficie se puede calcular a través de triangulación. La precisión del sistema es relativamente alta (hasta menos que un micrómetro) además el rango de medida es largo. Desventaja: difícil escanear objetos en movimiento porque solo un punto o una línea puede ser medido a la vez.</p>	
<p>Luz estructurada</p>	<p>Similar a la técnica pasiva de estero visión, excepto que una de las cámaras se reemplaza con una unidad de proyección. Los patrones proyectados son franjas o estructuras arbitrarias [21]. Los patrones se distorsionan debido a la forma del objeto [12], y el desplazamiento de las franjas permite una recuperación exacta de las coordenadas 3D de la superficie del objeto. Es capaz de obtener la R3D a gran velocidad, y el costo de computación se reduce debido a que los patrones proyectados miden múltiples puntos a la vez, reduciendo el problema de distorsión por el movimiento. La técnica proporciona medición de alta precisión [27]. Las técnicas de luz estructurada se pueden dividir en dos grandes categorías:</p>	
	<p>Método basado en la intensidad</p>	<p>Usa valores de escala de grises para la identificación de la correspondencia. Resolución no muy alta, las líneas proyectadas son mayores que un píxel del proyector y de la cámara. Requiere varios patrones para lograr alta resolución, disminuyendo la velocidad de medida, sensible a ruido en la cámara e imagen desenfocada.</p>
	<p>Métodos basados en la fase</p>	<p>La información 3D se extrae de los valores de fase codificados en el patrón de franjas. Un método para obtener la información de la fase es usando varios patrones de franjas con ciertos desplazamientos de fase (phase-shifting) [14]. Estos métodos son más robustos al ruido de la cámara y variaciones de la iluminación. Métodos para generar y proyectar patrones de franjas:</p> <p>Interferencia láser usa dos haz de láser de ondas planas para generar patrones sinusoidales por medio de interferencia. Ventaja genera patrones finos con profundidad ilimitada de campo. Desventaja alto costo de implementación, los láseres pueden afectar la calidad de medición (ruido speckle).</p>
	<p>efecto Moiré: generado por disparos de luz sobre dos rejillas superpuestas mientras son desplazadas o rotadas una pequeña cantidad. Desventaja complicado obtener información cuantitativa de altura y el requisito de controles precisos adicionales si se adopta la técnica de desplazamiento de fase.</p>	

	<p>Técnica de proyección de patrones de franjas sinusoidales Ya que esta técnica utiliza la tecnología digital, muestra grandes ventajas sobre los métodos convencionales: *Diferentes tipos de patrones pueden ser generados. *No hay error causado por los dispositivos mecánicos para desplazamiento de fase. *Gran velocidad y alta precisión de la forma 3D.</p>
--	--

Tabla 1.2: Técnicas activas y algunos ejemplos.

1.2. PROYECCIÓN DE FRANJAS

La proyección digital de franjas es una técnica de luz estructurada en la que se proyectan imágenes de franjas con intensidad de perfil tradicionalmente binario o sinusoidal, mediante el uso de dispositivos electro-ópticos, tales como vídeo-proyectores digitales [9].

Las imágenes de franjas son proyectadas sobre un objeto, adquiridas con una cámara y procesadas posteriormente para obtener la forma tridimensional del objeto [27]. El sistema de proyección de franjas está formado por un computador que genera el patrón digital de franjas (líneas rectas paralelas) que son enviadas a un proyector de vídeo (vídeo beam), y éste es el encargado de proyectarlas sobre el objeto; la forma del objeto generará una distorsión de las franjas. Con la cámara se capturan las imágenes de las franjas deformadas, cuya fase contiene implícita información de la forma tridimensional del objeto sobre el que se realiza la proyección. La Figura 1.2 hace una ilustración de esto.

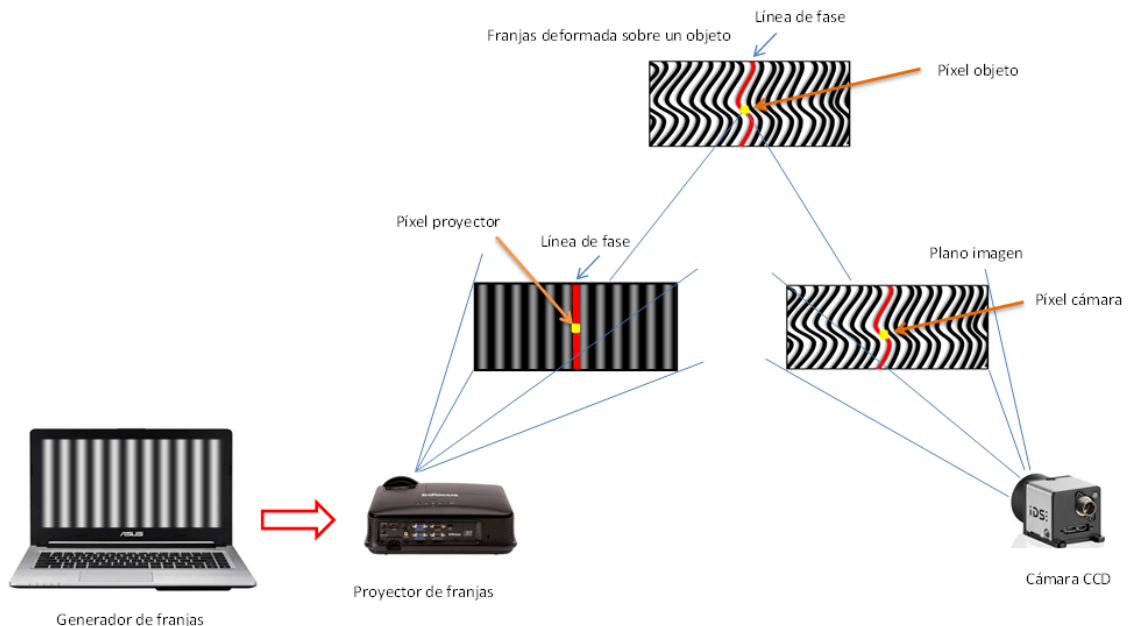


Figura 1.2: Montaje experimental de la técnica de proyección de franjas.

Existen diferentes perfiles de patrón de franjas que pueden ser usados para la proyección tales como sinusoidal [28], binario, [24] triangular [19], trapezoidal [15] o diente de sierra

[4]; en este trabajo se implementará el perfil sinusoidal debido a su facilidad para obtener la fase. Cuando se habla de un patrón de perfil sinusoidal se hace referencia que al hacer un corte transversal del patrón de intensidad de las franjas se obtiene una función seno o coseno. Ver figura 1.3.

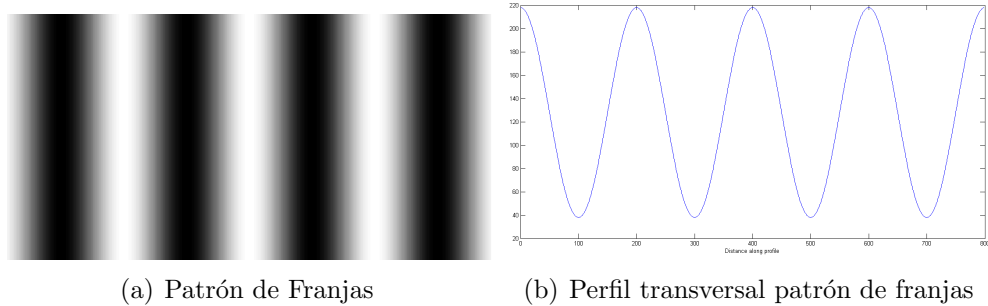


Figura 1.3: Patrón de franjas con perfil sinusoidal.

La intensidad de distribución del patrón de franjas deformadas que observa la cámara se expresa como muestra la ecuación 1.1

$$I(x, y) = A(x, y) + B(x, y) \cos \left(\frac{2\pi}{P_0} x + \Delta\varphi(x, y) \right), \quad (1.1)$$

donde $\Delta\varphi(x, y)$ es el término de fase que matemáticamente contiene información de la distribución de alturas de la superficie del objeto, $A(x, y)$ es la intensidad de fondo, $B(x, y)$ contraste de las franjas y P_0 es el periodo espacial promedio del sistema de franjas.

1.2.1. Etapas

La técnica de proyección de franjas para extraer información 3D consta de varias etapas, la primera de estas es la **proyección**, donde el patrón de franjas generado por un computador se proyecta con ayuda de un vídeo proyector (vídeo beam) sobre el objeto. Al entrar en contacto con el anterior, las franjas se deforman debido a la topografía; una cámara CCD se encarga de capturar las imágenes deformadas, siendo esta la segunda etapa denominada **adquisición**.

Con las imágenes obtenidas en el paso anterior se procede a la tercera etapa llamada **procesamiento** en la cual se trabaja con las imágenes para encontrar el valor de la *fase* que está implícito en el argumento de la función coseno; con el valor de fase obtenido se realiza el proceso de conversión fase-altura que es necesario para la implementación de la última etapa, la **reconstrucción tridimensional**.

Un proceso de calibración es realizado previamente con el fin de obtener la relación teórica o experimental que permite convertir la fase en alturas, teniendo en cuenta los parámetros del sistema.

La figura 1.4 muestra la secuencia de las etapas en la técnica de proyección de franjas, mencionadas anteriormente.

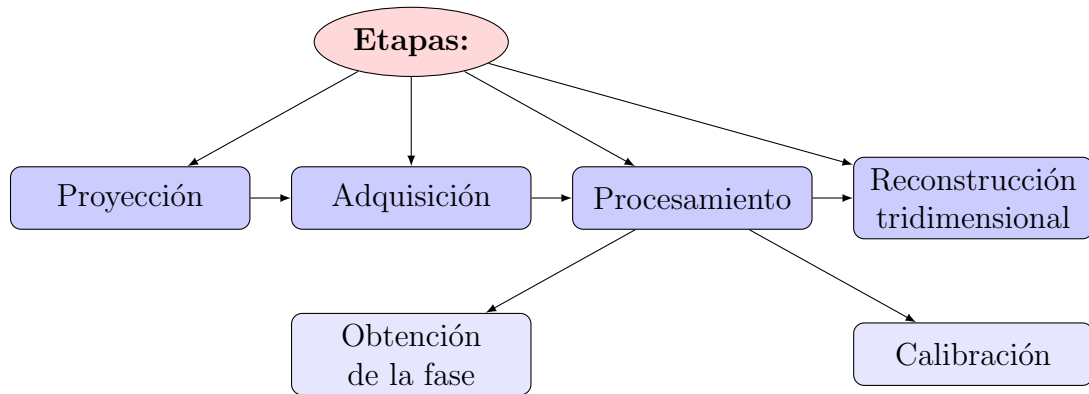


Figura 1.4: Etapas de la técnica de proyección de franjas.

La información de altura del objeto se encuentra implícita en la fase de la función coseno, por ende se debe realizar algún procedimiento para la recuperación de este valor; en la literatura se reportan diferentes métodos para la obtención de la fase tales como: correlación digital de imágenes (DIC) [2], Análisis de Fourier [23], corrimiento de fase (phase shifting) [27]. En este proyecto se utiliza el método de corrimiento de fase debido a las numerosas ventajas que presenta [11].

1.2.2. Método de corrimiento de fase

Como se dijo anteriormente el método de corrimiento de fase presenta numerosas ventajas, por esta razón es uno de los más utilizados en metrología óptica. Algunas de sus ventajas son:

1. Medición punto a punto: el cálculo de la fase en un punto es independiente de la información de los puntos vecinos.
2. Menos sensible a las variaciones de la reflectividad de la superficie, por lo tanto es utilizado para medir superficies complejas.
3. Menos sensibilidad a la luz ambiente.

Principio de funcionamiento

Con el método de corrimiento de fase se determina la fase del sistema de franjas a partir de la intensidad luminosa de n imágenes, modificando el valor de la fase en valores conocidos dentro del intervalo de 2π . En este trabajo se implementa el algoritmo con $n = 4$ imágenes, es decir que el patrón de franjas se va a desplazar un valor de $\pi/2$ consecutivamente y se adquieren 4 imágenes, Ver figura 1.5.

Debido a que las franjas se desfasaron cada $\pi/2$, un nuevo término es introducido al argumento de la función coseno modificando las ecuaciones de intensidades:

$$I_1 = A(x, y) + B(x, y) \cdot \cos[\varphi(x, y)]; \quad (1.2)$$

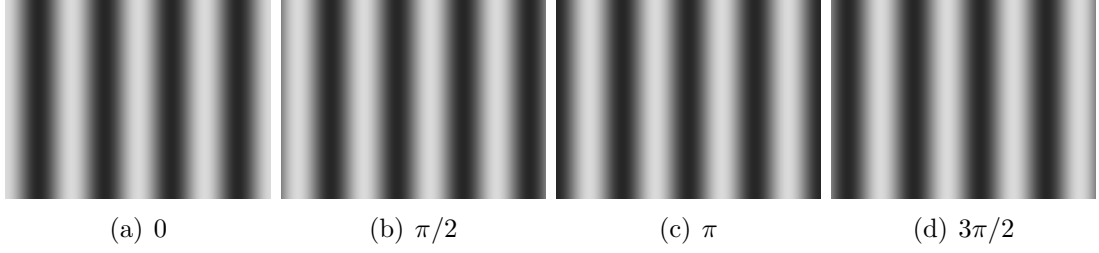


Figura 1.5: Patrón de franjas desfasados $\pi/2$ radianes.

$$I_2 = A(x, y) + B(x, y) \cdot \cos[\varphi(x, y) + \pi/2], \quad (1.3)$$

$$I_2 = A(x, y) - B(x, y) \cdot \sin[\varphi(x, y)]; \quad (1.4)$$

$$I_3 = A(x, y) + B(x, y) \cdot \cos[\varphi(x, y) + \pi], \quad (1.5)$$

$$I_3 = A(x, y) - B(x, y) \cdot \cos[\varphi(x, y)]; \quad (1.6)$$

$$I_4 = A(x, y) + B(x, y) \cdot \cos[\varphi(x, y) + 3\pi/2], \quad (1.7)$$

$$I_4 = A(x, y) + B(x, y) \cdot \sin[\varphi(x, y)]. \quad (1.8)$$

Operando con las ecuaciones 1.2, 1.4, 1.6 y 1.8, se llega a la ecuación 1.12 para calcular el valor de la fase $\varphi(x, y)$.

$$I_4 - I_2 = 2B(x, y) \cdot \sin[\varphi(x, y)], \quad (1.9)$$

$$I_1 - I_3 = 2B(x, y) \cdot \cos[\varphi(x, y)], \quad (1.10)$$

$$\frac{I_4 - I_2}{I_1 - I_3} = \frac{\sin[\varphi(x, y)]}{\cos[\varphi(x, y)]} = \tan[\varphi(x, y)], \quad (1.11)$$

$$\varphi(x, y) = \arctan\left(\frac{I_4 - I_2}{I_1 - I_3}\right) \quad (1.12)$$

El valor de $\varphi(x, y)$ que se obtiene a partir de la ecuación 1.12 está acotado de $-\pi$ a π por el comportamiento de la función arco-tangente; entonces se dice que esta fase está envuelta (wrapped) o discontinua. La figura 1.6 muestra el comportamiento de la fase obtenida con la ecuación 1.12.

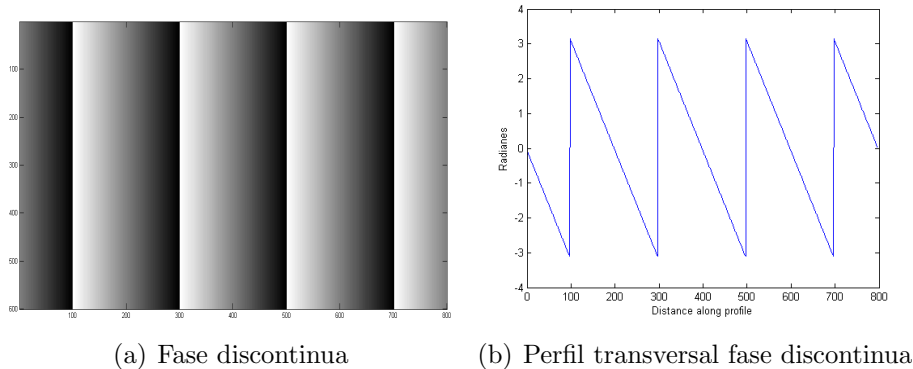


Figura 1.6: Fase envuelta y su respectivo perfil.

1.3. CORRECCIÓN DE FASE

Como se observa en la figura 1.6 la fase hallada a partir de la función arco-tangente es discontinua, por ende se debe implementar alguna técnica para corregirla, es decir, convertir esa fase discontinua a continua. Los puntos de discontinuidades corresponden a transiciones de $-\pi$ a π o viceversa y se pueden corregir adicionando o sustrayendo 2π . En los últimos años se han venido desarrollando diferentes algoritmos para el desenvolvimiento de fase, los cuales se clasifican en dos grupos: algoritmos de desenvolvimiento de fase espacial (SPU¹) y algoritmos de desenvolvimiento de fase temporal (TPU²).

1.3.1. Desenvolvimiento de fase espacial

Este método consiste en comparar los valores de fase de píxeles vecinos; Dada la fase en un píxel "A", la fase en otro píxel "B" se puede desenvolver contando el número de 2π discontinuidades a lo largo de un camino que va del píxel A al píxel B. La secuencia de los valores de fase a lo largo del camino se denotan como $\varphi(i)$ ($i = 0, 1, 2, \dots, N$), el número de discontinuidades $d(i)$ ($i = 0, 1, 2, \dots, N$) entre los dos píxeles $(i - 1)$ y (i) se pueden calcular acorde a la ecuación (1.13).

$$d(i) = \left[\frac{\varphi(i) - \varphi(i - 1)}{2\pi} \right], \quad (1.13)$$

Donde [...] denota redondeo al entero más cercano.

El verdadero cambio de fase entre dos píxeles vecinos debe estar en el rango de $-\pi$ a π , valores fuera de este rango indican que hay una discontinuidad presente dentro de estos dos píxeles y da como resultado un valor de d el cual puede ser -1 o +1. El número total

¹Spatial phase-unwrapping

²Temporal phase-unwrapping

de 2π discontinuidades a lo largo del camino se calcula como se ve en la ecuación 1.14:

$$\nu = \sum_{i=1}^N d(i), \quad (1.14)$$

luego al valor de la fase del píxel "B" se le adiciona $\pm 2\pi\nu$.

En otras palabras, este método consiste en comparar la fase entre píxeles vecinos y cuando esta diferencia es cercana a 2π se dice que hay una discontinuidad; para resolver esta discontinuidad se adiciona o sustrae $2\pi\nu$ dependiendo de la dirección en la que se lleva a cabo el desenvolvimiento, para obtener finalmente una fase continua como se observa en la figura 1.7.

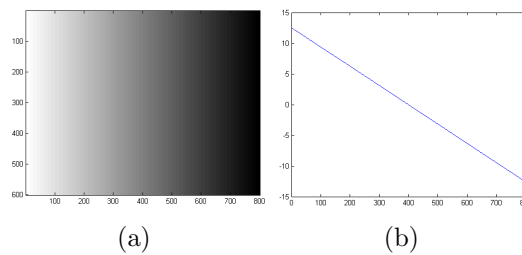
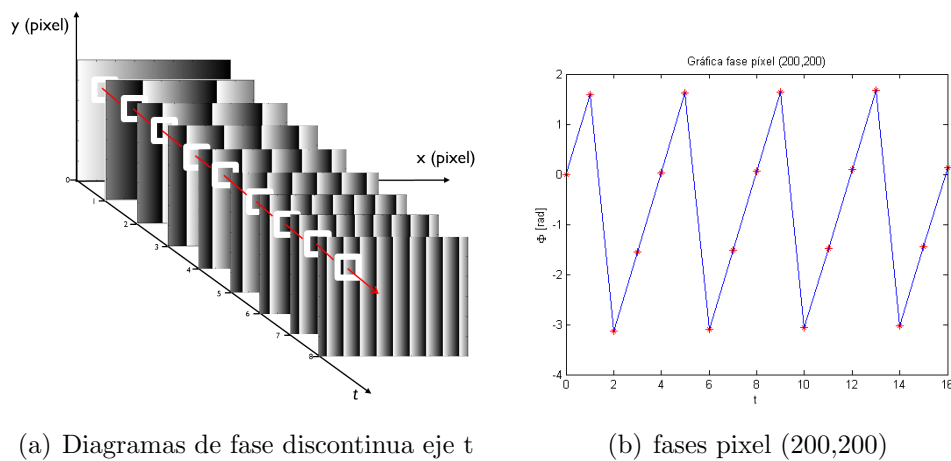


Figura 1.7: (a) Distribución fase continua, (b) Corte transversal de la fase.

1.3.2. Desenvolvimiento de fase temporal

La idea básica de este método es que la fase de cada píxel es medida como una función del tiempo. El desenvolvimiento se lleva a cabo a lo largo del eje temporal para cada píxel, independiente de otros puntos. De esta manera se evita la propagación de errores espacialmente.



(a) Diagramas de fase discontinua eje t

(b) fases píxel (200,200)

Figura 1.8: Desenvolvimiento de fase temporal.

El eje del tiempo se obtiene al proyectar franjas de diferentes pasos; para cada paso se adquieren las cuatro imágenes de intensidad con corrimiento $\pi/2$ consecutivos y se calcula la fase discontinua. La figura 1.8 muestra las fases discontinuas obtenidas cuando se proyectan sobre una superficie plana 9 patrones de franjas con pasos crecientes. Cada posición t corresponde a la distribución de fase discontinua correspondiente.

La fase de un píxel de coordenadas (m, n) en el t th ($t = 0, 1, 2, \dots, s$) mapa de fase se denota como $\varphi(m, n, t)$. Las 2π discontinuidades se calculan convirtiendo la ecuación 1.13 en tres dimensiones:

$$d(m, n, t) = \left[\frac{\Delta\varphi(m, n, t)}{2\pi} \right], \quad (1.15)$$

Donde $\Delta\varphi(m, n, t)$ es la diferencia entre el t th y $(t - 1)$ th mapa de fase ó

$$\Delta\varphi(m, n, t) = \varphi(m, n, t) - \varphi(m, n, t - 1), \quad (1.16)$$

El total de 2π discontinuidades hasta el s th mapa de fase se calcula como

$$\nu(m, n, s) = \sum_{t=1}^s d(m, n, t), \quad (1.17)$$

y $2\pi\nu(m, n, s)$ es sustraído o adicionado de $\varphi(m, n, s)$ para obtener la fase en el píxel.

Los mapas de fase discontinua se calculan de acuerdo a la ecuación 1.18

$$\varphi(t) = \arctan \left(\frac{\Delta I_{42}(t)}{\Delta I_{13}(t)} \right), \quad (1.18)$$

donde

$$\Delta I_{ij}(t) = I_i(t) - I_j(t) \quad (1.19)$$

Los valores de fase $\varphi(t)$ todos caen el rango $-\pi$ a $+\pi$ por el inverso de la función tangente, así que $\Delta\varphi(t)$ siempre cae en el rango -2π a $+2\pi$, según la ecuación 1.16. Normalmente φ esta en el rango $-\pi$ a $+\pi$, pero en este caso $\Delta\varphi$ esta en el rango -2π a $-\pi$ ó $+\pi$ a $+2\pi$ cuando una 2π discontinuidad de fase se cruza.

Los valores de $\Delta\varphi$ que se encuentran fuera del rango $-\pi$ a $+\pi$ producen valores distintos de cero de $d(t)$ (ecuación 1.15), además contribuyen al requisito de desenvolvimiento del s th mapa, (ecuación 1.17). Sin embargo $\Delta\varphi(t)$ se puede calcular directamente desde $I_1(t), \dots, I_4(t)$ y $I_1(t - 1), \dots, I_4(t - 1)$:

$$\Delta\varphi(t) = \arctan \left[\frac{\Delta I_{42}(t)\Delta I_{13}(t - 1) - \Delta I_{13}(t)\Delta I_{42}(t - 1)}{\Delta I_{13}(t)\Delta I_{13}(t - 1) + \Delta I_{42}(t)\Delta I_{42}(t - 1)} \right] \quad (1.20)$$

Lo importante es que los valores $\Delta\varphi$ calculados con la ecuación 1.20 caen automáticamente en el rango $-\pi$ a $+\pi$; es decir todos los elementos de la matriz de desenvolvimiento

$d(m, n, t)$ son cero, por la ecuación 1.15. Entonces el s th mapa de fase puede ser calculado por la suma de las diferencias de fase sin un desenvolvimiento adicional [16]:

$$\varphi(m, n, s) = \sum_{t=1}^s \Delta\varphi(m, n, t). \quad (1.21)$$

Las siguientes figuras muestran las ventajas del desenvolvimiento temporal de fase. La figura 1.9 muestra algunos diagramas de fase discontinua con diferentes pasos de franjas proyectadas.

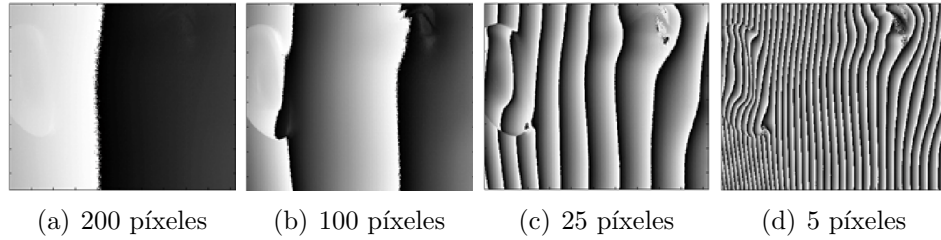


Figura 1.9: Fase discontinua para cada paso de proyección.

La idea central del algoritmo consiste en obtener la fase continua del último sistema de franjas usando la información de fase de las etapas anteriores, sin recurrir a análisis espaciales de fase discontinua. La ecuación 1.21 indica que en la medida que se proyecta un nuevo sistema de franjas se calcula su fase continua sumando las diferencias de las fases anteriores. La Figura 1.10 muestra como evoluciona la fase continua hasta obtener la fase del ultimo sistema de franjas proyectados.

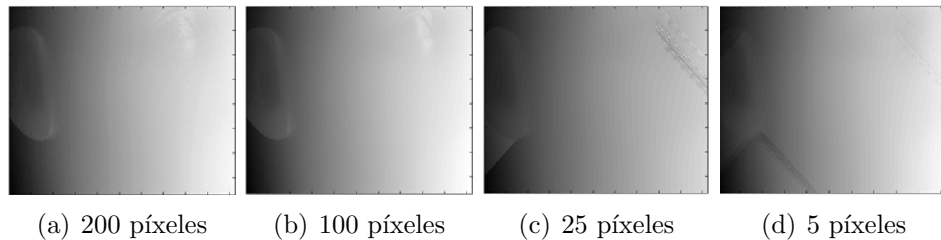


Figura 1.10: Fase continua para cada paso de proyección.

1.4. PROCEDIMIENTO EXPERIMENTAL DE R3D

Como se mostró en la figura 1.4 para llegar a la R3D³ de un objeto, primero debe obtenerse el valor de la fase, una vez obtenida esta es necesario previamente calibrar el sistema para así poder determinar la relación fase-altura y convertir la fase calculada en distribución de alturas.

³Reconstrucción tridimensional

La calibración consta de dos partes, calibración de la cámara CCD y calibración del sistema; para calibrar la cámara CCD se deben encontrar los parámetros intrínsecos y extrínsecos de esta y posibles distorsiones geométricas de los sistemas formadores de imagen, donde se utilizan expresiones que se deducen a partir de la relación de las características de la cámara y su geometría. Esta calibración es requerida para convertir píxeles de la cámara en mm sobre el plano del objeto [6].

El proceso de calibración experimental del sistema se realiza proyectando patrones de franjas sobre un plano de referencia, el cual es desplazado a intervalos conocidos ΔZ en el rango de volumen Z de calibración. En cada posición Z se realiza la respectiva captura de las imágenes para así lograr obtener la fase correspondiente. De esta manera para cada píxel en la imagen se tienen los valores de $\Delta\Phi$ y Z en el rango de calibración.

Debido a que el sistema de proyección no es telecéntrico la relación fase-altura no es lineal, por esta razón se realiza una corrección adicionando un término cuadrático, generando así la ecuación de calibración de la siguiente manera:

$$\Delta\varphi = \varphi_z - \varphi_o = a(x, y)Z^2 + b(x, y)Z + c(x, y), \quad (1.22)$$

donde φ_z es la fase para cada posición Z , $a(x, y), b(x, y), c(x, y)$ son los coeficientes de calibración calculados para cada píxel sobre la imagen de fase. φ_o define la fase de referencia para la posición Z tomando como origen $Z_0 = 0$ mm. Estos parámetros son almacenados y conforman los parámetros de calibración del sistema.

Para realizar el proceso de R3D lo principal es garantizar que el objeto a reconstruir se encuentra en el rango de calibración, luego se proyectan las 4 imágenes de franjas desplazadas, se calcula φ_z (ec.1.12) y se calcula $\Delta\varphi$. Despejando de la ecuación 1.22 se obtiene Z . De esta manera para cada píxel de la imagen con información de fase se calcula la altura en mm. Obteniéndose así una imagen en coordenadas (píxeles, píxeles, mm) siendo los mm en Z . Con los parámetros de calibración de la cámara CCD se convierten los píxeles en mm sobre el plano del objeto y se obtiene la imagen final de R3D en mm [3].

1.5. PLANTEAMIENTO DEL PROBLEMA

Desde el interior del grupo de óptica y tratamiento de señales (GOTS) se han desarrollado diferentes trabajos de grado e investigaciones en el área de la metrología óptica empleando como método la reconstrucción tridimensional de objetos. Con base a experiencias de socialización en eventos especializados (nacionales e internacionales) de estos trabajos, se considera que se tiene un amplio dominio en el tema, específicamente respecto a la técnica de proyección de franjas.

Dentro de las principales aplicaciones para las que se han desarrollado los trabajos del grupo GOTS en proyección de franjas se destacan: estudios de alteraciones cutáneas como

celulitis [3], medida de la forma del torso humano [11], análisis de arrugas faciales [20], e inspección de fallas superficiales en tuberías no metálicas para transporte de crudo [1], entre otros. Todos estos trabajos han despertado gran interés en el sector industrial de la región motivando así la idea de llevarlos hacia dispositivos comerciales.

Se ha presentado un inconveniente común en la implementación de la técnica de proyección de franjas que impide el desarrollo de un proceso automático para el desenvolvimiento de la fase, siendo este parte fundamental para lograr consolidar un dispositivo. El problema se presenta cuando el objeto a reconstruir presenta discontinuidades en su superficie o variaciones abruptas en la altura, generando sombras o corrimientos en las franjas que alteran la continuidad de estas, ver figura 1.11.

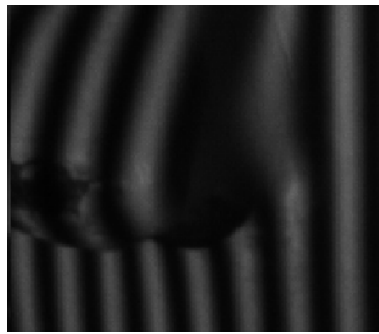


Figura 1.11: Discontinuidad de las franjas sobre un objeto.

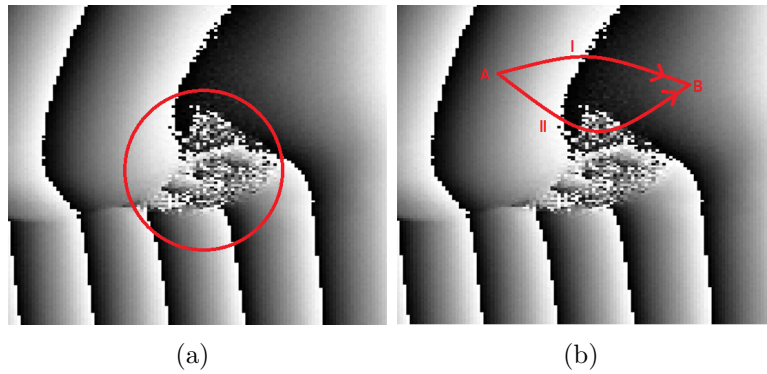


Figura 1.12: Pérdida de información en el cálculo de la fase.

Para este caso, se presenta pérdida de información en el cálculo de la fase, debido a que no es posible conocer la continuidad en las franjas. La Figura 1.12 ilustra esta situación. La Figura 1.12(a) muestra la fase discontinua en una región que posee sombras: se observa alta influencia de ruido en la fase. La Figura 1.12(b) ilustra el inconveniente que se presenta: al corregir la fase del punto B a partir del punto A, siguiendo el camino I se detecta una transición de fase cercana a -2π y se corrige sin inconvenientes; al usarse el camino II se encuentra que al realizar el cálculo de la diferencia de fase entre vecinos, se

detectan más de una transición cercana a 2π , corrigiéndose la fase incorrectamente, ver figura 1.13. Así un algoritmo de corrección de fase espacial propaga errores a zonas que poseen información correcta, causada por regiones con alto contenido de ruido.



Figura 1.13: Propagación de error en la fase continúa.

Bibliográficamente se han planteado soluciones a este inconveniente, donde se destaca el uso de mascararas que validan regiones útiles y evitan que el algoritmo de corrección de fase pase por regiones invalidadas. Desafortunadamente estas propuestas impiden la obtención de un algoritmo automático, ya que la presencia e influencia del ruido depende del objeto y además en la generación de mascararas es necesaria la intervención del usuario.

En la búsqueda de soluciones para este problema, surge la hipótesis que al implementar un algoritmo de desenvolvimiento de fase temporal se elimina este error, ya que el desenvolvimiento no se hace comparando los píxeles vecinos, sino comparando la fase anterior a lo largo del eje del tiempo, que fue hallada con un paso de proyección menor para reducir la perdida de información.

Otro de los problemas que se han encontrado al implementar el algoritmo de desenvolvimiento espacial es que la fase continua que se obtiene difiere de la fase absoluta de proyección en un valor entero de 2π , ya que depende del punto de partida del algoritmo de desenvolvimiento. Es decir, al hacer un corte transversal de la fase esta puede estar desplazada hacia arriba o hacia abajo en valores enteros de 2π (figura 1.14). La fase relativa genera una alteración de la forma 3D del objeto reconstruido al emplearse una relación fase-altura no lineal (ec 1.22). El concepto de fase absoluta se muestra en la figura 1.2. En el sistema coordenado del proyector se ubica un sistema de franjas que posee una fase lineal, ubicando la fase $\varphi = 0$ en el centro del sistema coordenado (línea roja). Al proyectar sobre el objeto y obtener la imagen en la CCD, la franja central o franja de fase cero no se diferencia de las otras. En el proceso de corrección de fase, el algoritmo calcula una fase continua cuya franja central no posee fase cero, debido a la dificultad en identificar dicha franja.

De esta manera el algoritmo de desenvolvimiento de fase requiere un punto de par-

tida para hacer la exploración espacial, detectar discontinuidades y corregir adicionando enteros apropiados de 2π . Si el punto de partida no se escoge en la franja central, la fase obtenida difiere en enteros de 2π de la fase absoluta de proyección.

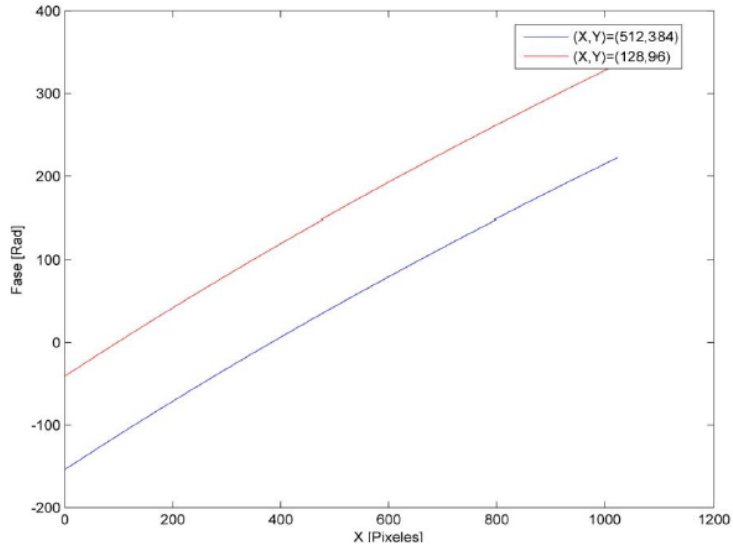


Figura 1.14: Perfil fase continua para puntos de partida diferente. Fuente: [3]

Para evitar esto se proyecta una línea blanca ubicada en el centro del sistema coordenado de proyección (figura 1.15). La imagen adquirida por la cámara permite definir el punto de partida para el algoritmo de corrección de fase.

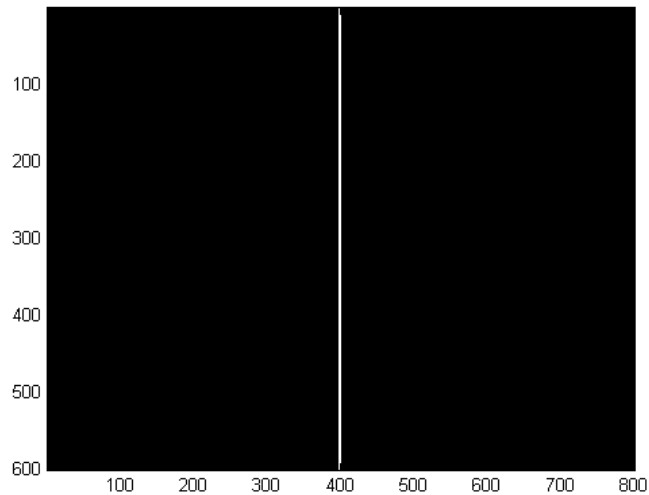


Figura 1.15: Línea central

La ventaja que presenta el algoritmo de desenvolvimiento temporal respecto al espacial, es que al incrementar el número de franjas entre imágenes se realiza de tal manera

que la fase cero siempre se encuentra en el centro del campo de proyección, es decir se obtendrá siempre la fase absoluta sin necesidad de hacer correcciones de ningún tipo.

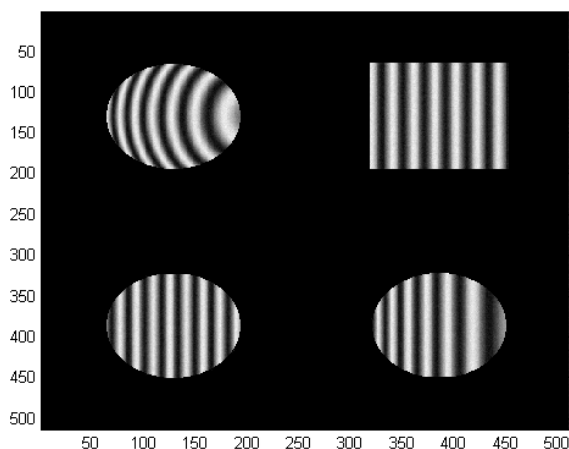


Figura 1.16: Problema de objetos aislados

La figura 1.16 muestra otro inconveniente presente en el algoritmo espacial, la reconstrucción de objetos aislados. Cuando el objeto está conformado por diversas regiones aisladas con información topográfica, al proyectar las franjas se observa en la imagen adquirida zonas sin información. En el proceso de corrección espacial de fase cada zona se debe corregir de manera independiente perdiéndose la posición 3D relativa de una zona con respecto a la otra.

2. *ALGORITMOS DE DESENVOLVIMIENTO DE FASE TEMPORAL TPU*

La corrección de la fase usando las técnicas de desenvolvimiento temporal es un tema estudiado en los últimos años debido a los problemas que se encontraron al implementar el algoritmo espacial para corregir la fase sobre objetos que presentan topografía compleja¹. El algoritmo espacial funciona muy bien cuando se aplica sobre superficies sin cambios drásticos y bajo contenido de ruido, pero cuando se aplica para objetos complejos se presenta propagación de errores debido a pérdidas de información que pueden ser causadas por sombras o discontinuidad en las franjas.

Con el fin de corregir estos problemas se desarrolló el algoritmo de desenvolvimiento temporal, el cual ha sufrido diversas modificaciones sobre el algoritmo base planteado por Huntley and Saldner (1993). Estas modificaciones se han generado con el propósito de obtener mejores resultados en la fase, reduciendo el tiempo de proyección-adquisición y a su vez evitando la intervención del usuario.

Realizando una revisión bibliográfica se encuentra que algunos de los algoritmos de desenvolvimiento de fase temporal ampliamente utilizados y referenciados en múltiples trabajos de investigación son: Algoritmos de secuencia lineal (LTPU) [16], secuencia exponencial (ETPU) [18], secuencia exponencial inversa (RETPU) [17], y generalizado (GT-PU) [26].

En este capítulo se plantea el fundamento teórico de cada uno de los algoritmos mencionados anteriormente y a su vez se incluirán algunas simulaciones de cada uno de estos sobre objetos planos y objetos discontinuos.

Es importante resaltar que la fase se obtiene a partir de la función arcotangente, y por esta razón es discontinua; para desenvolver la fase, es decir volverla continua, la idea básica del algoritmo temporal es que la fase de cada píxel se mide como una función del tiempo, donde los píxeles se desenvuelven independiente de la información de los vecinos.

¹Que presentan cambio abrupto en la forma o protuberancias que generan sombras.

2.1. DESENVOLVIMIENTO DE FASE TEMPORAL LINEAL (LTPU)

Este algoritmo consiste en proyectar patrones de franjas con un incremento lineal del número de las franjas en el campo de proyección, es decir, se proyecta una franja, dos franjas, tres franjas en todo el espacio y así sucesivamente hasta llegar al número máximo de franjas deseadas. Para cada sistema de franjas se calcula el mapa de fase discontinuo usando el algoritmo a 4 imágenes. Cada fase discontinua presenta las transiciones de fase $\pm\pi$, excepto la primera que se obtiene con una sola franja en todo el campo de proyección.

El desenvolvimiento o corrección de fase en este algoritmo se realiza usando la siguiente ecuación recursiva:

$$\varphi_c(t) = \varphi_d(t) - 2\pi \left[\frac{\varphi_d(t) - \varphi_c(t-1)}{2\pi} \right] \quad (2.1)$$

$$\varphi_c(t) = U \{ \varphi_d(t), \varphi_c(t-1) \} \quad (2.2)$$

donde $U \{ \varphi_1, \varphi_2 \}$ representa el operador desenvolvimiento el cual sustrae un múltiplo entero de 2π de φ_1 tal que $\varphi_2 - \varphi_1$ cae dentro el rango $-\pi$ a $+\pi$, es decir,

$$U \{ \varphi_1, \varphi_2 \} = \varphi_1 - 2\pi \left[\frac{\varphi_1 - \varphi_2}{2\pi} \right] \quad (2.3)$$

[...] representa redondeo al entero más cercano. Los valores enteros indican las veces que 2π debe ser adicionado o sustraído de la fase que se esta desarrollando; esto valores se conocen con el nombre de 2π discontinuidades y comúnmente se denota por la letra "d".

En la ec. 2.1, $t = 2, 3, \dots, s$, siendo s el número de franjas finales. Con el fin de evaluar el funcionamiento de la ecuación 2.1, se mostrarán los pasos intermedios para valores progresivos de t simulando una superficie plana. La figura 2.1 muestra la fase discontinua obtenida al proyectar una franja en el campo de observación, esta no debe tener discontinuidades, lo cual implica que corresponde a $\varphi_c(1)$. La figura 2.2 muestra la fase discontinua para $t = 2$ ($\varphi_d(2)$). Con el fin de calcular las 2π discontinuidades se emplea la ecuación 2.4, la figura 2.3 muestra la diferencia $\varphi_d(2) - \varphi_c(1)$ y la figura 2.4 muestra las 2π discontinuidades al usar la ec. 2.4.

$$d = \left[\frac{\varphi_d(t=2) - \varphi_c(t=1)}{2\pi} \right], \quad (2.4)$$

Comparando las gráficas de la figura 2.2b y la figura 2.4b se ve que para que la fase sea continua, es decir, no presente saltos se le debe adicionar o sustraer 2π en los píxeles

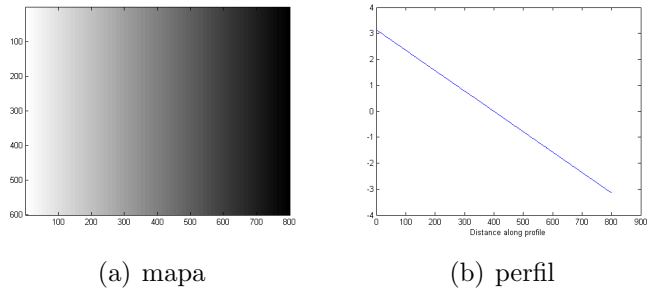


Figura 2.1: Mapa de fase discontinua $t = 1$.

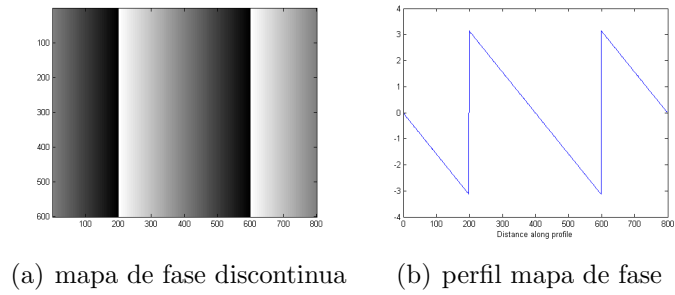


Figura 2.2: Mapa de fase discontinua $t = 2$.

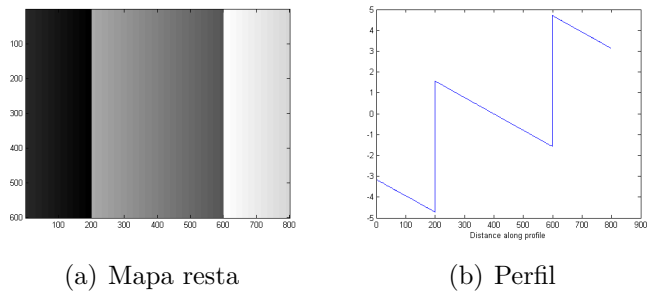


Figura 2.3: Resta de los mapas de fase ($t = 2$) - ($t = 1$).

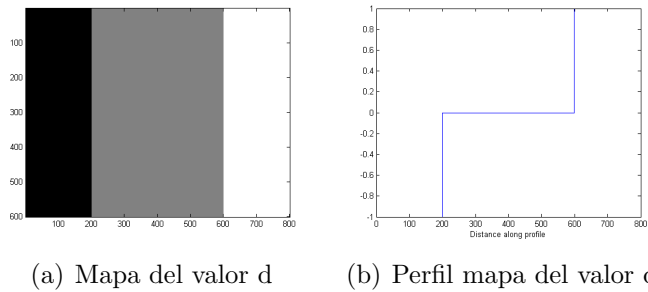
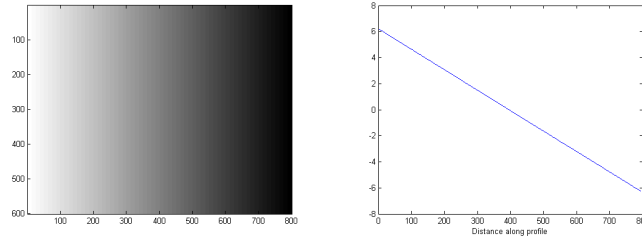


Figura 2.4: valores de las 2π discontinuidades

0-200 y 600-800, el proceso descrito anteriormente lo realiza la ecuación 2.5. Con lo que finalmente se obtiene la fase continua del mapa $t = 2$ figura 2.5.

$$\varphi_c(2) = U \{ \varphi_d(2), \varphi_c(1) \} = \varphi_d(2) - 2\pi d \quad (2.5)$$



(a) Mapa del valor d (b) Perfil mapa del valor d

Figura 2.5: Fase continua del mapa $t=2$

La fase continua del mapa $t = 2$ sirve para determinar los valores de "d" para la fase discontinua con 3 franjas proyectadas. El proceso se repite hasta llegar al último sistema de franjas proyectadas $t = s$ franjas en el campo de proyección. Las ecuaciones generalizadas para el mapa $t = s$ se muestran a continuación.

$$\varphi_c(t = s) = \varphi_d(t = s) - 2\pi d(\Delta\varphi) \quad (2.6)$$

$$\Delta\varphi = \varphi_d(t = s) - \varphi_c(t = s - 1) \quad (2.7)$$

Para comprender mejor como funciona el algoritmo lineal la figura 2.6 muestra un esquema del proceso que se lleva a cabo.

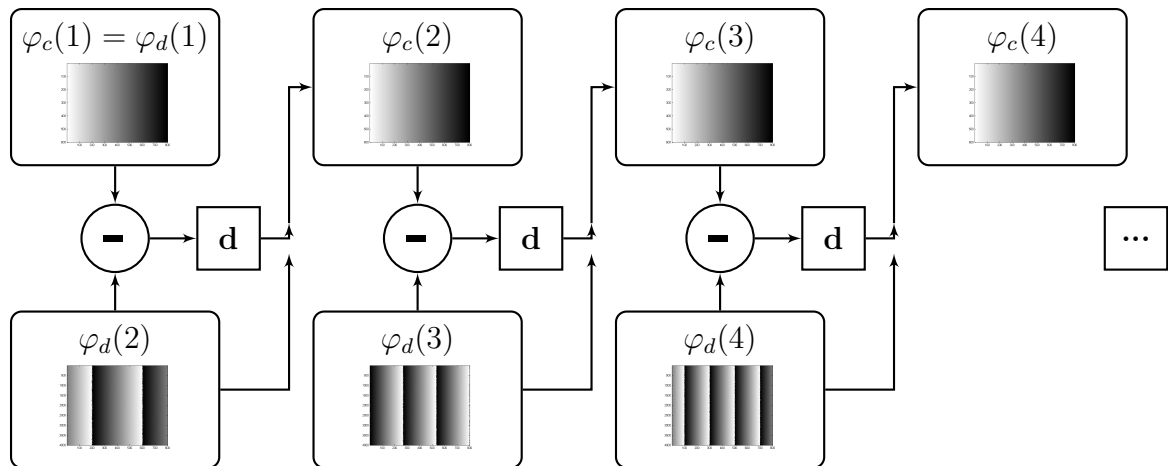


Figura 2.6: Algoritmo lineal

Como se mostró en la sección 1.2 el patrón de franjas se genera a partir de una función coseno de la forma: $\cos(2\pi x/P_t)$, (ecuación 1.1) donde P_t es el periodo espacial proyectado en el instante t . Sustituyendo P_t por $1/f_t$ la función coseno se transforma en $\cos(2\pi f_t x)$ donde $f_t = t/\Delta$, por ende la fase de esta función sera:

$$\varphi(x) = \frac{2\pi t}{\Delta}x, \quad (2.8)$$

Siendo Δ el campo de proyección en píxeles sobre el vídeo proyector. El eje x corresponde al sistema coordenado del proyector en valores enteros; $x = -\Delta/2, \dots, 0, \dots, \Delta/2$. De esta manera $\varphi(0) = 0$ se ubica en el centro de proyección. t corresponde al número de franjas que se desean proyectar si $t = 1$, φ varía entre $-\pi$ a π pasando por cero en el centro de proyección.

La explicación teórica de por que funciona el algoritmo lineal se mostrará a partir de la gráfica 2.7; en esta gráfica se encuentran la fase discontinua del mapa $t = 1$ (azul), la fase discontinua del mapa $t = 2$ (roja) y la diferencia entre la fase $t = 2$ y $t = 1$ (rosa), las líneas discontinuas de color rojo representan el comportamiento que debería tener la fase en el mapa $t = 2$.

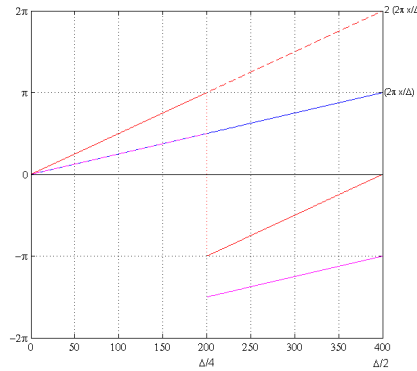


Figura 2.7: Gráfica del perfil de las fases y su diferencia

A partir de la ec. 2.8 la ecuación de la recta para $t = 1$ es:

$$\varphi(x) = \frac{2\pi}{\Delta}x. \quad (2.9)$$

Para el caso de $t = 2$ la ecuación de la recta debería ser:

$$\varphi(x) = \frac{4\pi}{\Delta}x, \quad (2.10)$$

pero debido a que las fases se calculan haciendo uso de la función arcotangente, la ecuación de la recta para $t = 2$ se convierte en una función a trozos en el rango de $-\pi$ a π , es decir, la ecuación 2.10 se transforma en:

$$\varphi(x) = \begin{cases} \frac{4\pi x}{\Delta} & \text{si } x \leq \Delta/4 \\ \frac{4\pi x}{\Delta} - 2\pi & \text{si } x > \Delta/4 \end{cases} \quad (2.11)$$

Para corregir la fase discontinua de $t = 2$ es necesario encontrar los valores "d" que indican las veces que 2π debe ser adicionado para obtener la fase continua; se debe hacer la resta entre el mapa $t = 2$ (ec 2.11) y $t = 1$ (ec 2.9),

$$\varphi(x)_{t=2} - \varphi(x)_{t=1} = \begin{cases} \frac{2\pi x}{\Delta} & \text{si } x \leq \Delta/4 \\ \frac{2\pi x}{\Delta} - 2\pi & \text{si } x > \Delta/4 \end{cases} \quad (2.12)$$

con la diferencia entre los mapas el siguiente paso es dividir en 2π y calcular la parte entera para así obtener finalmente el valor "d"

$$d(\Delta\varphi) = \begin{cases} 0 & \text{si } x \leq \Delta/4 \\ -1 & \text{si } x > \Delta/4 \end{cases} \quad (2.13)$$

los valores de la ecuación 2.13 indican que para $x \leq \Delta/4$ no es necesario adicionar 2π ya que esta se encuentra en la posición correcta (figura 2.7 rojo) pero en $x > \Delta/4$ se debe adicionar -2π el cual desplaza el segmento de la función hacia arriba y convierte finalmente la fase del mapa $t = 2$ en continua.

Con la fase continua de $t = 2$ es posible calcular la fase continua del mapa siguiente, $t = 3$. En la gráfica 2.8 se muestra la fase continua de $t = 2$ (verde), la fase discontinua en $t = 3$ (azul) y la resta entre el mapa discontinuo $t = 3$ y el mapa continuo $t = 2$ (rosa).

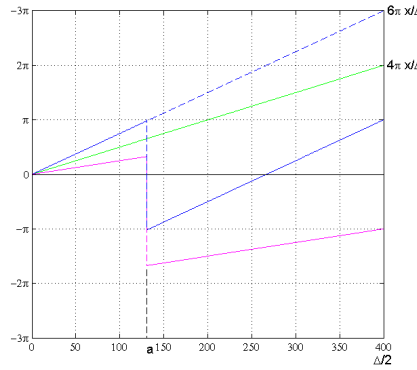


Figura 2.8: Gráfica del perfil de las fases y su diferencia

La ecuación de la fase continua de $t = 2$ ahora si se comporta como se mostró en la ec. 2.10, la ecuación de la fase discontinua de $t = 3$ nuevamente es una función a trazos de la forma:

$$\varphi(x) = \begin{cases} \frac{6\pi x}{\Delta} & \text{si } 0 < x \leq a \\ \frac{6\pi x}{\Delta} - 2\pi & \text{si } a < x < \Delta/2 \end{cases} \quad (2.14)$$

Al hacer la resta entre la fase discontinua de $t = 3$ (ec. 2.14) y la fase continua de $t = 2$ (ec. 2.10) se obtiene:

$$\varphi(x)_{d,t=3} - \varphi(x)_{c,t=2} = \begin{cases} \frac{2\pi x}{\Delta} & \text{si } 0 < x \leq a \\ \frac{2\pi x}{\Delta} - 2\pi & \text{si } a < x < \Delta/2 \end{cases} \quad (2.15)$$

finalmente los valores de "d" se obtienen dividiendo en 2π y sacando la parte entera de la ecuación 2.15.

$$d(\Delta\varphi) = \begin{cases} 0 & \text{si } 0 < x \leq a \\ -1 & \text{si } a < x < \Delta/2 \end{cases} \quad (2.16)$$

estos valores "d" corrigen la fase discontinua del mapa $t = 3$. El proceso se repite hasta llegar a la fase continua del mapa $t = s$.

Lo anterior demuestra que el algoritmo funciona debido a que se conoce la fase continua del mapa inmediatamente anterior haciendo que $\left[\frac{2\pi x}{\Delta}\right] = 0$. Es decir, por el hecho de ir incrementando una sola franja en todo el campo el término $\left[\frac{2\pi x}{\Delta}\right]$ es igual a 0 y el valor de "d" lo define la transición entera de 2π incluida en las ecs. 2.13 y 2.16, que corresponde al valor entero exacto para corregir la fase discontinua. Es importante destacar que la fase cero se ubica en $x = 0$, es decir en el centro del sistema de proyección. A partir de esta posición las pendientes van incrementando, asegurando un aumento lineal en el número de franjas.

2.1.1. Simulación con objeto plano

La eficiencia del algoritmo se mostrará realizando una simulación en la cual se proyectan franjas sobre un objeto plano adicionándoles ruido gaussiano; en el proceso experimental este ruido podría ser debido a la cámara o al vídeo-proyector y es adicionado a la imagen final del sistema de franjas.

Para observar como se comporta el algoritmo en presencia de ruido, primero se introducirá ruido gaussiano de media cero y desviación estándar σ baja, se obtiene la respectiva fase y luego se hace la prueba en presencia de ruido con desviación estándar alta; así se podrá mostrar que tan inmune al ruido es el algoritmo. El ruido introducido está dado en niveles de gris (NG) adicionado a los patrones de franjas.

El primer paso es generar el patrón de franjas desfasadas $\pi/2$ con ruido introducido, modificando las ecuaciones 1.2, 1.3, 1.5 y 1.7 en:

$$I_1 = A(x, y) + B(x, y) \cos[\varphi(x, y)] + \varepsilon_1; \quad (2.17)$$

$$I_2 = A(x, y) + B(x, y) \cos[\varphi(x, y) + \pi/2] + \varepsilon_2; \quad (2.18)$$

$$I_3 = A(x, y) + B(x, y) \cos[\varphi(x, y) + \pi] + \varepsilon_3; \quad (2.19)$$

$$I_4 = A(x, y) + B(x, y) \cos[\varphi(x, y) + 3\pi/2] + \varepsilon_4, \quad (2.20)$$

Ruido bajo

En esta sección se mostrarán resultados introduciendo un error gaussiano de $\sigma = 5$ NG y con un patrón máximo de franjas proyectadas $s = 32$. La figura 2.9 muestra los primeros cuatro patrones de franjas sin desfase, los cuales llevan introducido el error ε_i . La figura 2.10 muestra los primeros cuatro mapas de fases discontinuos obtenidos al desfasar $\pi/2$ los respectivos sistemas de franjas.

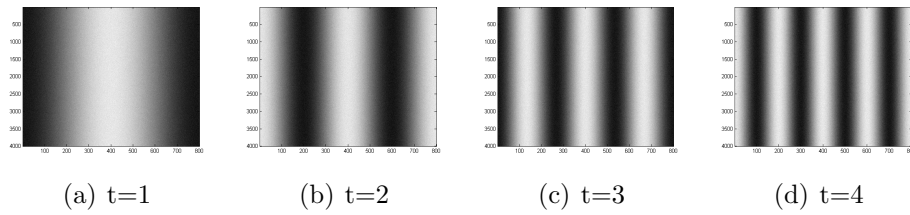


Figura 2.9: Primeros cuatro patrones de franjas.

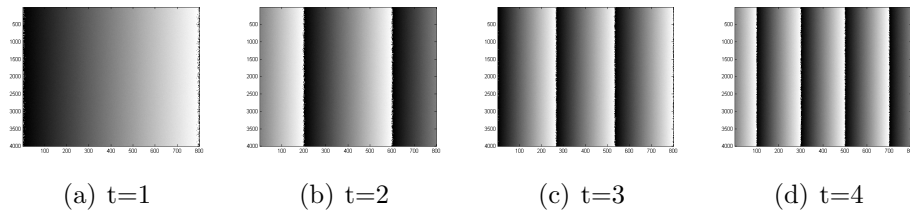


Figura 2.10: Fases discontinuas

Una vez se obtienen los mapas de fase discontinuos, se procede a calcular la fase continua implementando las ecuaciones 2.4, 2.6 y 2.7. El primer paso es hacer la diferencia entre el mapa $t = 2$ y $t = 1$, figura 2.11.

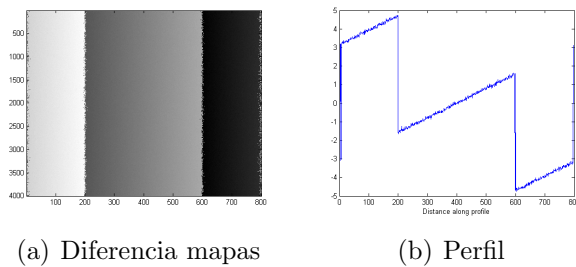


Figura 2.11: Diferencia t=2 y t=1.

El siguiente paso es calcular las 2π discontinuidades, como se mostró en la ecuación 2.4, las cuales gráficamente se ven en la figura 2.12.

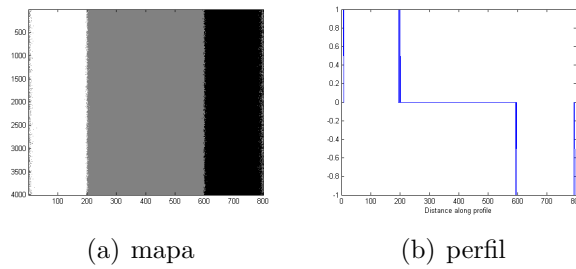


Figura 2.12: Valores "d" para $t=2$

Finalmente los valores "d" se multiplican por 2π y se sustraen o adicionan a la fase discontinua que se desea desenvolver, en este caso $t = 2$; de esta forma se obtiene la fase continua del mapa, con la cual después se calcularán los "d" del mapa $t = 3$. Este proceso se repite hasta llegar a la fase continua de $s = 32$ la cual se muestra en la figura 2.13.

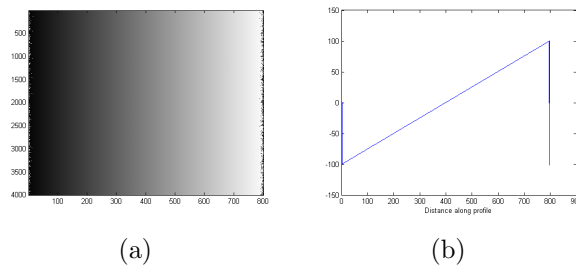


Figura 2.13: Fase continua para $s=32$, algoritmo lineal.

La figura 2.13b muestra un corte de intensidad de la fase continua para $s=32$. Se observa que en vez de obtener una línea recta, hay puntos ubicados en los extremos que se alejan de la línea recta, obteniéndose puntos erróneos causados por la influencia del ruido. Estos puntos erróneos son aislados e inevitablemente el dato es afectado debido a la alta influencia del ruido. No obstante si se hubiese empleado un algoritmo de corrección de fase espacial, este error se hubiera propagado a píxeles vecinos con información correcta.

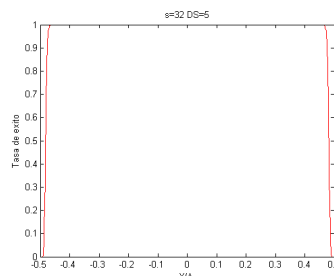


Figura 2.14: Tasa de éxito para ruido bajo lineal.

En la figura 2.14 se observa la tasa de éxito del desenvolvimiento con el ruido introducido, se aprecia que los extremos presentan éxito cero, lo que coincide con lo obtenido en la fase continua anterior. En las demás regiones el algoritmo tiene un éxito de 1 indicando que para baja presencia de ruido el algoritmo lineal podría ser muy estable.

Ruido alto

Anteriormente se mostró que el algoritmo funciona muy bien cuando se tiene ruido bajo, ahora se pasará a analizar que sucede si por el contrario hay presencia de ruido alto. Para esto se introducirá un ruido gaussiano con $\sigma = 30$ NG y nuevamente se calculará la fase para el mapa $s=32$.

A partir de las ecs 2.17, 2.18, 2.19 y 2.20 se obtienen los patrones de franjas, la figura 2.15 muestra los cuatro primeros patrones de franjas en presencia de un ruido alto, se puede apreciar que las franjas no están muy bien definidas comparadas con el caso anterior.

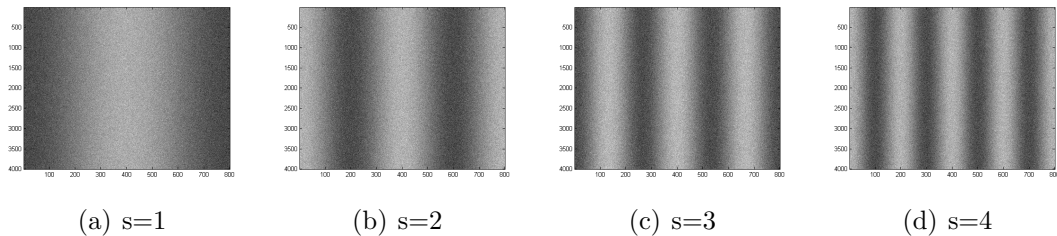


Figura 2.15: Primeros cuatro patrones de franjas.

Implementando el algoritmo a 4 imágenes se calculan las respectivas fases discontinuas, figura 2.16, en la cuales es más visible la presencia de ruido alto.

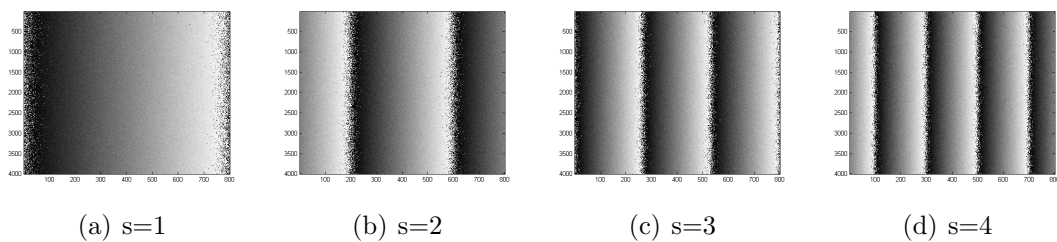


Figura 2.16: Fases discontinuas

El siguiente paso es proceder a calcular los valores "d" para el mapa $t = 2$, debido a que el primer mapa que se obtiene siempre es continuo. Para obtener estos valores "d" es necesario hacer la diferencia entre el mapa $t = 2$ y $t = 1$, figura 2.17

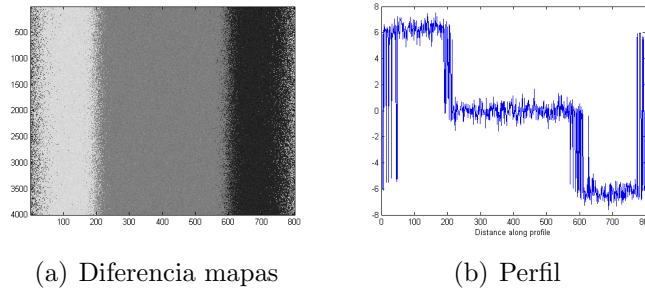


Figura 2.17: Diferencia $t=2$ y $t=1$.

dividiendo esta diferencia entre 2π y calculando su parte entera se obtienen los valores d , los cuales se ilustran en la gráfica 2.18, se observa que estos no están bien definidos contrario al caso de ruido bajo, es decir al introducir ruido alto en las franjas cada paso para obtener la fase continua se ve afectado por el ruido.

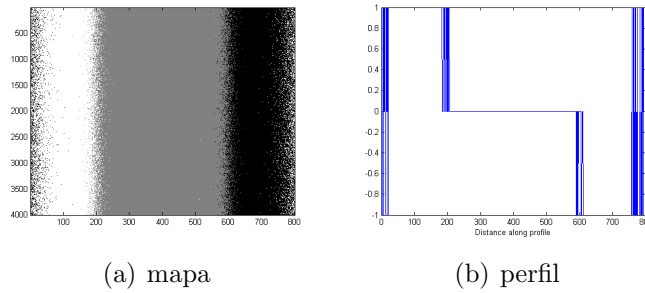


Figura 2.18: Valores "d" para $t=2$

Una vez obtenidos los "d" es posible obtener la fase continua del mapa $t = 2$, multiplicando estos por 2π y sustrayendolos de la fase discontinua de interés. Este proceso se repite hasta llegar a la fase de $t = s$; la figura 2.19 muestra la fase continua de $s = 32$, se observa nuevamente que el error se genera a los extremos.

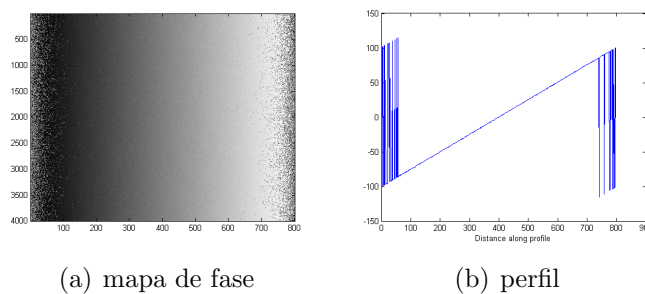


Figura 2.19: Fase continua para $s=32$.

La figura 2.20 muestra la tasa de éxito del algoritmo, se observa que el dominio donde la probabilidad de éxito es 1 se redujo y aumentaron las regiones con menos probabilidad de

éxito, nuevamente estos se mantienen en los extremos. Además demuestra que el algoritmo lineal no sería una buena alternativa si el sistema presenta ruido alto.

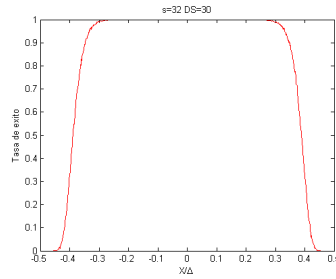


Figura 2.20: Tasa de éxito para ruido alto lineal.

2.1.2. Simulación con objeto discontinuo

En esta sección se implementará el algoritmo a partir de la simulación de objetos discontinuos con $s=32$, y un error introducido de $\sigma = 5$ NG; lo primero es obtener las franjas deformadas debido a la topografía del objeto, figura 2.21 las cuales son desfasadas $\pi/2$ y con el algoritmo a 4 imágenes se obtienen las fases discontinuas, figura 2.22.

Las imágenes de franjas son obtenidas simulando un sistema de proyección de franjas telecéntrico [10], con un ángulo de proyección $\theta = 25^\circ$ y un campo de observación de $513mm \times 513mm$. La cámara se ubica perpendicular al plano de referencia.

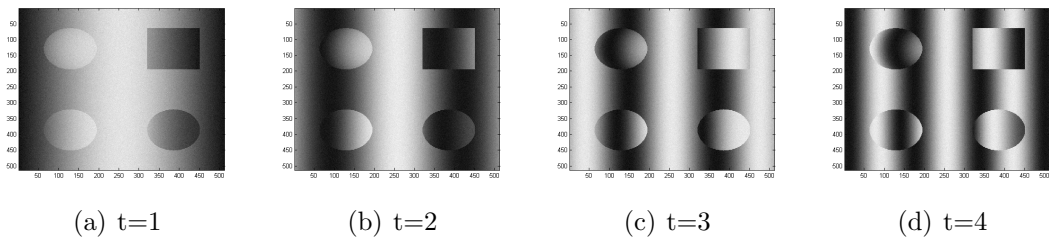


Figura 2.21: Primeras 4 franjas proyectadas

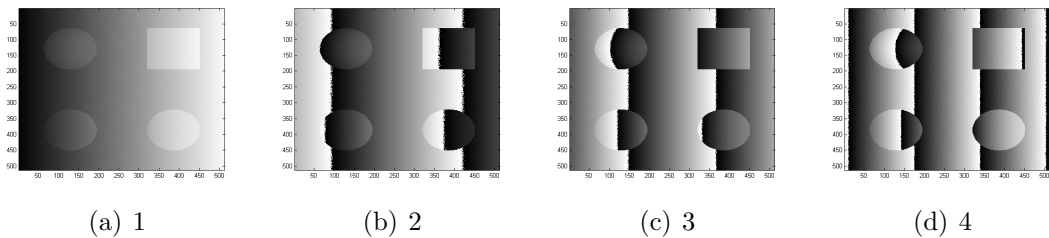


Figura 2.22: Primeras 4 fases discontinuas

La figura 2.23 muestra la resta $\varphi_d(2) - \varphi_c(1)$ y la figura 2.24 muestra los valores "d" para corregir $\varphi_d(2)$.

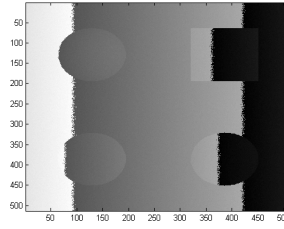


Figura 2.23: Diferencia entre el mapa t=2 y t=1

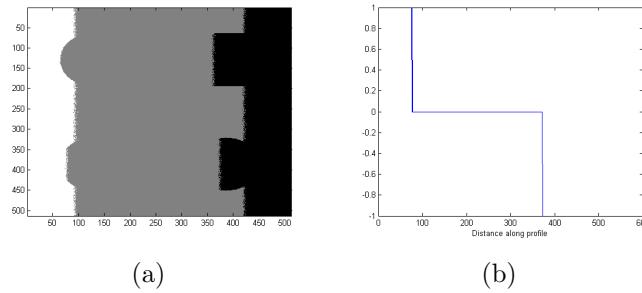


Figura 2.24: Valores "d" para t=2

El proceso se repite hasta llegar al mapa $s = 32$ y obtener su fase continua, el cual se ilustra en la figura 2.25(a), al hacer un corte transversal sobre la fase continua obtenida, es posible observar la topografía de algunos de los objetos figura 2.25(b). Como se indicó en el capítulo anterior, a la fase continua se le debe sustraer la fase de referencia φ_0 que corresponde a la fase para $Z = 0$.

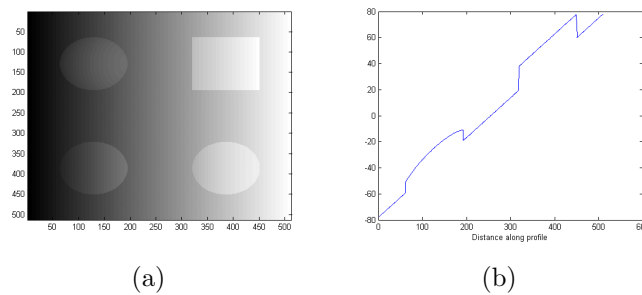


Figura 2.25: Fase continua para s=32

La figura 2.26 muestra $\Delta\varphi = \varphi_s - \varphi_0$ que corresponde a la forma de los objetos simulados.

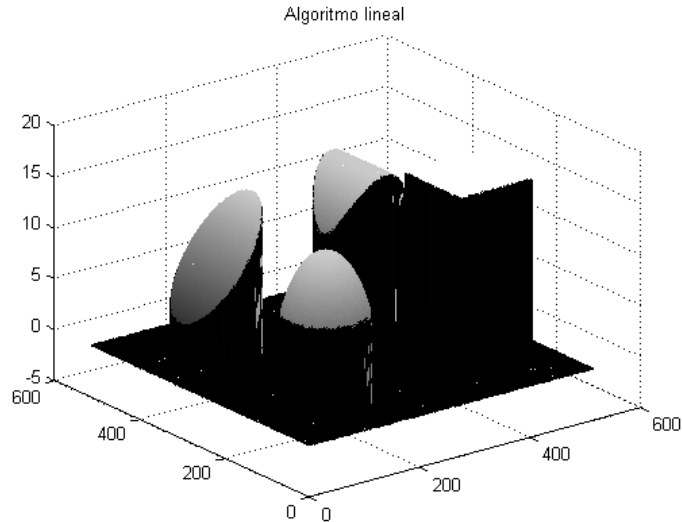


Figura 2.26: Forma de los objetos simulados

En la figura 2.26 se puede observar que los resultados son muy óptimos a pesar que hay presencia de ruido. Se obtiene automáticamente la fase absoluta del objeto, debido a que la fase cero siempre se encuentra en la posición cero, corrigiendo así el problema que se presentaba en el algoritmo espacial en el cual era necesario definir el punto a partir de donde se deseaba comenzar el desenvolvimiento.

2.1.3. Análisis de resultados

Los resultados anteriores indican que niveles de ruido alto afectan el funcionamiento del algoritmo. El procedimiento sugiere que sí existe una propagación de error como en el caso espacial, pero se propaga sobre el eje temporal, obteniéndose puntos erróneos en el mapa de fase final. La influencia del ruido afecta a los valores de intensidad de las franjas las cuales a su vez afectan los valores de fase discontinua. El error se evidencia cuando por influencia del ruido la diferencia entre la fase $\varphi_d(t)$ y $\varphi_c(t - 1)$ encuentra un valor fuera del rango $[-\pi, \pi]$ erróneo.

Debido a que el desenvolvimiento es en la dirección del eje temporal, un error encontrado en el valor "d" se propaga a valores superiores de t. Es decir, entre más alto sea el valor de s más alta es la probabilidad de encontrar valores erróneos. La figura 2.27 muestra los valores de fase para valores de $x = 0, \dots, \Delta/2$ que se emplean en la proyección para garantizar un incremento lineal en el número de franjas. Un punto erróneo corresponde a un punto donde en algún valor de t se ha calculado erróneamente "d". El valor "d" se obtiene del número de veces 2π que hay en la diferencia entre dos fases consecutivas. Como se observa en la figura la diferencia en los valores de los extremos, posiciones cercanas a $x = \pm\Delta/2$, es próximo o igual a π , si por influencia del ruido el valor supera levemente a π , "d" se incrementa en 1 erróneamente. Es decir es más probable encontrar puntos erróneos en los extremos que en el centro, ver figura 2.13 y 2.19.

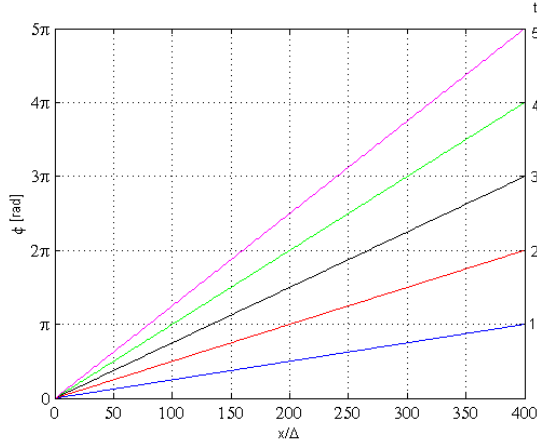


Figura 2.27: Valores de fase empleados

La figura 2.26 muestra que el procedimiento temporal de corrección de fase discontinua permite obtener la fase absoluta de proyección, recuperándose la topografía de objetos aislados.

Finalmente es importante definir el concepto de tasa de éxito, ya que servirá de referencia para comparar los diferentes métodos de corrección temporal de fase discontinua. La tasa de éxito se define como el número de coincidencias acertadas sobre el número total de posibles coincidencias, en este caso corresponde al número de puntos que obtienen una fase final correcta sobre el número total de puntos. De esta manera las figuras 2.14 y 2.20 corresponden a una fase de 4000 filas y 800 columnas donde las columnas corresponden a posiciones de $x = -400$ a $x = 400$ píxeles del proyector ($\Delta = 800$ píxeles). Para un valor de x hay 4000 posibles valores, la tasa de éxito se calcula determinando el número de píxeles por columna que tiene el valor correcto en la fase sobre 4000.

2.2. DESENVOLVIMIENTO DE FASE TEMPORAL EXPONENCIAL (ETPU)

Este algoritmo consiste en que las franjas proyectadas ya no aumentan linealmente sino a razón de una base [8]. Es decir si la base es 2, se proyectan 1, 2, 4, 8, 16, 32, ..., franjas. Esto hace que las etapas de proyección, adquisición y procesamiento se lleven a cabo en menos tiempo, debido a que serán necesarias menos imágenes para obtener la fase continua final. El número de las franjas a proyectar esta dado por $t = base^\nu$ con $t = 0, 1, 2, 3, \dots, \log_{base} s$. Históricamente se encuentra el algoritmo con base 2, la frecuencia varia de la siguiente manera $t = 2^\nu$, $\nu = 0, 1, 2, \dots, \log_2 s$.

Una de las formas para llevar a cabo el desenvolvimiento es la siguiente. Se parte del hecho que la fase continua del mapa 1 es igual a su fase discontinua,

$$\varphi_1^c(m, n) = \varphi_1^d(m, n), \quad (2.21)$$

los términos c y d hacen referencia a la fase continua y discontinua respectivamente. La fase discontinua del mapa $t = 2$ franjas, se desenvuelve de la siguiente manera:

$$\varphi_c(2) = U \{ \varphi_d(2), 2\varphi_c(1) \}. \quad (2.22)$$

$$\varphi_c(2) = \varphi_d(2) - 2\pi \left[\frac{\varphi_d(2) - 2\varphi_c(1)}{2\pi} \right], \quad (2.23)$$

donde $\varphi_d(2)$ denota la fase discontinua del mapa $t = 2$ e igualmente $\varphi_c(2)$ fase continua del mapa $t = 2$. Una vez se obtiene el valor $\varphi_c(2)$, este se usa para desenvolver $\varphi_d(4)$.

$$\varphi_c(4) = U \{ \varphi_d(4), 2\varphi_c(2) \}, \quad (2.24)$$

$$\varphi_c(4) = \varphi_d(4) - 2\pi \left[\frac{\varphi_d(4) - 2\varphi_c(2)}{2\pi} \right]. \quad (2.25)$$

El proceso se repite usando los valores de fase $\varphi(t)$ ($t = 1, 2, 4, 8, \dots, s$). Generalizando las ecuaciones anteriores se obtiene:

$$\varphi_c(2t) = U \{ \varphi_d(2t), 2\varphi_c(t) \}, \quad (2.26)$$

La fase continua para el algoritmo exponencial puede ser obtenida más fácilmente usando la ecuación(ec. 2.27), la cual permite implementarse a cualquier base además que requiere de menos iteraciones en el procesamiento de las imágenes.

$$\varphi_c(\text{base} \cdot t) = \varphi_d(\text{base} \cdot t) - 2\pi \left[\frac{\varphi_d(\text{base} \cdot t) - \text{base} \cdot \varphi_c(t)}{2\pi} \right] \quad (2.27)$$

donde $\varphi_d(\text{base} \cdot t)$ representa la fase discontinua que se esta desenvolviendo, $\varphi_c(t)$ la fase continua del mapa anterior y [...] redondeo al entero más cercano.

Para este algoritmo el operador "d" hará referencia al siguiente término:

$$d = \left[\frac{\varphi_d(\text{base} \cdot t) - \text{base} \cdot \varphi_c(t)}{2\pi} \right], \quad (2.28)$$

esto significa que la ecuación 2.27 puede ser re-escrita de la siguiente manera:

$$\varphi_c(\text{base} \cdot t) = \varphi_d(\text{base} \cdot t) - 2\pi d. \quad (2.29)$$

A continuación se realizará gráficamente el proceso para obtener la fase continua, con base 2 y $s = 32$, implementando la ecuación 2.27.

El primer paso en este proceso es multiplicar la fase continua del mapa 1 por el valor de la base "2", el cual luego será sustraído de la fase discontinua 2, figura 2.28.

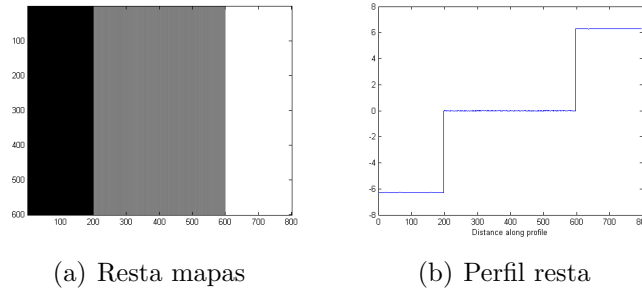


Figura 2.28: Resta de los mapas de fase ($t = 2$) - $2 \cdot (t = 1)$.

el siguiente paso es calcular los valores "d" a partir de la ec. 2.28, estos indican las veces que 2π debe se añadido para corregir la fase discontinua, figura 2.29.

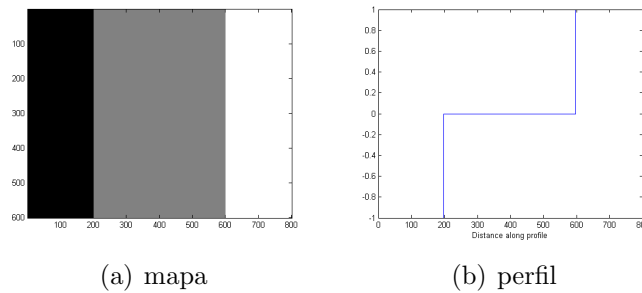


Figura 2.29: Valores de las discontinuidades "d"

Una vez se obtienen los valores "d" estos son multiplicados por 2π y sustraídos del mapa discontinuo que se esta desarrollando ec. 2.29, con lo que finalmente se obtiene la fase continua el mapa 2, figura 2.30.

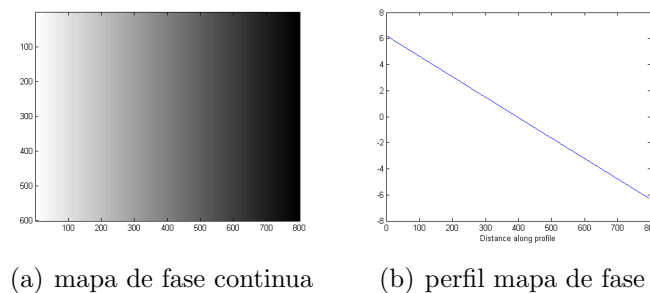


Figura 2.30: Fase continua $t=2$

Esta fase continua obtenida servirá para encontrar el valor de las discontinuidades del siguiente mapa, es decir, del mapa $t = 4$ ya que se está implementando el algoritmo a

base 2. El proceso se repite hasta llegar a la fase continua final que se desea desenvolver, figura 2.31.

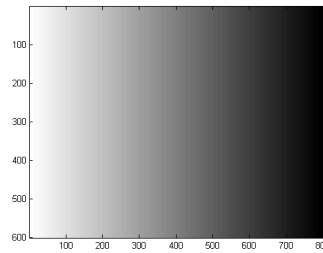


Figura 2.31: Fase continua $t = 32$

La figura 2.32 muestra el funcionamiento del algoritmo: la línea verde indica $\varphi_c(t)$ conocida y calculada en la iteración anterior. La línea roja muestra la fase discontinua $2t$, $\varphi_d(2t)$ que se desea corregir. Como se observa en la figura, al multiplicar por la base 2, $\varphi_c(t)$, las pendientes de $2\varphi_c(t)$ y $\varphi_d(2t)$ se igualan, pero con la diferencia de enteros de 2π que corresponde al valor requerido "d" para corregir $\varphi_d(2t)$.

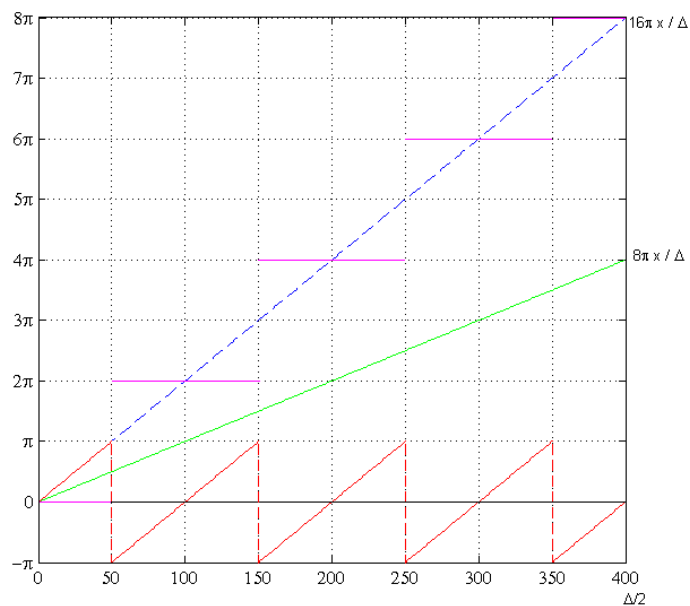


Figura 2.32: Funcionamiento del algoritmo exponencial según pendientes

Como la fase varía según la ecuación 2.8, la figura 2.33 muestra la diferencia del número de imágenes tomadas donde los círculos vacíos indican la variación del algoritmo lineal y los círculos rojos indican los valores de t para base 2. Según la ec. 2.8 se obtiene esta figura para un valor de x . La pendiente corresponde a $\frac{2\pi x}{\Delta}$ rad.

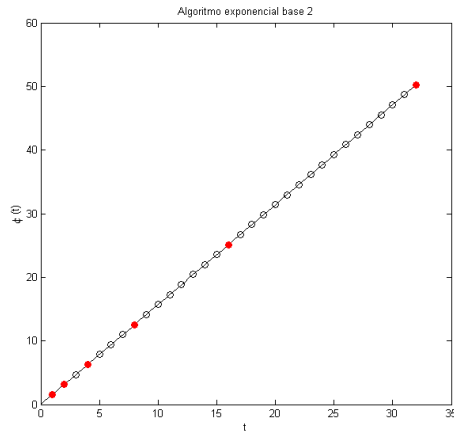


Figura 2.33: φ vs t

La figura 2.34, muestra un esquema del proceso que se lleva a cabo para obtener la fase continua implementando el algoritmo exponencial. Se observa que es similar al algoritmo lineal solo que no se utilizan franjas consecutivas sino que varían a razón de la base.

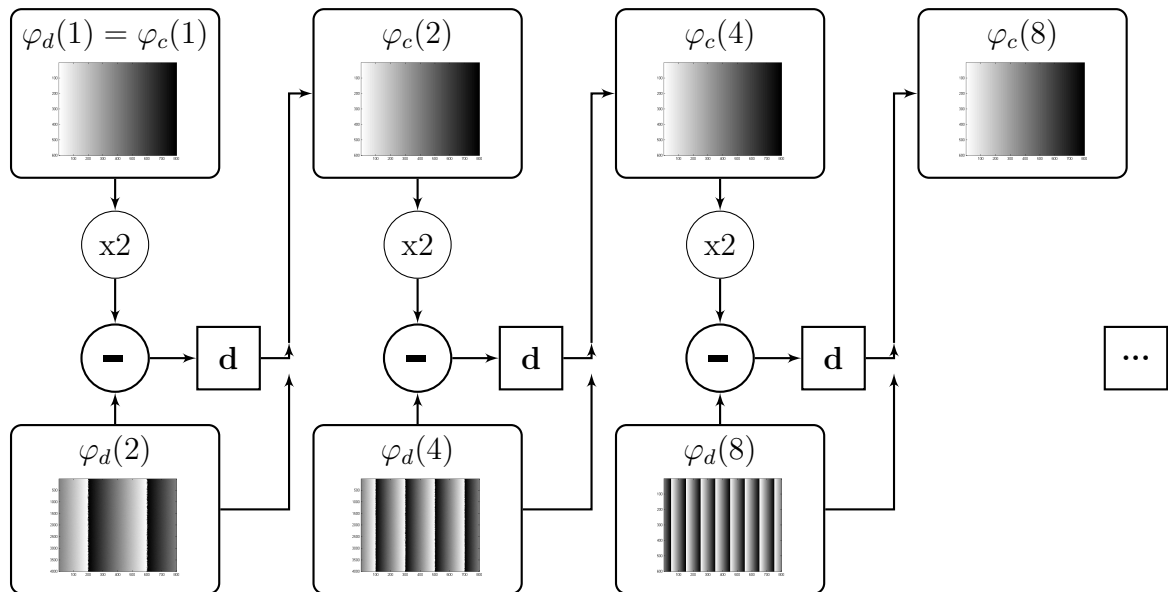


Figura 2.34: Algoritmo Exponencial base 2

2.2.1. Simulación con un objeto plano

Para llevar a cabo la simulación se generan los patrones de franjas partir de las ecuaciones de la sección anterior, ecs. 2.17, 2.18, 2.19 y 2.20 en las cuales se introdujo un ruido gaussiano; esta simulación se implementa usando el algoritmo base 2 y $s = 32$ franjas. Esta parte consta de dos etapas, comportamiento del algoritmo con ruido bajo y ruido alto.

Ruido bajo

En este caso un ruido gaussiano es introducido con $\sigma=5$ NG. Como el algoritmo a implementar es base 2, son necesarias las franjas 1, 2, 4, 8, 16 y 32, figura 2.35, es importante recordar que estas deben ser desfasadas $\pi/2$ e implementando el algoritmo a cuatro imágenes se obtiene las fases discontinuas, figura 2.36.

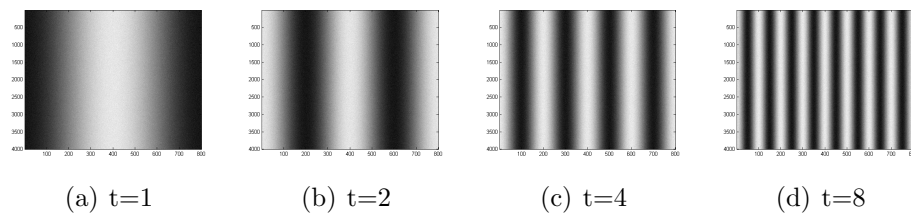


Figura 2.35: Primeros cuatro patrones de franjas exponencial base 2.

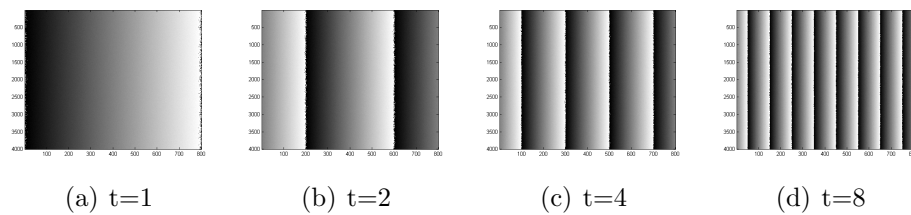


Figura 2.36: Primeras fases discontinuas exponencial base 2

Implementando la ec. 2.27, el primer paso es hacer el producto entre la *base* y el mapa de fase continua anterior, es decir mapa 1, figura 2.37

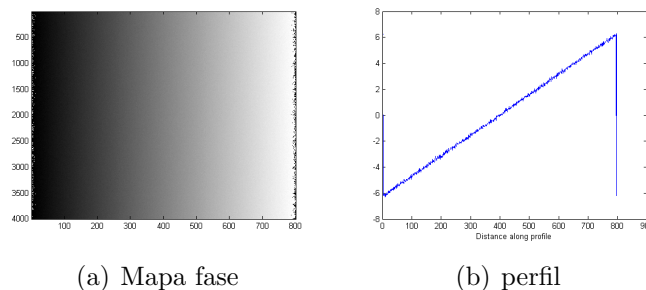


Figura 2.37: Fase continua 1 multiplicada por la base "2"

el segundo paso es hacer la diferencia entre el mapa discontinuo 2 y el resultado del producto entre la base y el mapa 1, figura 2.38

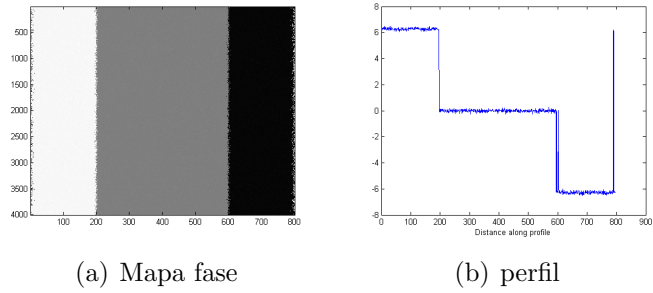


Figura 2.38: Fase continua 1 multiplicada por la base "2"

el tercer paso es obtener los valores d , ec. 2.28, (ver figura 2.39)

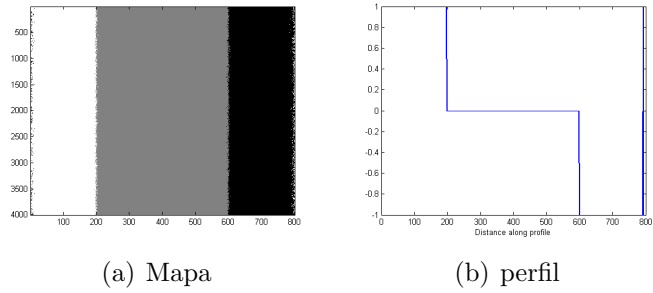


Figura 2.39: Valor de las discontinuidades "d"

estos valores indican cuantas veces 2π debe ser añadido para corregir la fase discontinua del mapa 2, figura 2.40.

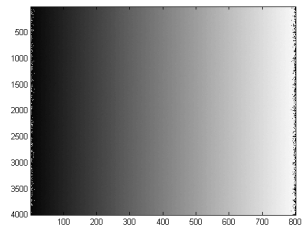


Figura 2.40: Fase continua para $s=2$

La fase continua del mapa 2 es necesaria para calcular los valores "d" del siguiente mapa. El proceso descrito anteriormente se repite hasta obtener la fase continua final en este caso para $s=32$, la cual se muestra en la figura 2.41

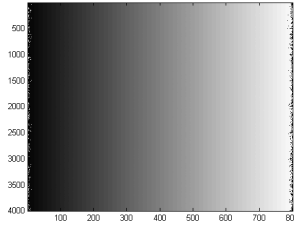


Figura 2.41: Fase continua para $s=32$

Se observa que el error de la fase se encuentra en los extremos, lo cual indica que la pérdida de información en un pixel no afecta sus pixeles vecinos, por ende el error generado representa el error en puntos determinados.

Para comprobar que tan bien se comporta el algoritmo, se realiza la gráfica de tasa de éxito, figura 2.42. Se observa que la probabilidad de éxito del algoritmo es muy alta, incluso mejor que la del algoritmo lineal, para el mismo nivel de ruido aditivo.

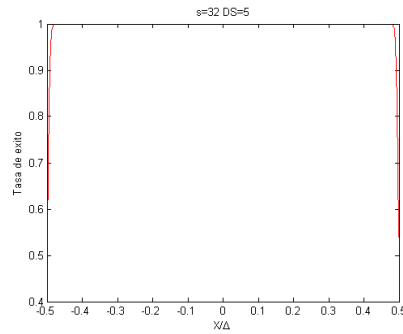


Figura 2.42: Tasa de éxito para ruido bajo exponencial.

Ruido alto

En la sección anterior se observó que el algoritmo exponencial responde muy bien cuando el ruido presente es bajo; ¿Pero que sucede si el ruido introducido es alto?, para responder esto se introducirá un ruido gaussiano de $\sigma = 30$ NG en las franjas. Primero se generan las franjas con ruido gaussiano de $\sigma = 30$ NG, a partir de las ecs. 2.17, 2.18, 2.19

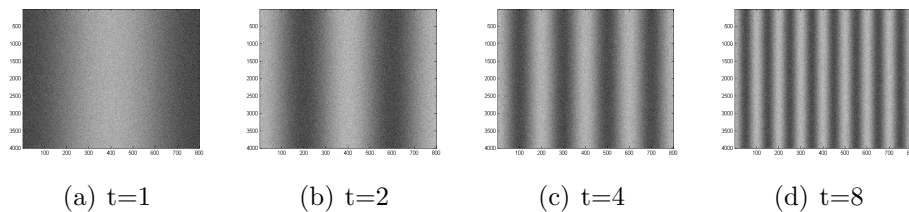


Figura 2.43: Primeros cuatro patrones de franjas exponencial base 2.

Implementando el algoritmo a 4 imágenes se calculan los respectivos mapas de fases discontinuas, figura 2.44.

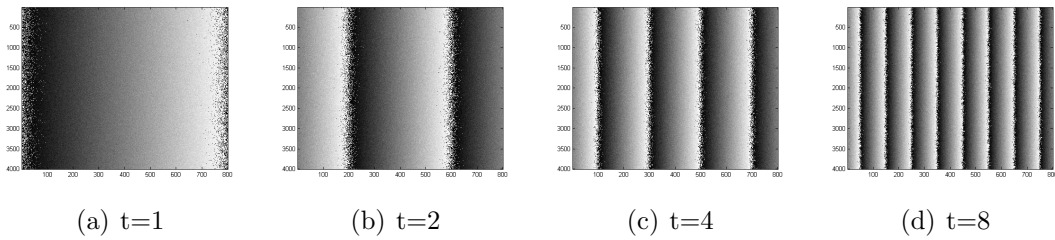


Figura 2.44: Primeros cuatro fases discontinuas exponencial base 2.

Segundo se multiplica por la *base* la fase continua 1, figura 2.45

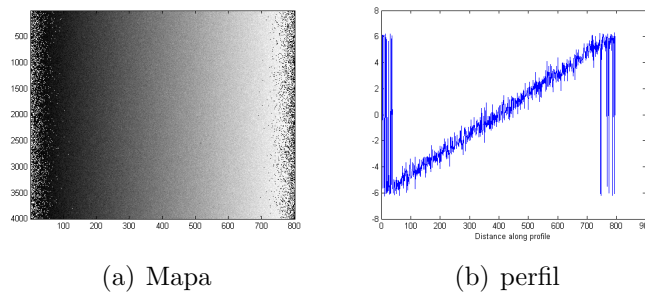


Figura 2.45: Fase continua 1 multiplicada por la base

Tercero se hace la diferencia entre el mapa de fase discontinuo 2, con el resultado obtenido del producto con la base, figura 2.46

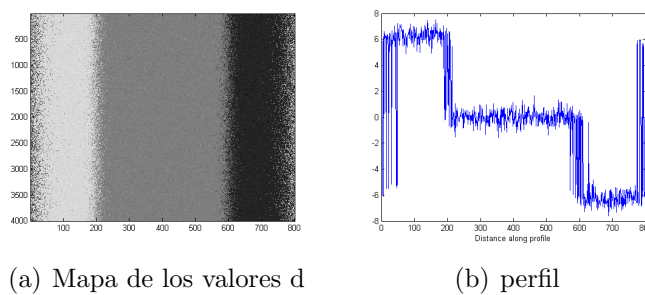


Figura 2.46: Diferencia entre $t = 2 - 2(t = 1)$

cuarto se encuentran los valores " d " utilizando la ec. 2.28, los cuales se ilustran en la figura 2.47

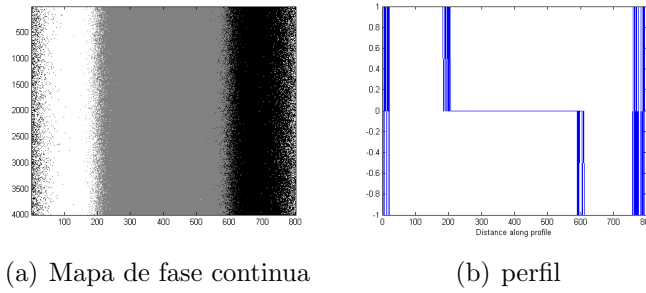


Figura 2.47: Valores "d"

estos valores "d" son multiplicados por 2π y sustraídos de la fase discontinua del mapa 2, con lo que finalmente se obtiene la fase continua del mapa 2, figura 2.48.

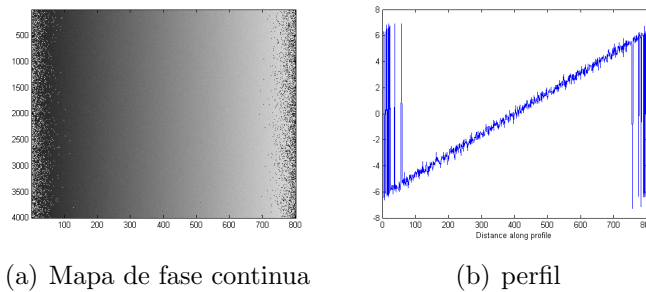


Figura 2.48: Fase continua para el mapa s=2

La fase continua del mapa 2 es necesaria para obtener los valores "d" del siguiente mapa. Así entonces el proceso se repite hasta lograr llegar a la fase continua final que se desea, para este caso para $s = 32$, la cual se muestra en la figura 2.49.

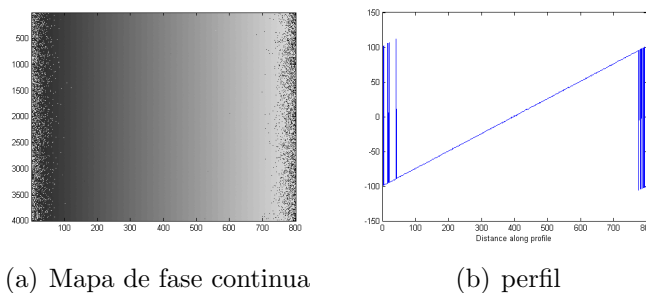


Figura 2.49: Fase continua para el mapa s=32

En la fase continua obtenida se observa que de nuevo el error se encuentra en los extremos y no se propagó a lo largo del resto de la fase como era de esperarse. Esto quiere decir que el error en un píxel no afecto la información de fase en sus píxeles vecinos, que

era uno de los principales problemas que se presentaban con el algoritmo espacial. Ahora para observar la eficiencia del algoritmo a ruido alto, se realiza la gráfica de tasa de éxito. De igual forma se encuentra que el algoritmo exponencial a base 2 es más robusto al ruido aditivo gaussiano.

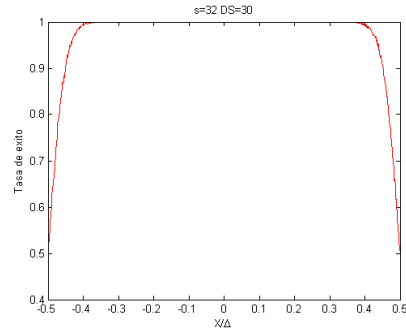


Figura 2.50: Tasa de éxito para ruido alto en algoritmo exponencial.

2.2.2. Simulación con objeto discontinuo

En esta sección se implementa algoritmo exponencial a base 2, con $s = 32$ franjas y una desviación estándar de $\sigma = 5$ NG sobre objetos discontinuos. Se generan las franjas figura 2.51 y estas presentan desplazamientos debido a la topográfica del objeto. La simulación se lleva a cabo como en el caso lineal para un sistema de proyección de franjas telecéntrico con un ángulo de proyección $\theta = 25^\circ$ y campo de observación de $513 \text{ mm} \times 513 \text{ mm}$.

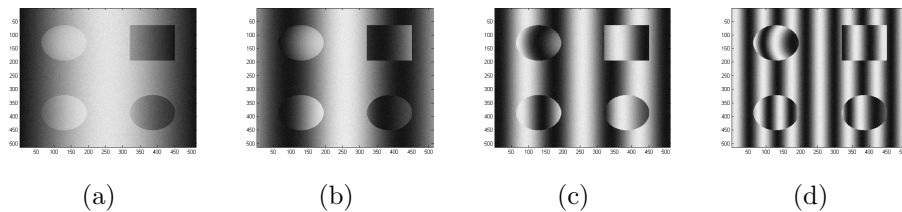


Figura 2.51: Primeras 4 franjas proyectadas exponencial

Las franjas se deben desfazar $\pi/2$, e implementando el algoritmo a cuatro imágenes se obtienen sus mapas de fases discontinuos figura 2.52.

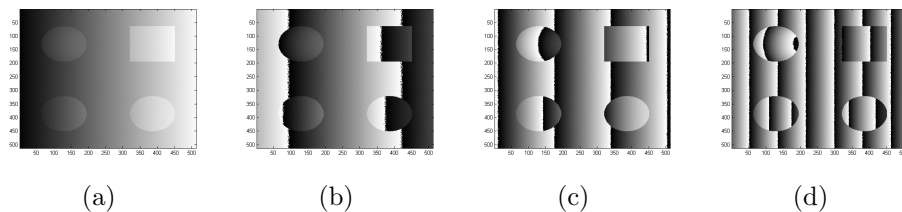


Figura 2.52: Primeras 4 fases discontinuas

Para realizar el desenvolvimiento, es necesario hacer el producto entre la base y la fase del mapa 1, la cual es continua, figura 2.53

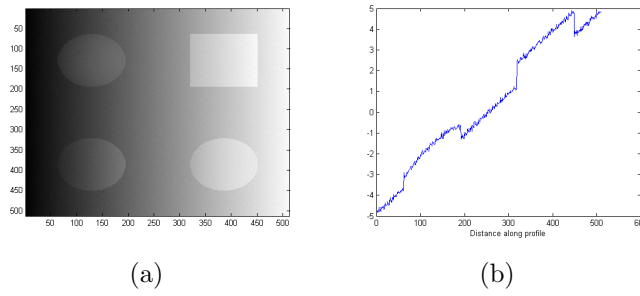


Figura 2.53: 2 veces la fase continua 1

Este producto se sustrae del mapa de fase discontinua 2; la diferencia se ilustra en la figura 2.54

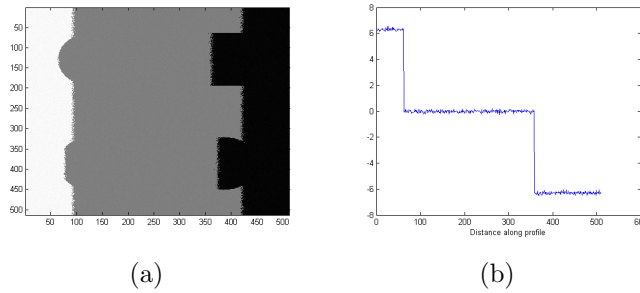


Figura 2.54: Diferencia mapa - 2 veces mapa 1

a continuación se calculan los valores "d" o también conocidos como las 2π discontinuidades, que son las encargadas de corregir la fase discontinua y llevarla a una fase continua, figura 2.55.

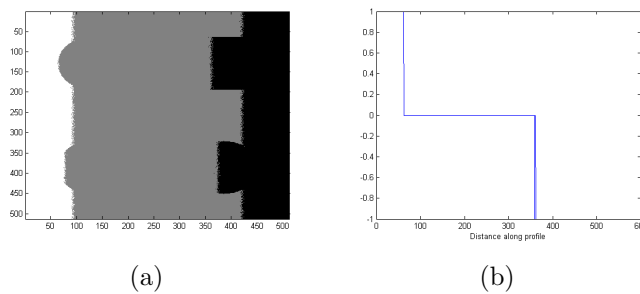


Figura 2.55: Valor de las "d"

finalmente multiplicando estos valores "d" por 2π , y sustrayendolos de la fase discontinua del mapa 2, se obtiene la fase continua para el mapa 2 figura 2.56.

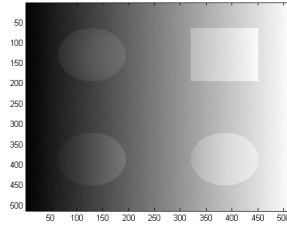


Figura 2.56: Fase continua mapa 2

La fase continua del mapa $t = 2$ es necesaria para calcular las discontinuidades del siguiente mapa. Repitiendo el proceso descrito anteriormente con cada uno de los mapas se logra desenvolver el mapa 32 con lo que se obtiene la fase continua final, figura 2.57.

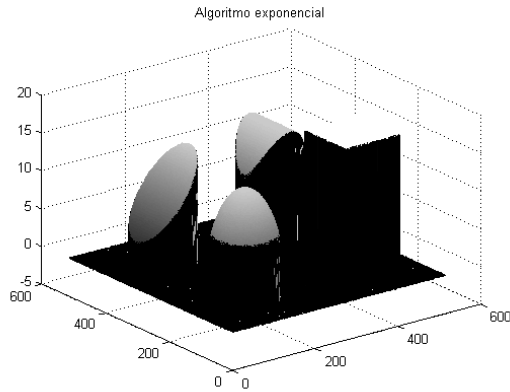


Figura 2.57: Fase absoluta con algoritmo exponencial.

2.2.3. Análisis de resultados

Los resultados anteriores demuestran que el algoritmo exponencial tiene un mejor desempeño que el algoritmo lineal, con puntos erróneos aislados ubicados en los extremos. El error en la determinación del número entero "d" se propaga en el eje temporal, incrementándose la probabilidad al incluir más valores de t . Debido al factor exponencial base 2, el número de imágenes se reduce, reduciéndose la probabilidad de errores en el cálculo de "d". La ubicación de puntos ruidosos en la fase tiene la misma explicación que en el caso lineal. También se encuentra el mismo desempeño al calcular la fase absoluta de proyección. La ventaja adicional consiste en el uso de menos imágenes a proyectar.

2.3. DESENVOLVIMIENTO DE FASE TEMPORAL EXPONENCIAL INVERSO (RETPU)

En este algoritmo se inicia el desenvolvimiento a partir de la máxima densidad de frecuencia ($t = s$), esto significa que la proyección comienza con el número máximo de franjas las cuales se van reduciendo en un factor $t' = base^\nu$ donde $\nu = 0, 1, 2, \dots, \log_{base} s$.

El desenvolvimiento es similar al algoritmo exponencial, excepto que el número de franjas se reduce. Es decir, se inicia desde el punto $t = s$ en vez de $t = 1$. Para los algoritmos anteriores la primera fase discontinua correspondía a $t = 1$, luego $\varphi_c(1) = \varphi_d(1)$. Para este algoritmo, como entre s y $s - 1$ hay una franja de diferencia en la fase, la primera fase continua se obtiene según:

$$U \{ \Delta\varphi_d(s, s - 1), 0 \} = \Delta\varphi_c(s, s - 1) \quad (2.30)$$

Esta fase servirá para corregir los siguientes mapas, cuya ecuación recursiva es:

$$\Delta\varphi_c(s, s - base \cdot t') = \Delta\varphi_c(s - t', s - base \cdot t') + \Delta\varphi_c(s, s - t'), \quad (2.31)$$

donde:

$$\Delta\varphi_c(s - t', s - base \cdot t') = U [\Delta\varphi_d(s - t', s - base \cdot t'), \Delta\varphi_c(s, s - t')], \quad (2.32)$$

el operador U se define ahora de la siguiente manera

$$U [\varphi(1), \varphi(2)] = \varphi(1) - 2\pi \left[\frac{\varphi(1) - T \cdot \varphi(2)}{2\pi} \right], \quad (2.33)$$

En este algoritmo el termino "d" se re-escibe en:

$$d = \left[\frac{\varphi(1) - T \cdot \varphi(2)}{2\pi} \right], \quad (2.34)$$

y el término T , es un término corrector que permite igualar las pendientes de $\varphi(1)$ y $T\varphi(2)$ para poder calcular los valores de "d" correctamente, ec. 2.34. Este término está definido como:

$$T = \frac{t' - (t' - 1)}{(t' - 1)}, \quad (2.35)$$

en la ecuación 2.35 t' representa el número de franjas y $(t' - 1)$ hace referencia al t' anterior. El valor de T variará según el número de franjas que se estén operando, pero cuando el algoritmo inverso es base 2 el término T siempre es igual a 1.

Para entender mejor el algoritmo se ilustrará asumiendo $s = 32$ y base 2 esto quiere decir que son necesarias las franjas 32, 31, 30, 28, 24, 16 y 0. El primer paso es hacer el desenvolvimiento para la diferencia del mapa 32 y mapa 31, $\varphi_1 = \varphi_d(32) - \varphi_d(31)$ figura

2.58 y $\varphi_2 = 0$, ec. 2.30. La figura 2.59 muestra los valores de "d" obtenidos del redondeo al entero más cercano de $\varphi_1/2\pi$.

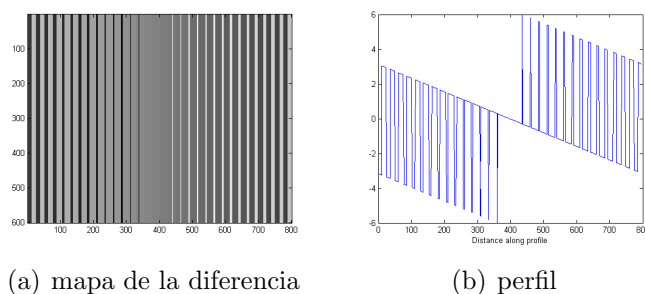


Figura 2.58: resta del mapa s y s-1

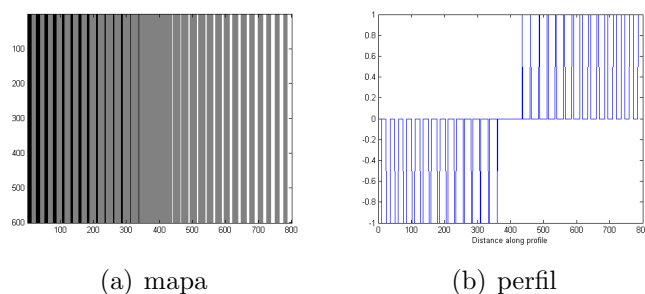


Figura 2.59: Valor de las 2π discontinuidades

Obtenidos los valores d , se multiplican por 2π y se adicionan o sustraen de la diferencia $\varphi_d(32) - \varphi_d(31)$ generando así la fase continua de la diferencia, figura 2.60. Esta fase sirve para corregir la fase de los valores sucesivos de $t' = 2, 4, \dots$

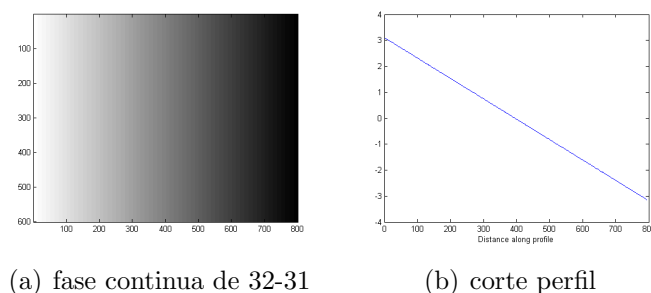


Figura 2.60: fase continua de la diferencia $\varphi_d(32) - \varphi_d(31)$

La diferencia de fase continua entre $t = 32$ y $t = 30$ se obtiene de las ecs 2.31 y 2.32:

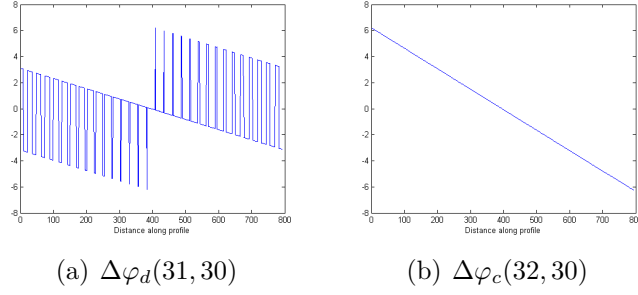


Figura 2.61: Proceso para obtener fase continua 32,30

$$\Delta\varphi_c(32,30) = \Delta\varphi_c(31,30) + \Delta\varphi_c(32,31), \quad (2.36)$$

$$\Delta\varphi_c(31,30) = U[\Delta\varphi_d(31,30), \Delta\varphi_c(32,31)]. \quad (2.37)$$

En la ecuación 2.36 se observa que para obtener la fase continua es necesario adicionar los valores $\Delta\varphi_c(31,30)$ (se obtiene con la ecuación 2.37) y $\Delta\varphi_c(32,31)$ (fase continua mostrada en la figura 2.60). La figura 2.61(a) muestra la diferencia de fase discontinua entre 31 y 30, que al aplicar el operador $U\{\dots\}$, tomando como base $\Delta\varphi_c(32,31)$, se corrige y permite obtener $\Delta\varphi_c(31,30)$. Al adicionar $\Delta\varphi_c(31,30)$ y $\Delta\varphi_c(32,31)$ se obtiene $\Delta\varphi_c(32,30)$, cuyo perfil se muestra en la figura 2.61(b). La siguiente diferencia a corregir es $\Delta\varphi_d(32,28)$ que emplea la diferencia de fase corregida $\Delta\varphi_c(30,28)$ y la diferencia de fase actual calculada $\Delta\varphi_c(32,30)$. $\Delta\varphi_c(30,28)$ usa el operador $U\{\dots\}$ tomando como base $\Delta\varphi_c(32,30)$. Al sumar $\Delta\varphi_c(30,28)$ y $\Delta\varphi_c(32,30)$ se obtiene $\Delta\varphi_c(32,28)$. El proceso se repite hasta lograr obtener la diferencia de fase continua de $\Delta\varphi_c(32,0)$ que es la fase desenvuelta de 32.

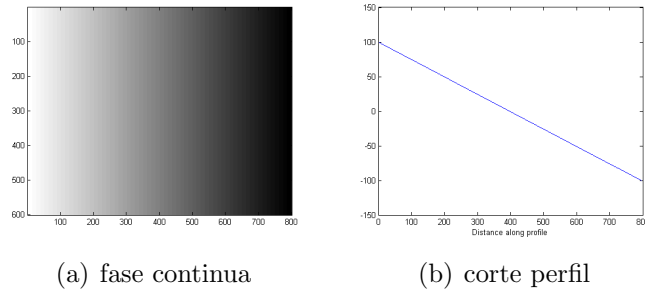


Figura 2.62: fase continua de 32 por algoritmo inverso

La figura 2.63 muestra las fases lineales ubicadas en el sistema coordenado del proyector para el algoritmo exponencial inverso base 2. La secuencia de proyección corresponde a $t = 32, 31, 30, 28, 24, 16, 0$. La fase continua para $t = 32$ se obtiene, según las ecs. 2.30 y 2.31, de la suma acumulada de diferencias consecutivas de fases discontinuas corregidas. Con el fin de corregir la diferencia de 2 fases discontinuas consecutivas, se usa el operador $U\{\dots\}$

tomando como segunda variable la fase acumulada anterior, ec. 2.31. De esta manera, cuando se adquiere y calcula la fase para $t = 24$, ya se tiene acumulada la diferencia de 32 a 28 ($\Delta\varphi_c(32, 28)$) y se desea corregir la diferencia de fase discontinua entre 28 y 24. Nótese que la variación del número de franjas entre las diferencias de fases discontinuas a corregir siempre es igual a la variación del número de franjas de la diferencia de fase acumulada. Entonces con $\Delta\varphi_c(32, 28)$ para 4 franjas de variación en todo el campo, la diferencia de fase discontinua a corregir tiene también 4 franjas, 28 a 24. Esto garantiza que las pendientes son iguales, al utilizar el operador U se obtienen los números enteros de 2π necesarios para corregir la diferencia de fase discontinua consecutiva. La igualdad de pendientes garantiza el correcto cálculo de "d", independiente del valor de t ; como se indica en la figura 2.63.

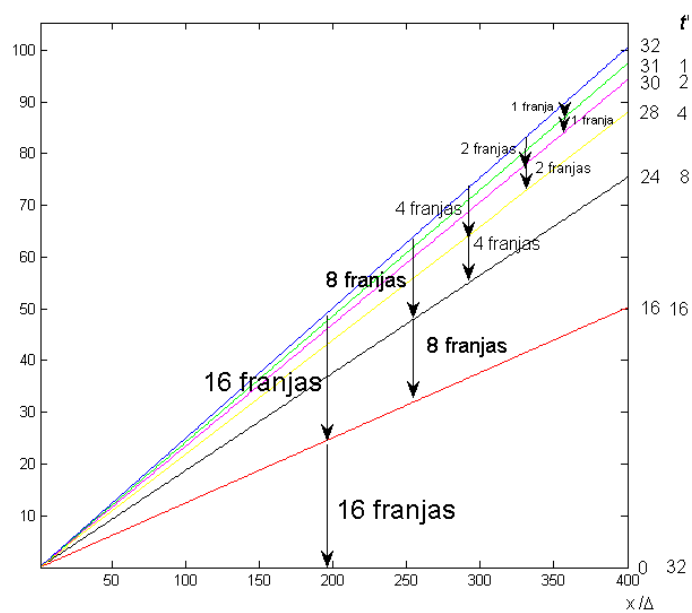


Figura 2.63: secuencia de proyección algoritmo inverso.

La figura 2.64 ilustra el proceso que debe llevarse a cabo para obtener la fase continua implementando el algoritmo exponencial inverso a base 2 con máxima frecuencia 32. Vale recordar que se calculan las fases continuas de las diferencias y estas debe ser adicionadas la fase continua anterior, ec. 2.36.

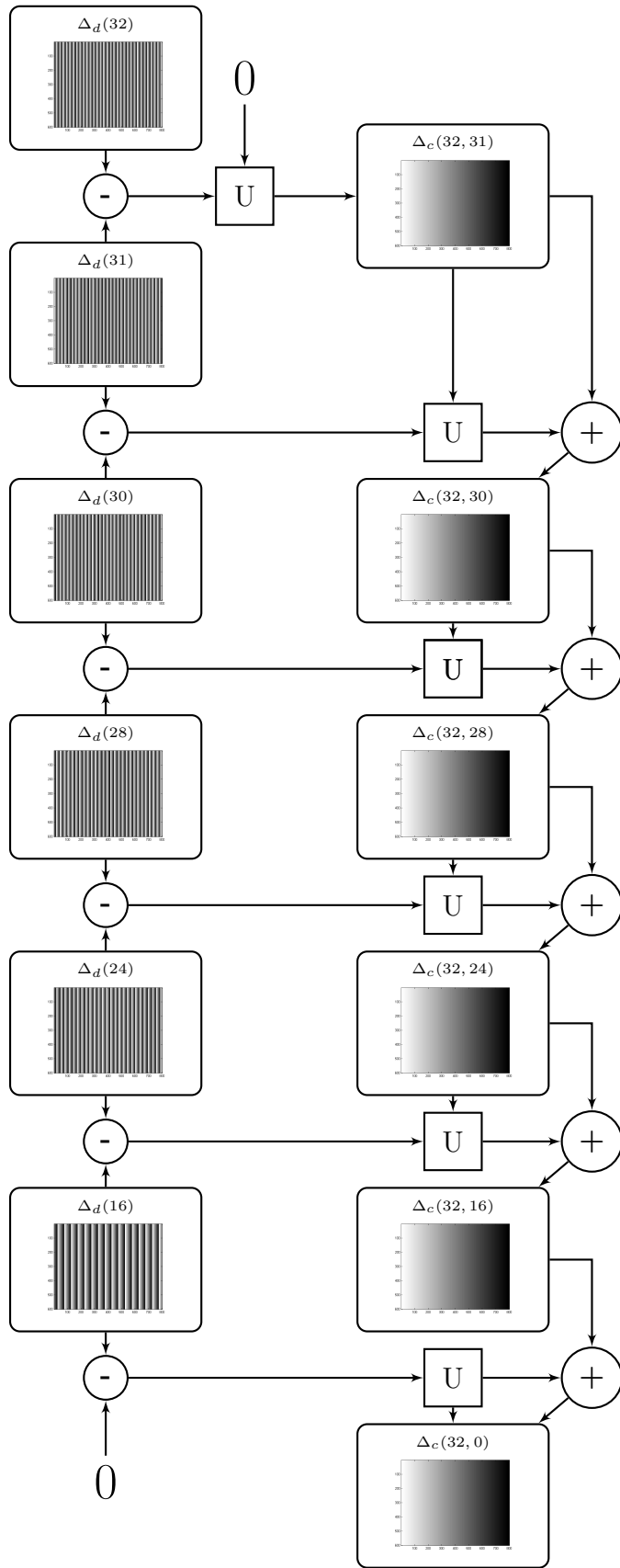


Figura 2.64: Algoritmo inverso

2.3.1. Simulación con un objeto plano

En esta sección se implementará el algoritmo exponencial inverso a base 2 simulado con un objeto plano; en el cual se introducirá un ruido gaussiano en la franjas de media cero y desviación estándar σ en NG para determinar que tan bien se comporta este algoritmo en presencia de ruido.

Ruido bajo

Se evaluará el algoritmo en presencia de un ruido gaussiano con $\sigma = 5$ NG en las franjas, permitiendo estudiar el comportamiento del algoritmo inverso en presencia de ruido bajo.

El primer paso es generar las franjas con el ruido introducido. Debido a que se va a implementar el algoritmo exponencial inverso a base 2 y máxima frecuencia 32 son necesarias las franjas 32, 31, 30, 28, 24, 16, 0; la figura 2.65 muestra los primeros cuatro patrones de franjas.

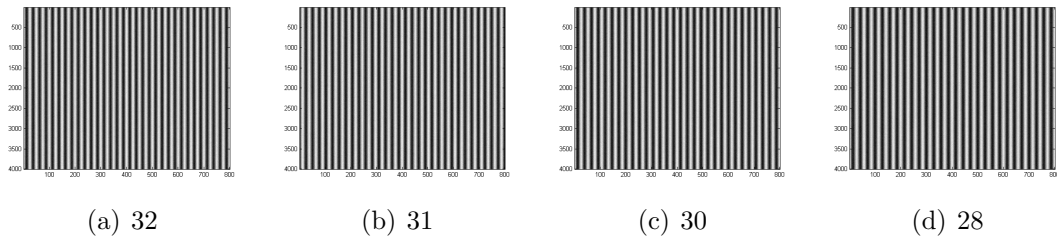


Figura 2.65: Franjas con error introducido, algoritmo inverso.

Desfasando los patrones de franjas $\pi/2$ e implementando el algoritmo a cuatro imágenes se obtienen las respectivas fases discontinuas, estas se ilustran en la figura 2.66.

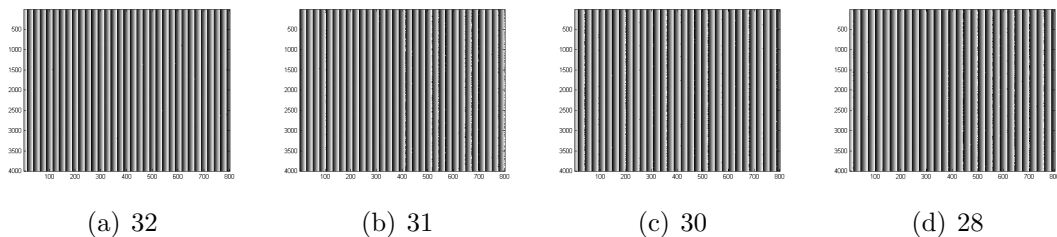


Figura 2.66: Fases discontinuas algoritmo inverso ruido bajo.

Una vez obtenidas las fases discontinuas de cada uno los patrones de franjas se procede a hacer la diferencia entre el mapa 32 y 31, figura 2.67.

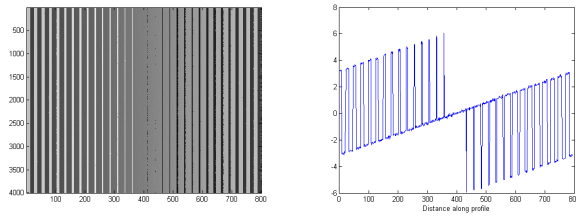


Figura 2.67: Diferencia entre el mapa 32 y 31.

A esta diferencia se le aplica el operador U , tomando como segunda variable $\varphi_2 = 0$ y generando así la fase continua de la diferencia entre el mapa 32 y 31, figura 2.68.

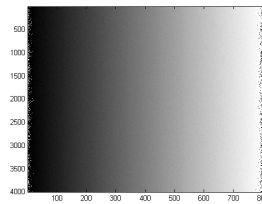


Figura 2.68: Fase continua de la diferencia entre el mapa 32 y 31.

Con la fase continua obtenida, figura 2.68 y la diferencia de fase discontinua entre el mapa 31 - 30, figura 2.69, se aplica el operador U tomando como segunda variable la diferencia continua de los mapas 32 y 31 generando la fase continua de 31-30.

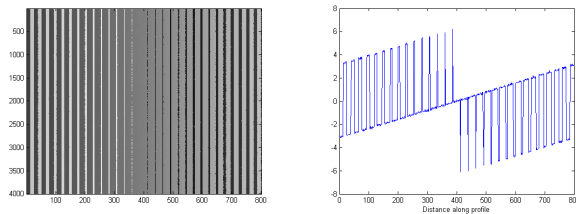


Figura 2.69: Diferencia entre el mapa 31 y 30.

Debido que la fase continua obtenida es de la diferencia 31 - 30, se debe adicionar la fase continua de la diferencia 32 - 31, lo que genera la fase 32 - 30, figura 2.70

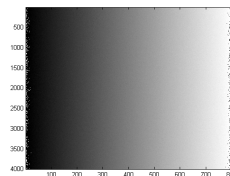


Figura 2.70: Fase continua 32-30

El proceso descrito anteriormente se repite hasta lograr obtener la fase continua 32-0, que es la fase continua de interés, figura 2.71.

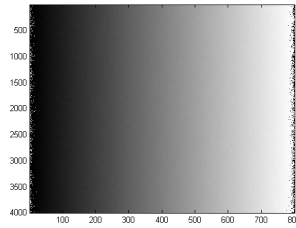


Figura 2.71: Fase continua 32-0

Para saber que tan buenos son los resultados obtenidos con el algoritmo exponencial inverso, se realiza la gráfica de la tasa de éxito la cual muestra que tan estable es el algoritmo con ruido gaussiano de $\sigma = 5$ NG figura 2.72.

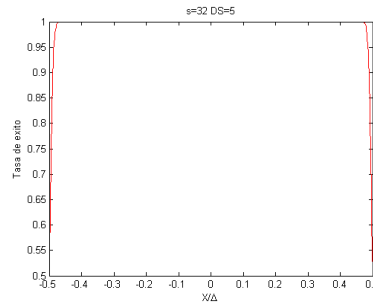


Figura 2.72: Tasa de éxito con ruido bajo algoritmo inverso

En la figura 2.72 se observa que el algoritmo inverso es muy estable en la mayoría de los puntos pero nuevamente en los extremos se presenta error, como se observó en la fase final figura 2.71, por ende la tasa de éxito en estos no logran el valor de 1 pero tampoco toma valores de cero similar a lo que sucede con el algoritmo exponencial.

Ruido alto

En esta sección se implementa el algoritmo a base 2 con máxima frecuencia 32, con un ruido en las franjas de $\sigma = 30$ NG, figura 2.73.

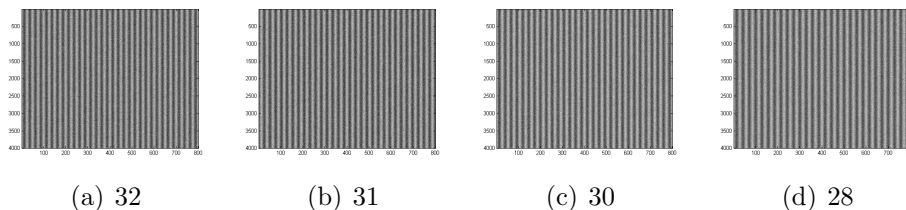


Figura 2.73: Franjas con error alto, algoritmo inverso.

Con el algoritmo a cuatro imágenes se obtienen las respectivas fases discontinuas, figura 2.74.

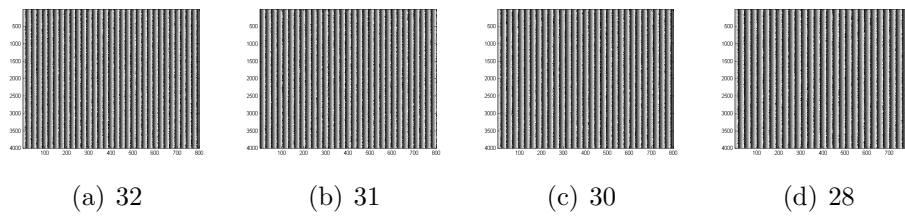


Figura 2.74: Fases discontinuas algoritmo inverso ruido alto.

Una vez se obtienen las fases discontinuas, el primer paso es hacer la diferencia entre el mapa 32-31, figura 2.75.

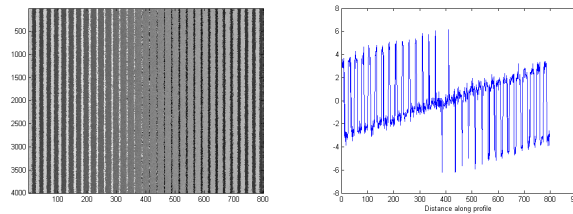


Figura 2.75: Diferencia entre el mapa 32 y 31.

Aplicando el operador U , se obtiene la fase continua de la diferencia 32-31, figura 2.76

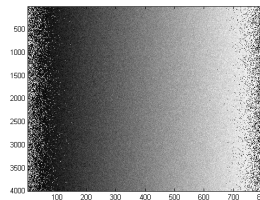


Figura 2.76: Fase continua de diferencia entre mapa 32 y 31 error alto.

luego se hace la diferencia entre el mapa 31 -30; se aplica el operador U con la fase continua calculada anteriormente, el cual genera la fase continua de la diferencia 31-30, figura 2.77

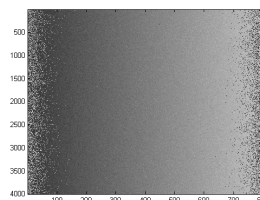


Figura 2.77: Fase continua de diferencia entre mapa 31 y 30 error alto.

finalmente adicionando las fases continuas calculadas anteriormente figuras 2.76 y 2.77 se obtiene la fase continua 32-30, figura 2.78; el proceso se repite hasta obtener la fase continua 32-0.

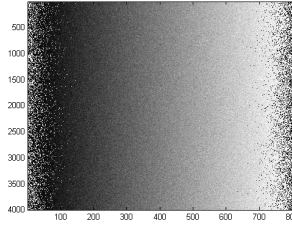


Figura 2.78: Fase continua 32-30 con error alto.

Para evaluar la tasa de éxito del algoritmo inverso con un ruido gaussiano de $\sigma = 30$ NG se realiza la gráfica de tasa de éxito vs x/Δ , figura 2.79. Se observa que el algoritmo presenta buen comportamiento ya que no toma valores de cero y tiene un rango amplio con tasa de éxito 1. Nuevamente la probabilidad de éxito más baja se encuentra en los extremos, esto sucede por que los puntos en esas posiciones presentan cambios más abruptos que los puntos que se encuentran en el centro.

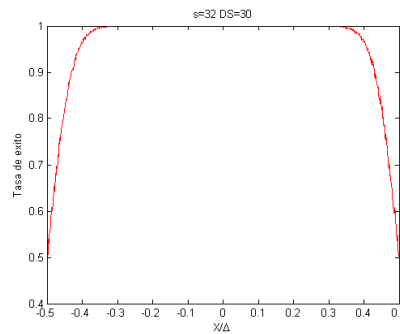


Figura 2.79: Tasa de éxito ruido alto inverso

2.3.2. Simulación con un objeto discontinuo

En esta etapa se implementa el algoritmo inverso a base 2 con máxima frecuencia 32 sobre objetos que presentan discontinuidades. La simulación se lleva a cabo como en el caso anterior para un sistema de proyección de franjas telecéntrico con un ángulo de proyección $\theta = 25^\circ$ y campo de observación de $513 \text{ mm} \times 513 \text{ mm}$.

Primero se genera el patrón de franjas deformado debido a la topografía del objeto con un ruido de $\sigma = 5$ NG, figura 2.80, e implementando el algoritmo a cuatro imágenes se obtienen las fases discontinuas, figura 2.81.

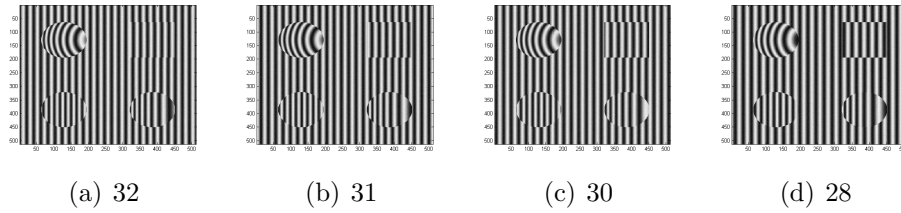


Figura 2.80: Primeras 4 franjas proyectadas inverso

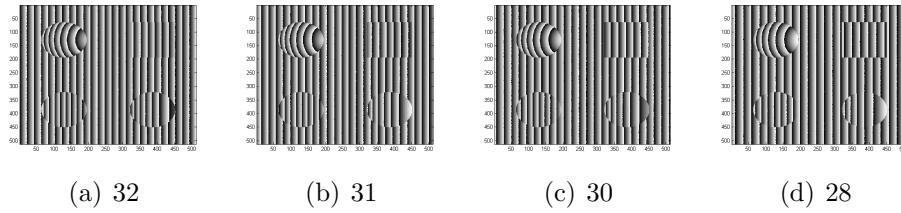


Figura 2.81: Primeras 4 fases discontinuas inverso

Con las fases discontinuas obtenidas, se lleva a cabo el desenvolvimiento, para esto es necesario calcular la diferencia del mapa 32-31, figura 2.82

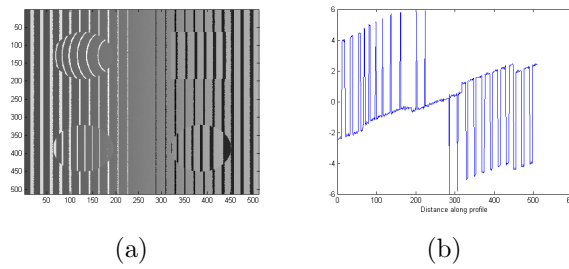


Figura 2.82: Diferencia mapa 32-31

aplicando el operador U se obtiene el primer mapa de fase continua, figura 2.83, el cual es necesario para realizar el desenvolvimiento del siguiente mapa.

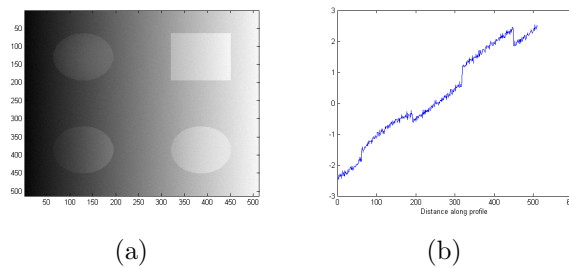


Figura 2.83: Fase continua mapa 32-31

Este proceso se repite hasta obtener la fase continua final deseada en este caso 32.

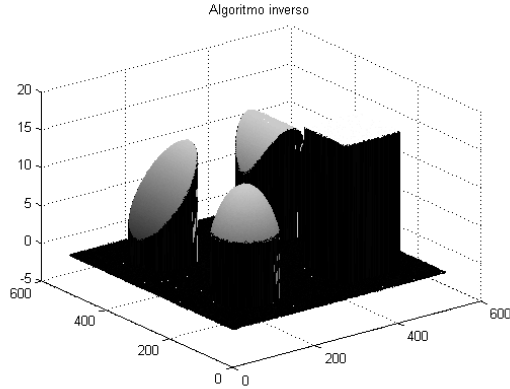


Figura 2.84: Fase continua objeto mapa 32 inverso

2.3.3. Análisis de resultados

El algoritmo presenta un comportamiento similar al encontrado en los algoritmos precedentes: puntos aislados erróneos en los extremos de la imagen, capacidad para calcular fase absoluta y resolver objetos aislados. Como los anteriores, el algoritmo es sensible al nivel de ruido adicionado a las imágenes de las franjas. La tasa de éxito será empleada para comparar los diferentes algoritmos en idénticas condiciones experimentales o teóricas. En este algoritmo la fase continua del sistema de franjas con mayor número de franjas proyectadas se obtiene de la suma acumulada de las diferencias continuas de fases discontinuas consecutivas obtenidas del operador U según ec. 2.32. La figura 2.85 muestra la diferencia de fase discontinua consecutiva para t y $2t$, $\varphi_d(t, 2t)$ (azul) y la diferencia de fase acumulada continua $\Delta\varphi_c(s, t)$ (rojo): se observa que las pendientes son iguales, esto garantiza que $U\{\dots\}$ determina los valores correctos de "d". Este comportamiento es independiente de la fase empleada y del valor de t .

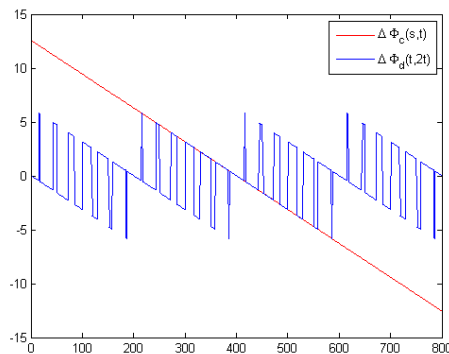


Figura 2.85: diferencia de fase discontinua consecutiva para t y $2t$, $\varphi_d(t, 2t)$

2.4. DESENVOLVIMIENTO DE FASE TEMPORAL GENERALIZADO (GTPU)

Este algoritmo consta de patrones de franjas los cuales no presentan ninguna relación de un patrón con el otro. Para este algoritmo se seleccionan patrones aleatorios y se calcula la fase del patrón final a partir de los demás, el único cuidado que debe tenerse es que siempre se debe proyectar el patrón que presenta una sola franja en todo el espacio, es decir, $t = 1$.

Como se ha mostrado anteriormente, existe una relación lineal entre φ y t , la cual también se cumple en GTPU: $\varphi_c(t_1)/t_1 = \varphi_c(t_2)/t_2$, entonces $\varphi_c(t_2) = k\varphi_c(t_1)$, donde

$$k = \frac{t_2}{t_1}, \quad (2.38)$$

este valor k relaciona las pendientes de dos patrones de franjas seleccionados, permitiendo llevar a cabo correctamente el desenvolvimiento. En la ecuación 2.38, los términos t_2 y t_1 hacen referencia al número de franjas presentes en los respectivos mapas. A partir de esta relación las ecuaciones se modifican de la siguiente manera, ecs 2.39 y 2.41

$$\varphi_c(t_2) = U \{ \varphi_d(t_2), \varphi_c(t_1) \}, \quad (2.39)$$

Donde el operador $U\{\dots\}$ esta definido de la siguiente manera

$$U \{ \varphi_1, \varphi_2 \} = \varphi_1 - 2\pi \left[\frac{\varphi_1 - k \cdot \varphi_2}{2\pi} \right], \quad (2.40)$$

esto quiere decir que la fase continua para el algoritmo generalizado se obtiene de la siguiente manera:

$$\varphi_c(t_2) = \varphi_d(t_2) - 2\pi \left[\frac{\varphi_d(t_2) - k \cdot \varphi_c(t_1)}{2\pi} \right], \quad (2.41)$$

Para que el desenvolvimiento pueda llevarse a cabo deben seleccionarse como mínimo dos patrones de franjas. Un punto a tenerse en cuenta es que el término k , ec. 2.38 es un factor que controla el ruido del algoritmo. Si $k = 2$, el ruido presente es igual que en algoritmo exponencial base 2; si $k \neq 2$, el algoritmo GTPU muestra flexibilidad para el control del ruido; si $k < 2$ el comportamiento de GTPU es mejor que en otros métodos TPU en términos de inmunidad al ruido. Lo dicho anteriormente implica que GTPU es capaz de desenvolver los mapas de fase con menos ruido tomando valores de k pequeños comparado con los otros algoritmos. Para un entorno más favorable con alta relación señal a ruido, GTPU requiere menos patrones de franjas, tomando valores de k más grandes.

Para que sea fácil la comprensión de este algoritmo se realizará de manera gráfica; primero se deben seleccionar los patrones de franjas con los que se van a trabajar, para este caso se implementaran cinco patrones $t = 1, 7, 13, 24, 32$. Aplicando el algoritmo a 4 imágenes se obtienen las respectivas fases discontinuas, figura 2.87

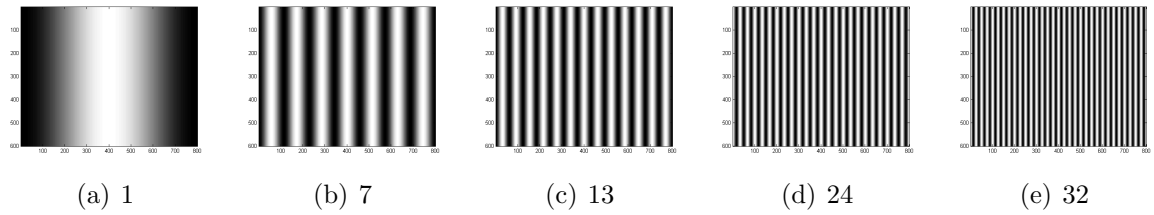


Figura 2.86: Numero de franjas utilizado algoritmo generalizado

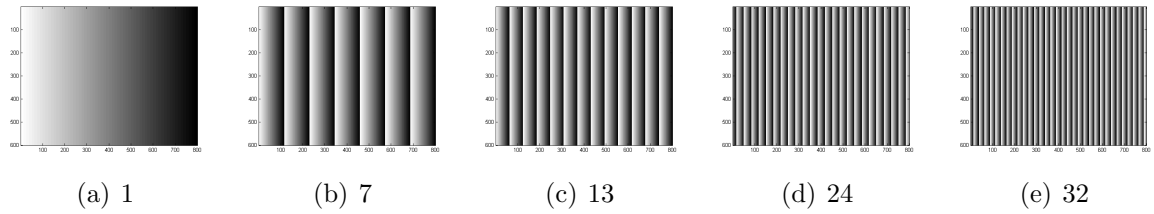


Figura 2.87: Fases discontinuas algoritmo generalizado

Una vez se obtienen las respectivas fases discontinuas, para poder aplicar el operador U , ec. 2.40 se debe hacer el producto de la fase continua por el valor de k que este caso corresponde a 7, figura 2.88,

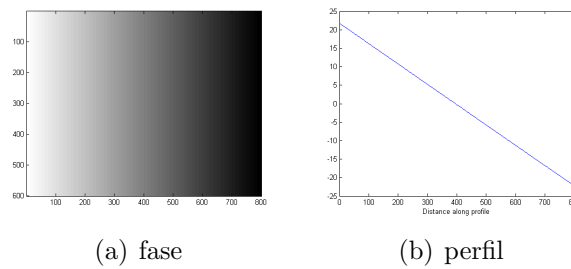


Figura 2.88: k veces fase continua 1

al hacer este producto se igualan las pendientes de las fase 1 con la fase 7 permitiendo así aplicar el operador U para obtener la fase continua del mapa 7 ,ver figura 2.89.

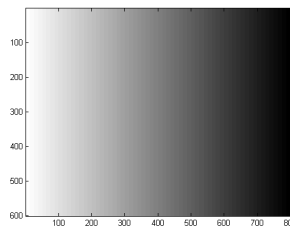


Figura 2.89: Fase continua mapa 7

La fase continua del mapa 7 será necesaria para llevar a cabo el desenvolvimiento del siguiente mapa "13"; el proceso descrito anteriormente se repite hasta obtener finalmente la fase continua del mapa 32, figura 2.90.

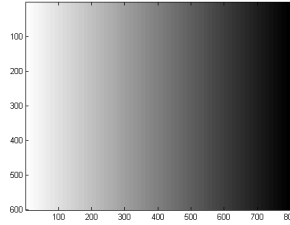


Figura 2.90: Fase continua en 32 algoritmo generalizado

Como se dijo anteriormente el factor k garantiza que las pendientes de φ_1 y $k\varphi_2$ en la ec. 2.40 sean iguales. De esta manera $\left[\frac{\varphi_1 - k\varphi_2}{2\pi}\right]$ genera los valores de "d" requeridos para corregir la diferencia de fases discontinuas consecutivas.

La figura 2.91 muestra un esquema del proceso que es necesario llevar a cabo para obtener la fase continua implementando el algoritmo exponencial generalizado. En este caso el esquema se muestra con las franjas seleccionadas anteriormente, pero vale recordar que estas pueden ser elegidas a interés del usuario, la única que siempre debe estar presente es la inicial $t = 1$.

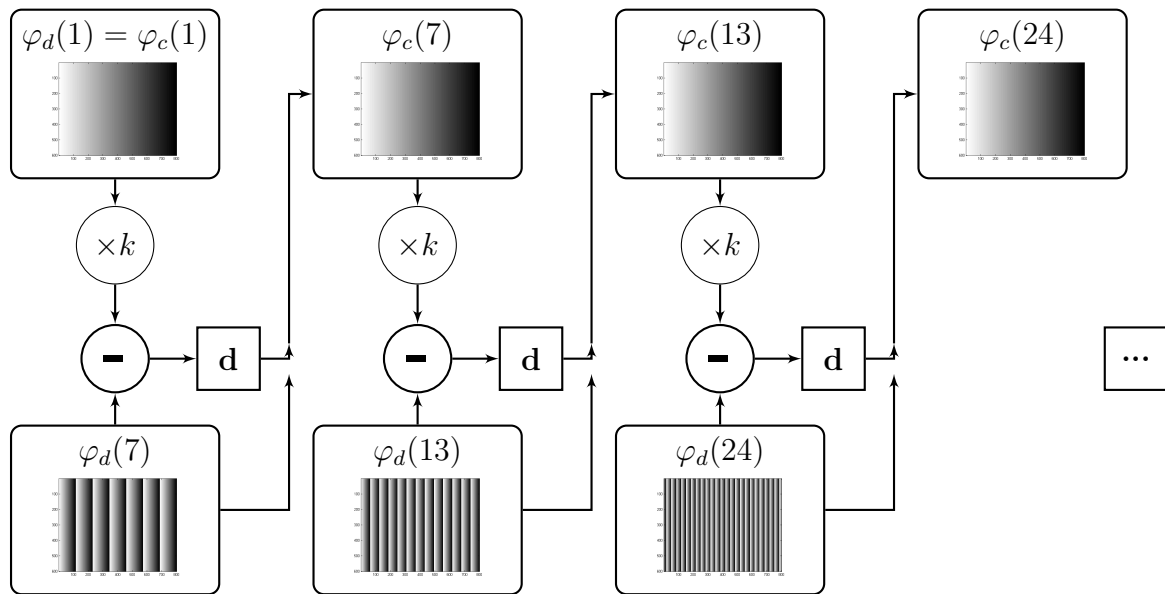


Figura 2.91: Algoritmo Generalizado

2.4.1. Simulación con objeto plano

En esta sección se evalúa el algoritmo sobre un objeto plano introduciendo un ruido gaussiano de media cero y desviación estándar σ en las franjas proyectadas para determinar su comportamiento en presencia de ruido bajo y alto. Las franjas que se utilizarán para desenvolver la fase continua son 1, 7, 13, 24, 32.

Ruido bajo

Para evaluar el algoritmo con ruido bajo se introduce ruido gaussiano de $\sigma = 5$ NG en las franjas simuladas, y a partir de esto se calculará la fase continua implementado el algoritmo generalizado. La figura 2.92 muestra los primeros 4 patrones de franjas con el error introducido, con las cuales implementando el algoritmo a cuatro imágenes se obtienen sus respectivas fases discontinuas, figura 2.93.

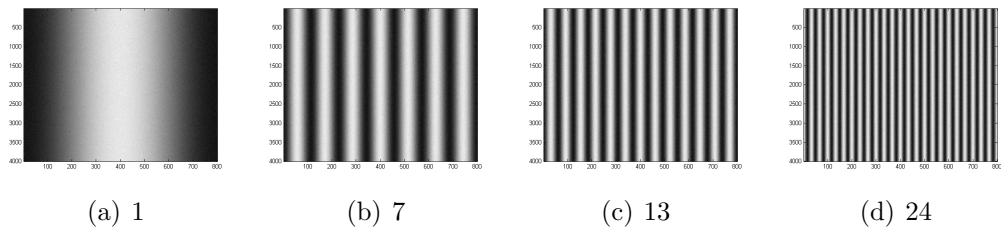


Figura 2.92: Franjas con error introducido, algoritmo generalizado.

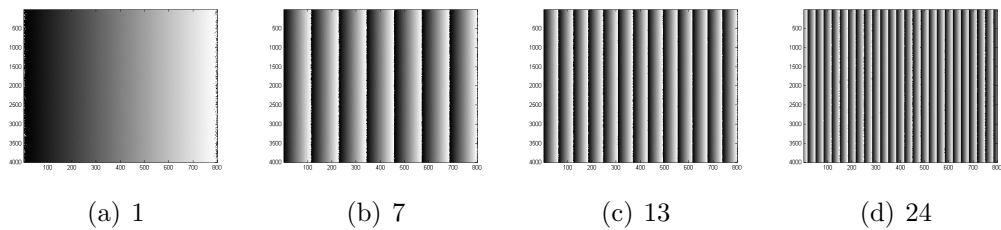


Figura 2.93: Fases discontinuas, algoritmo generalizado.

El primer paso es hacer el producto entre el mapa 1 y el valor de k , figura 2.94

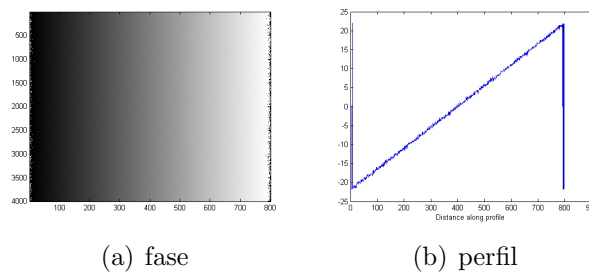


Figura 2.94: 7 veces fase continua 1.

Una vez se obtiene el producto de la fase continua 1 con el valor de $k = 7$ aplicando el operador $U\{\dots\}$, ec. 2.40 se obtiene la fase continua del mapa 7, figura 2.95.

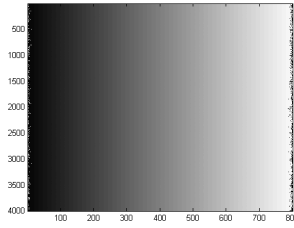


Figura 2.95: Fases continua del mapa 7.

Esta fase continua permitirá calcular la fase continua del siguiente mapa (13 franjas); para el siguiente desenvolvimiento $k = 13/7$, y aplicando el operador $U\{\dots\}$ a la fase discontinua de 13 y la fase continua de 7 (calculada en el paso anterior) se obtiene la fase continua del siguiente mapa, figura 2.96.

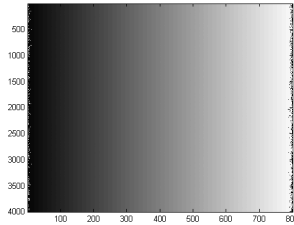


Figura 2.96: Fases continua del mapa 13.

Este proceso se repite hasta obtener la fase continua del ultimo mapa, en este caso 32, la figura 2.97 muestra la fase continua final obtenida implementando el algoritmo generalizado.

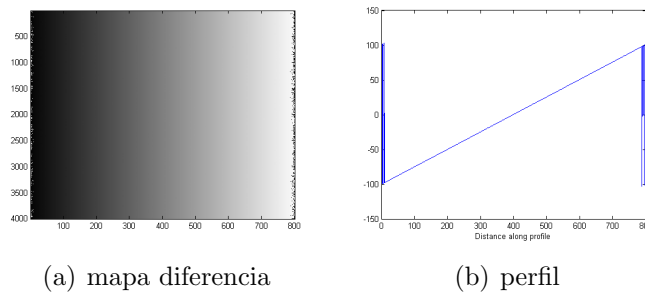


Figura 2.97: Fases discontinuas, algoritmo generalizado.

Para evaluar que tan exitosos fueron los resultados obtenidos aplicando el algoritmo generalizado con ruido gaussiano introducido de $\sigma = 5$ NG, se realiza la gráfica de la tasa de éxito, figura 2.98.

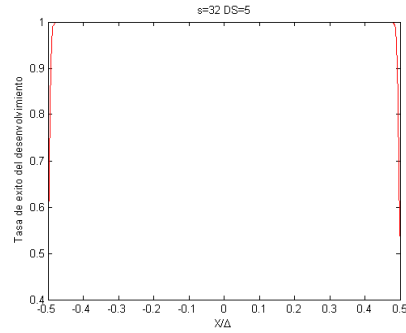


Figura 2.98: Tasa de éxito ruido bajo, algoritmo generalizado.

Se observa nuevamente que la tasa de éxito disminuye en los extremos, ya que estos puntos presentan cambios altos en la fase a medida que varían las franjas, mientras los puntos en el centro son más estables con valores de fase bajos.

Ruido alto

En esta sección se introducirá un ruido de $\sigma = 30$ NG en las franjas simuladas, implementando nuevamente las franjas 1, 7, 13, 24, 32, figura 2.99, y utilizando el algoritmo a cuatro imágenes se obtienen las fases discontinuas, figura 2.100.

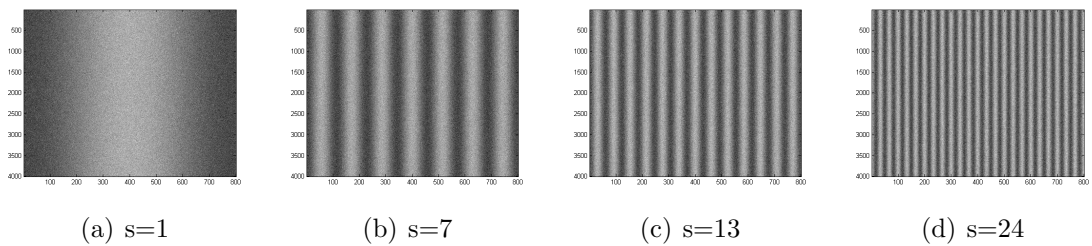


Figura 2.99: Primeros cuatro patrones de franjas generalizado.

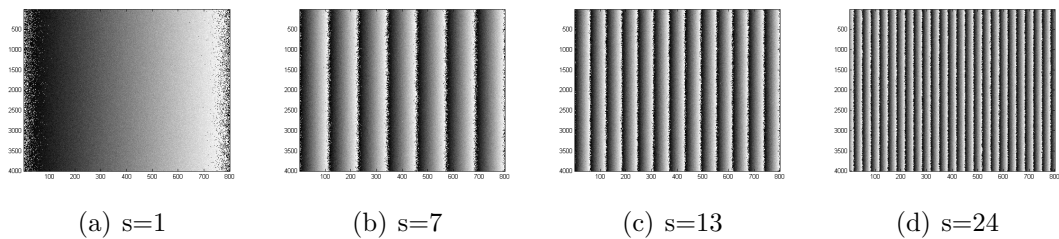


Figura 2.100: Fases discontinuas

Al hacer el producto de k con el mapa 1 se obtiene la siguiente gráfica:

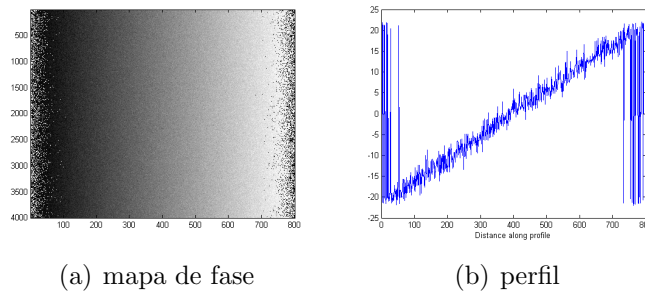


Figura 2.101: Diferencia mapa 7 -1

Una vez se tiene el producto del primer mapa se aplica el operador U el cual genera la fase continua del mapa 7, figura 2.102.

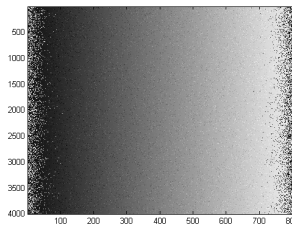


Figura 2.102: Fase continua mapa 7.

Repitiendo el procedimiento descrito anteriormente es posible llegar a la fase continua final del mapa 32, la figura 2.103 muestra la fase continua obtenida y su respectivo perfil.

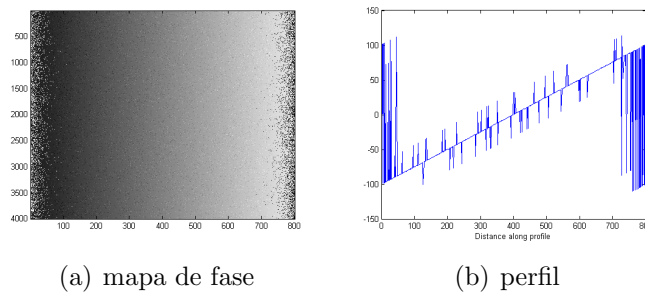


Figura 2.103: Fase continua mapa 32 con algoritmo generalizado.

Se observa que la fase continua que se obtuvo presenta errores altos, por ende se realiza la gráfica de la tasa de éxito en la cual es posible ver que el algoritmo generalizado no presenta tan buenos resultados como en los algoritmos anteriores debido a que la tasa de éxito nunca logra el valor 1, además se observa que no existe estabilidad en los valores a medida que se mueve por el eje x/Δ . Es necesario recordar que estos resultados se pueden mejorar jugando con el valor que toma k .

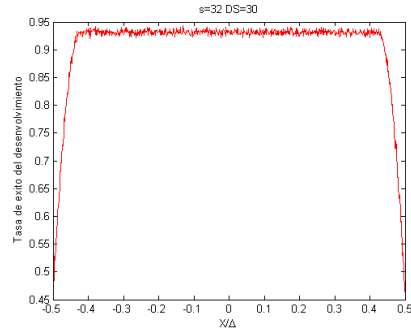


Figura 2.104: Tasa de éxito ruido alto generalizado

2.4.2. simulación de un objeto discontinuo

En esta etapa se evaluará el algoritmo generalizado con objetos discontinuos, donde se introducirá un ruido gaussiano $\sigma = 5$ NG y se implementaran las franjas 1, 7, 13, 24, 32. En primer lugar es necesario generar el patrón de franjas deformadas debido a la topografía del objeto, figura 2.105, e implementando el algoritmo a cuatro imágenes se obtienen los respectivos mapas de fases discontinuos, figura 2.106

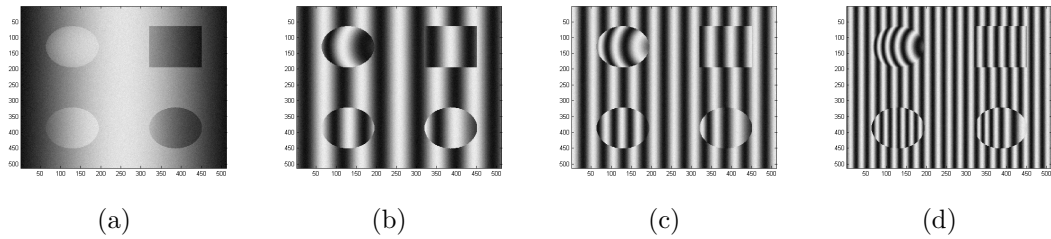


Figura 2.105: Primeras 4 franjas proyectadas exponencial

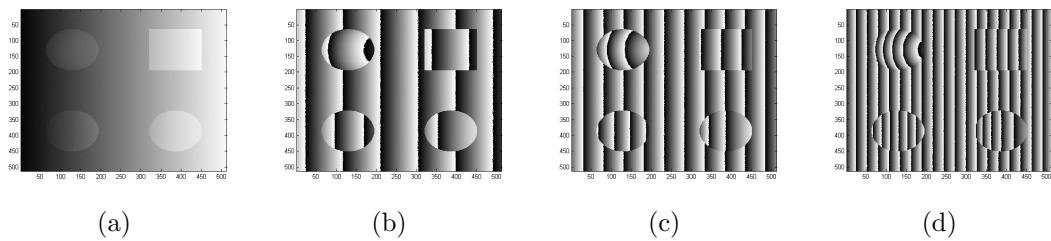


Figura 2.106: Primeras 4 fases discontinuas

A partir de estas fases discontinuas se hace el producto entre k y la fase continua del primer mapa la cual es necesaria para obtener la fase continua final, la figura 2.107 muestra el producto de la fase inicial con el valor k ,

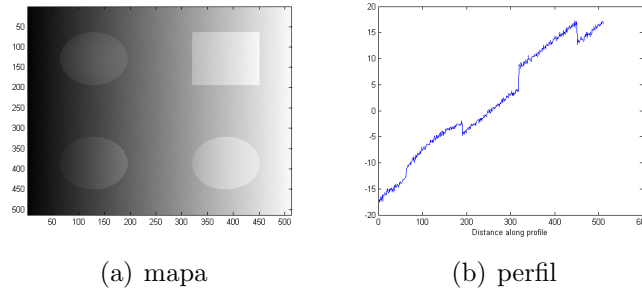


Figura 2.107: Diferencia mapa 7-1 objeto discontinuo

Obtenida la diferencia es posible aplicar el operador U generando así la fase continua del mapa 7, figura 2.108

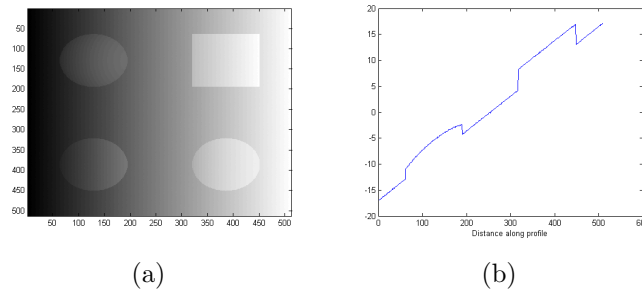


Figura 2.108: Fase continua del mapa 7

finalmente repitiendo este proceso se obtiene la fase continua final de los objetos discontinuos implementando el algoritmo generalizado.

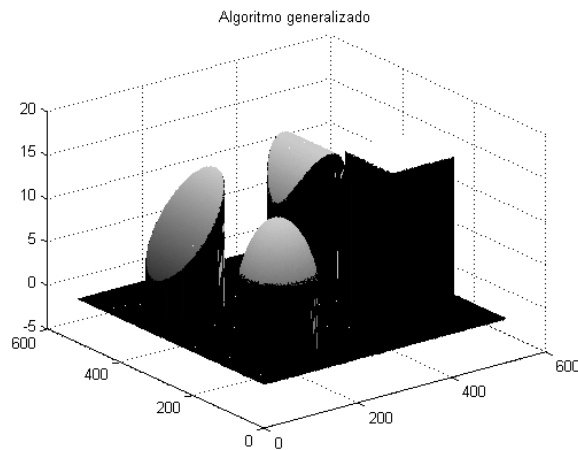


Figura 2.109: Fase continua objeto mapa 32 generalizado

2.4.3. Análisis de resultados

A diferencia de los algoritmos anteriores, en este caso la tasa de éxito no posee valores de 1 para σ altos. La causa debe ser la misma, un valor de "d" mal calculado en la corrección de las fases discontinuas propaga el error en el eje temporal. Pero a diferencia del caso anterior, el operador $U\{\dots\}$ es afectado por el término k que posee valores de 7, 13/7, 24/13 y 32/24. Un valor alto como $k = 6$ amplifica la influencia del error, incrementándose la probabilidad de encontrar diferencias alteradas con valores fuera del rango $-\pi$ a π e incrementado erróneamente el valor "d".

Una variante del algoritmo, bibliográficamente importante fue propuesta por Zhao et al en 1994 [29]. El algoritmo utiliza 2 valores de t , $t = 1$ y $t = s$. Como en el primero se proyecta una franja, $\varphi_d(1) = \varphi_c(1)$, el último tendrá un fase continua estimada de $\varphi_c(s) = s\varphi_c(1)$. De esta manera la ecuación de corrección para este algoritmo sería:

$$\varphi_c(s) = U \{ \varphi_d(s), s\varphi_c(1) \}. \quad (2.42)$$

2.5. ANÁLISIS DEL COMPORTAMIENTO DE LOS ALGORITMOS

En las secciones anteriores se mostró cada uno de los algoritmos, se explicó paso a paso como se lleva a cabo el desenvolvimiento y se empleó para objeto plano y objetos discontinuos. El objetivo de esta sección es comparar los resultado obtenidos en los algoritmos anteriores y a partir de esto poder determinar cual algoritmo sería el más indicado a utilizar dependiendo de las condiciones experimentales. Sabiendo que un algoritmo de corrección de fase adiciona valores enteros de 2π la fase final que se obtiene tendrá únicamente puntos aislados incorrectamente calculados por influencia del ruido. De esta manera, el criterio de comparación se basará en determinar cual o cuales algoritmos son más robustos al ruido, siendo la tasa de éxito la más indicada para este caso.

La figura 2.110 muestra la tasa de éxito vs x/Δ con un error gaussiano de $\sigma = 5$ NG para cada uno de los algoritmos.

En la figura 2.110 **HS** hace referencia al algoritmo lineal, **INV2** algoritmo inverso base 2, **GEN** algoritmo generalizado y **EXP2** exponencial creciente base 2.

Observando los resultados de la figura 2.110, se puede concluir que cuando se tiene ruido bajo, todos los algoritmos se comportan muy similares excepto el lineal que presenta tasa de éxito cero en los extremos, diferente a los demás algoritmos que no alcanzan valores menores a 0.56; esto puede deberse en que al implementar el algoritmo lineal son necesarios más mapas para hacer el desenvolvimiento generando así más posibilidades de detectar un valor entero de discontinuidad 2π erróneo y propagarlo al mapa final.

La figura 2.111 muestra la confiabilidad de los algoritmos temporales cuando el sistema presenta un ruido gaussiano introducido de $\sigma = 30$ NG.

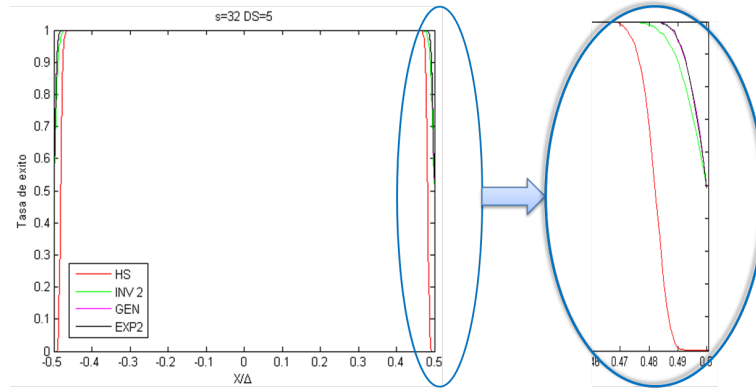


Figura 2.110: Tasa de éxito ruido gaussiano con $\sigma = 5$ NG.

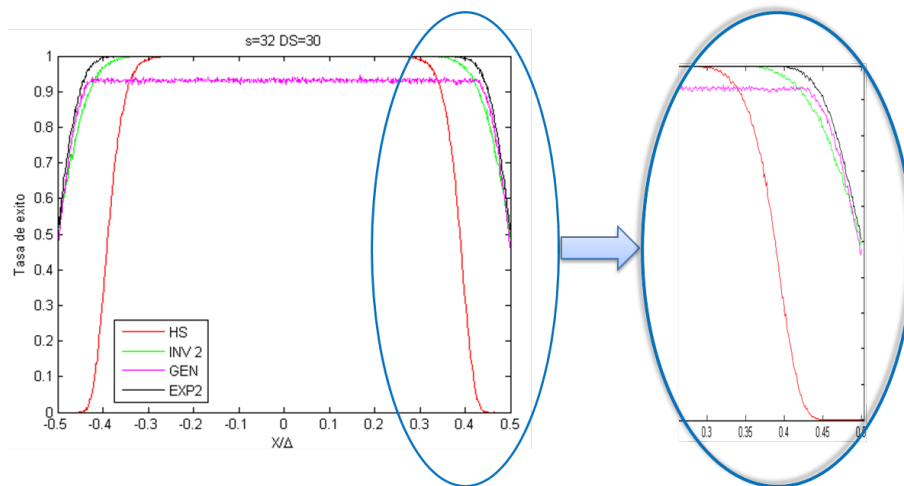


Figura 2.111: Tasa de éxito ruido gaussiano con $\sigma = 30$ NG.

En esta gráfica se puede observar que presenta menor tasa de éxito el algoritmo generalizado, ya que no logra obtener valores de 1 aunque tampoco toma valores de cero, contrario al lineal que nuevamente en los extremos presenta tasa de éxito cero, pero ahora el rango donde es cero es mayor que para el caso de $\sigma = 5$ NG.

El algoritmo exponencial creciente a base 2 nuevamente es el más estable como se muestra en las figuras 2.110, 2.111 (línea negra), a partir de esto se puede concluir que independientemente si el error presente en el sistema es alto o bajo este algoritmo reporta los mejores resultados. Por otra parte el algoritmo generalizado es más sensible al ruido, ya que aunque presenta buenos resultados en presencia de ruido bajo, para ruido alto los resultados obtenidos no son muy buenos, lo que indica que cuando se tiene un sistema con mucho ruido el algoritmo generalizado no es una buena elección (vale aclarar que esto ocurre para los valores de t que se tomaron en la simulación, pero se pueden ajustar los resultados tomando diferentes valores de k).

Como se observó en cada uno de los algoritmos la influencia de ruido en la fase siempre se encuentra en los extremos, que implica una tasa de éxito menor de 1. Como se explicó en las secciones anteriores en los extremos la diferencia de fase discontinua consecutiva es más grande que en el centro, debido a la relación lineal de la fase que pasa por cero en el centro de la zona de proyección. De esta manera en los extremos la probabilidad es mayor que un píxel presente una diferencia de fase que por influencia del ruido el valor de $[\Delta\varphi/2\pi]$ se incrementa en uno aleatoriamente.

En un sistema experimental, la fase que se ubica en el proyector tiene una relación perfectamente lineal y definida por $\varphi = \frac{2\pi}{\Delta}tx$, donde Δ es el campo de proyección en píxeles y t es el número de franjas que se desea proyectar, x es la variable de posición y se ajusta de tal manera que $x = 0$ se ubica en el centro de proyección. Esta expresión se puede escribir como $\varphi = \omega t$, donde $\omega = \frac{2\pi}{\Delta}x$ toma valores en el rango de $[-\pi, +\pi]$. Al ubicar el objeto, en la imagen adquirida por la cámara matemáticamente se encuentra que por influencia de la topografía, la franja se ha desplazado en el plano x, y de la cámara con respecto a la posición de cada franja en el plano de referencia ($Z = 0$). Adicionalmente, el ruido aditivo en la intensidad se ha propagado hasta la fase del sistema de franjas, calculando la fase absoluta del sistema de franjas deformado por el objeto se debe obtener ω de la relación lineal entre φ vs t , para los diferentes sistemas de franjas proyectados. De esta manera, bibliográficamente se ha empleado la tasa de éxito en el cálculo de ω en la relación lineal φ vs t como criterio de comparación entre los algoritmos de desenvolvimiento de fase temporal.

Según lo anterior, la confiabilidad de los algoritmos para el desenvolvimiento temporal de la fase se evaluó numéricamente mediante la creación de secuencias de valores de fase

$$\varphi(t) = \omega t + \varepsilon \quad (2.43)$$

donde $t = 1, 2, 3, \dots, s$, $\varphi(0) = 0$ y ε es una variable aleatoria tomada de una distribución gaussiana con media cero y desviación estándar σ . La ec. 2.43 se define para una posición x en el plano de proyección.

En la figura 2.112 se muestran los valores de fase continua vs t para un valor de $\omega = \pi/2$, que corresponde a $x = \Delta/4$, para los diferentes algoritmos empleados con $s = 32$ número máximo de franjas.

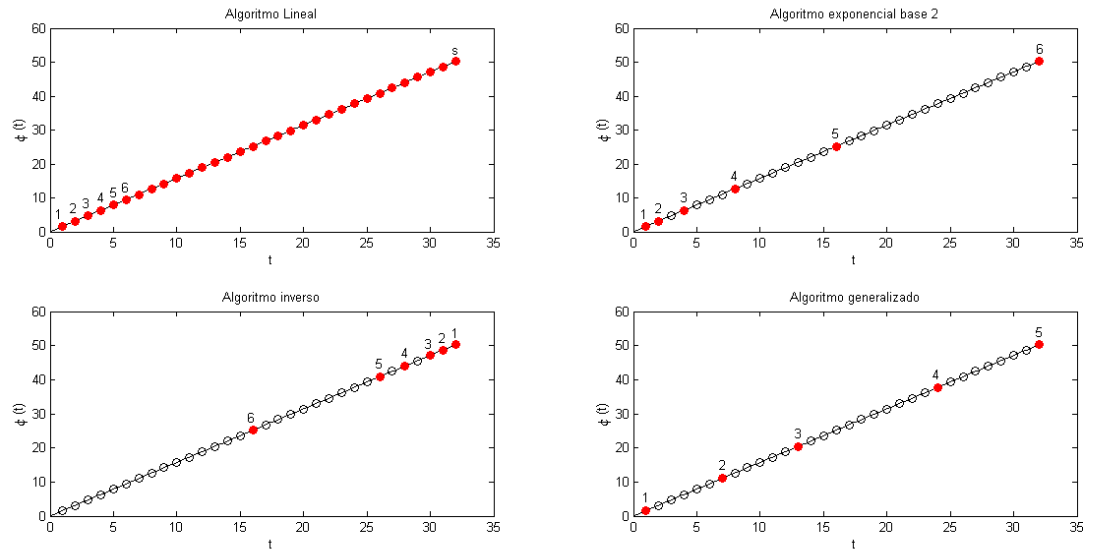


Figura 2.112: Gráficas φ vs t para cada algoritmo

Los círculos rojos en la figura 2.112 representan los valores de fase proyectados, los círculos negros no son medidos y los números encima de los círculos indican el orden en que se adquieren los valores de fase. En la figura se muestra que para el caso del algoritmo lineal todos los mapas de fases están separados a igual distancia. Para el caso exponencial debido a que se inicia en la mínima frecuencia y se va aumentando el número de franjas, los mapas se agrupan más en el extremo inferior. En el algoritmo generalizado se pueden agrupar según como se desee, pero en el algoritmo inverso debido a que comienza en la máxima frecuencia y se van reduciendo el número de franjas, los valores de fase medida se agrupan en valores altos de t . Estas gráficas son importantes por 2 razones: permiten comparar la reducción efectiva del número de franjas adquiridas entre los algoritmos y si se desea calcular ω por mínimos cuadrados de los datos experimentales, con el menor número de franjas presentadas, se requiere de un algoritmo que agrupe valores de fase en valores altos de t ya que todas las curvas pasan por $\varphi = 0$ y $t = 0$.

En la gráfica 2.113 se muestra la tasa de éxito de cada uno de los algoritmos en función de la desviación estándar del ruido gaussiano adicionado a la fase, para los algoritmos de desenvolvimiento de fase temporal. El término **HS** representa el algoritmo lineal, **ZCT**

σ en el rango de 0,15 a 0,3 se pueden implementar: lineal, inverso base 2 y exponencial base 2 y cuando σ esta en el rango de 0.3 a 0.4 se pueden implementar: inverso base 2 y exponencial base 2.

Finalmente cuando $\omega = 0,9\pi$ se observa que todos los algoritmos decaen demasiado rápido, es decir, para obtener buenos resultados en este caso se debe tener una desviación estándar no mayor a 0.1. En conclusión, se observa que los algoritmos exponencial e inverso base 2 poseen un comportamiento estable y robusto frente al ruido para diferentes valores de ω . El algoritmo lineal es muy robusto para $\omega \approx 0$ pero decae fuertemente mientras ω se acerca a π .

En los gráficos anteriores solo se ilustraron los resultados obtenidos con los algoritmos exponencial base 2 e inverso base 2. Surge la duda en determinar el comportamiento frente a ruido adicionado a la fase para diferentes valores de base, por esta razón a continuación se mostrarán los resultados del comportamiento del algoritmo exponencial e inverso cuando se varia la base, y para el caso en que $\omega = 0,5\pi$.

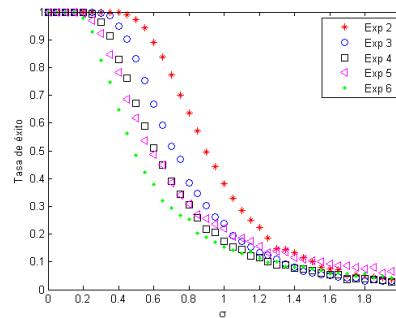


Figura 2.114: Comparación de los algoritmos exponenciales

En la gráfica 2.114, *Exp 2*: representa exponencial base 2, *Exp 3*: exponencial base 3, *Exp 4*: exponencial base 4, *Exp 5*: exponencial base 5 y *Exp 6*: exponencial base 6. Se observa que a medida que la base aumenta, el algoritmo exponencial decae más rápidamente, haciéndose menos resistente al ruido. Se concluye que el exponencial base 2 es el más indicado al momento de implementar un algoritmo exponencial.

En la figura 2.115 *Inv 2*: representa inverso base 2, *Inv 3*: inverso base 3, *Inv 4*: inverso base 4, *Inv 5*: inverso base e *Inv 6*: inverso base 6; se observa que el comportamiento es similar al exponencial ya que a medida que se aumenta el valor de la base el σ en el que la tasa se éxito comienza a decaer es cada vez menor. De igual forma el algoritmo exponencial inverso base 2 es el más indicado al momento de implementar un algoritmo inverso.

Las gráficas mostradas anteriormente son muy útiles ya que a partir de estas se puede seleccionar el algoritmo que presente mejores resultados según el nivel de ruido de las imágenes adquiridas experimentalmente. Estos valores de σ se pueden ajustar utilizando cámaras y vídeo proyectores de alto desempeño y menor ruido adicionado o también se

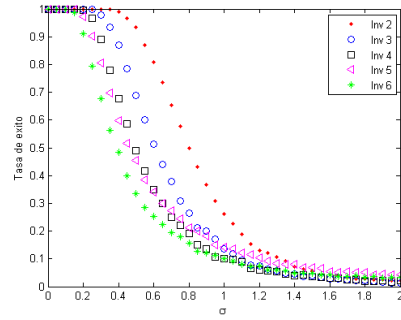


Figura 2.115: Comparación de los algoritmos inversos

puede reducir el nivel de ruido promediando sobre varias imágenes adquiridas. Según lo analizado en esta sección se concluye que un sistema experimental de proyección de franjas con corrección de fase temporal debe construirse con sistemas electro-ópticos de bajo ruido adicionado, usando algoritmos exponencial creciente o inverso base 2 y seleccionando una zona central del campo de proyección para ubicar los objetos de estudio.

3. ANÁLISIS EXPERIMENTAL DE LOS ALGORITMOS PROPUESTOS

En el capítulo anterior se realizó un análisis del ruido presente en los algoritmos a partir de simulaciones y se logró comparar como variaba la tasa de éxito de los diferentes algoritmos respecto a la desviación estándar del ruido que se adicionaba en la fase lineal y a los sistemas de franjas. En este capítulo se realizará un análisis experimental a partir del sistema de proyección de franjas implementado. Se pretende con las bases teóricas definidas, determinar cuál es la mejor estrategia para definir el algoritmo de corrección de fase temporal que permite obtener la reconstrucción tridimensional de un objeto determinado.

3.1. MONTAJE EXPERIMENTAL



Figura 3.1: Configuración experimental del sistema

La figura 3.1 muestra el sistema de proyección de franjas implementado; se empleó un vídeo proyector DELL M210X con brillo de 200 lúmenes y resolución XGA(1024 × 768), una cámara CMOS Edmund Optics uEye E0-2013C con resolución de 1600 píxeles × 1200 píxeles, lente de $f = 8,5$ mm y un computador portátil ASUS S46C core i7 con

6GB de RAM. El plano de referencia (PR) se ubicó a una distancia aproximada de 60 cm, perpendicular a la cámara. El ángulo de proyección fue de aproximadamente 30°. El campo máximo transversal sobre el PR que ocupa todo el campo de observación de la cámara fue de 295mm × 240mm. Empleando *Matlab*[®] versión *R2013b* se construyó una interfaz gráfica que permite la sincronización entre la proyección del sistema de franjas y la adquisición de las imágenes. De igual forma el software implementado permite ajustar los parámetros de la cámara, tales como tiempo de exposición, velocidad de adquisición (fps), submuestreo de imágenes, tipo de imagen cromática a obtener, etc.

Con el fin de incrementar a la máxima velocidad de transferencia de imagen (fps) se ajustó a adquisición monocromática (imagen BMP a 8 bits), submuestreo 2 (un dato adquirido cada 2 píxeles, esto define una resolución final de la imagen de 800×600 píxeles² para el campo de observación) y un tiempo de exposición de 25.07 ms, figura 3.2.

La cámara es muy sensible al tiempo de exposición del sensor: variaciones leves del tiempo de exposición generan variaciones en la calidad de la imagen. Se empleó un tiempo de exposición que generara una imagen estable.

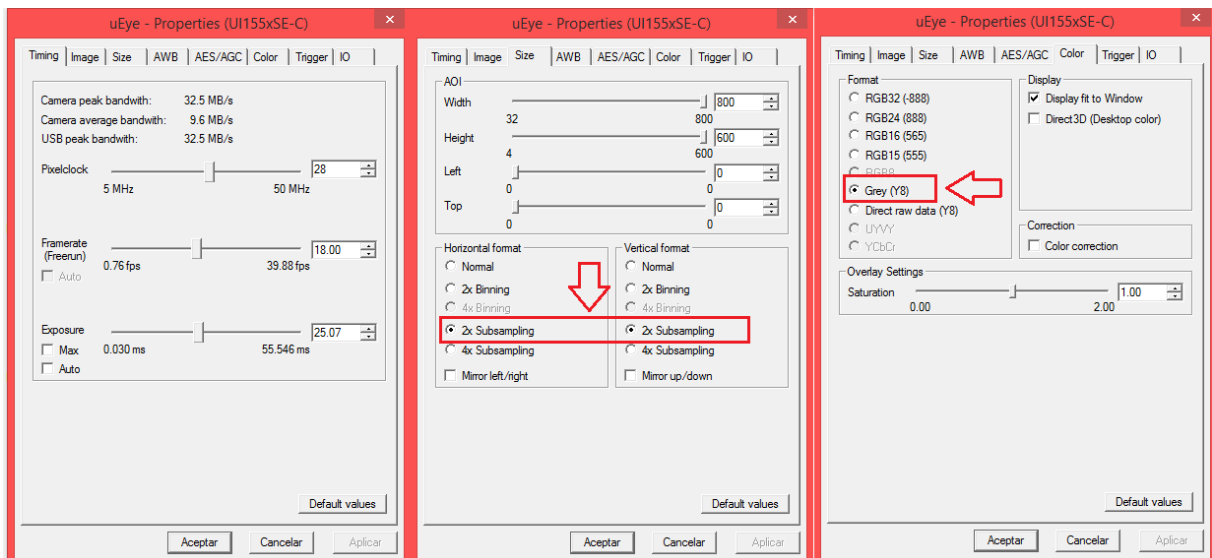


Figura 3.2: Configuración de la CCD.

3.2. ANÁLISIS PRELIMINAR

Según el análisis realizado en el capítulo anterior se concluye que el nivel de ruido afecta el rendimiento de los algoritmos temporales. Altas cantidades de ruido pueden generar un valor erróneo en el cálculo de las 2π discontinuidades "d" y propagaría un error hasta el último sistema de franjas proyectado. Al realizar un análisis detallado de las fuentes de ruido en el montaje implementado se identifican varias fuentes: ruido electrónico en el sistema de adquisición de la cámara, ruido electrónico en el proyector que

genera variaciones aleatorias del campo luminoso de proyección, variaciones aleatorias de iluminación ambiental externa al objeto, etc. El objetivo no consiste en caracterizar cada una de estas fuentes ruidosas; se pretende, siendo prácticos, verificar el nivel de ruido en la fase del sistema de franjas proyectado y su influencia en el proceso de corrección de fase discontinua. La primera etapa del análisis experimental consistió en determinar el nivel de ruido contenido en el diagrama de fase del plano de referencia. El plano de referencia se construyó usando una superficie plana homogénea en color. La figura 3.3 muestra la fase continua obtenida al usar el algoritmo exponencial de base 2 con $s = 64$ franjas en el campo de proyección. Siendo el objeto una superficie plana y debido a la no telecéntrica, se ajustó por mínimos cuadrados un polinomio de orden 4. La figura 3.4 muestra el nivel de ruido al hacer la diferencia entre la fase experimental y la fase interpolada. La desviación estándar corresponde a $\sigma = 0,0261$ rad. El procedimiento experimental de análisis consistió en establecer la manera de incrementar o disminuir σ de la fase final. Según la bibliografía y la experiencia adquirida al interior del GOTS, dos variables que afectan principalmente σ de la fase del sistema de franjas para un montaje definido son: la modulación del sistema de franjas y el tiempo de exposición. Como se analizó anteriormente, el tiempo de exposición no debe ser alterado ya que se perdería estabilidad en la imagen final de la cámara. La modulación de las franjas depende principalmente de: Diafragma de la cámara, textura del objeto y nivel de intensidad luminosa del proyector. Con el fin de variar la modulación de las franjas, se fijó el diafragma de la cámara, se usó una superficie homogénea plana no blanca, y se varió el nivel de intensidad luminosa del proyector.

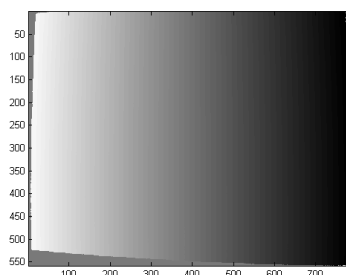


Figura 3.3: fase continua obtenida experimentalmente

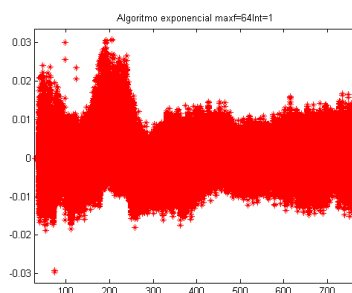


Figura 3.4: Nivel de ruido de la diferencia experimental e interpolada

Se ajustó una escala de intensidad de 0 a 1 con variaciones de 0.1 en el perfil sinusoidal del sistema de franjas proyectado, siendo 1 un valor máximo de 255 y 0 un mínimo de 0 del perfil. La figura 3.5(a) muestra los valores de intensidad empleados en el proyector y la figura 3.5(b) los perfiles de las imágenes adquiridas por la cámara para un paso de $s = 64$. El valor 255 corresponde a la máxima transmitancia de la matriz LCD del vídeo proyector y máxima luminosidad sobre el objeto.

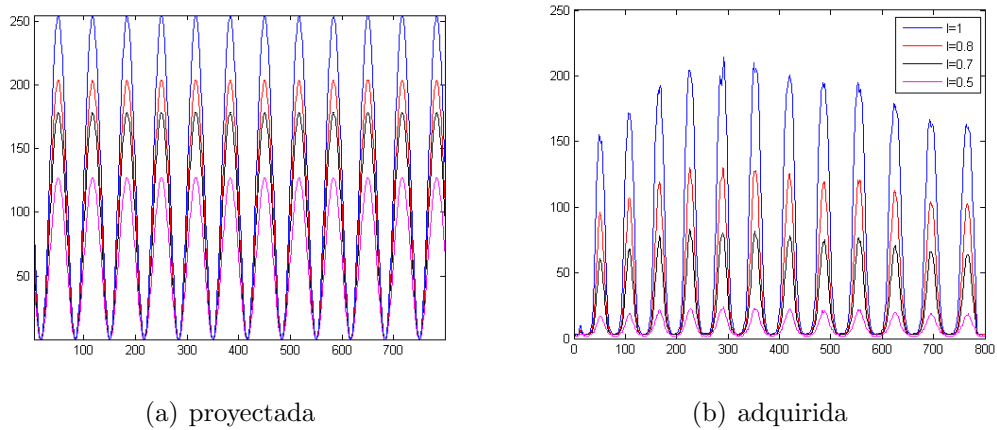


Figura 3.5: Amplitud de la onda según la intensidad

De esta manera la figura 3.4 corresponde a un nivel de proyección $I = 1$. La figura 3.6 muestra el ruido para un nivel en la proyección de $I = 0,7$ donde $\sigma = 0,2413$; variaciones más pequeñas de la modulación final de las franjas pueden obtenerse si se varía el diafragma de la lente de la cámara y se utilizan superficies planas con colores en nivel de gris variables.

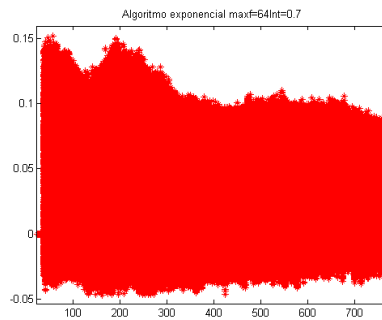


Figura 3.6: Nivel de ruido de $I=0.7$

Experimentalmente para un objeto dado, se ajusta el diafragma de la cámara y se fija un nivel de intensidad luminosa I , de tal manera que no se obtenga una saturación en niveles de gris sobre la imagen final. Se obtendrá un nivel de ruido que afectará el proceso de corrección de fase discontinua. El objetivo consiste en determinar la tasa de éxito en función de σ_ϕ , modulación de las franjas para los diferentes algoritmos. Esta gráfica

servirá para determinar cual algoritmo debe emplearse para una modulación dada de tal manera que la tasa de éxito sea cercana a 1. A continuación se muestran las tasas de éxito para los 4 algoritmos temporales usando una superficie plana homogénea en textura. Para todos los algoritmos el procedimiento consistió en calcular la tasa de éxito del algoritmo variando el nivel de intensidad I y fijando el diafragma de la lente de la cámara. Para todos los casos el sistema de franjas final tenía un número de franjas totales $s = 64$, lo cual implicaba que la fase teórica para calcular el nivel de ruido, correspondía a la figura 3.7 de la fase interpolada con un polinomio de orden 4. La diferencia entre la fase experimental y la fase interpolada permite determinar: 1) el valor de σ , y b) los valores $2\pi d$ para los puntos donde falló el algoritmo.

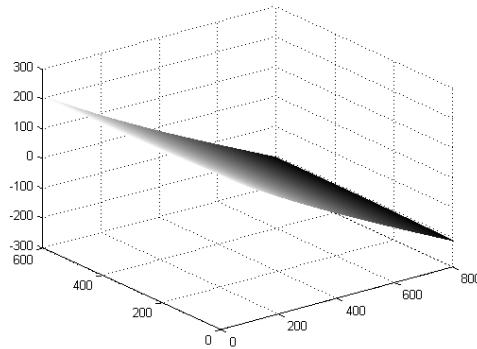


Figura 3.7: Fase interpolada

3.3. ALGORITMO LINEAL

El algoritmo lineal se implementó proyectando 64 patrones de franjas con incremento de 1 entre patrones proyectados. Para cada patrón de franjas se tomaron 4 imágenes con corrimiento de $\pi/2$ y se calculó la fase discontinua. El proceso se repitió para los diferentes valores de nivel de intensidad I . Se tomaron valores de $I = 0,5$ a 1, valores inferiores de 0.5 no muestreaban correctamente el sistema de franjas, y se perdía el concepto de fase. Para cada fase final obtenida se calculó el nivel de ruido. La figura 3.8 muestra los errores para los diferentes valores de nivel de intensidad I . Las figuras 3.8(a) y 3.8(b) muestra los valores $2\pi d$ donde se cometieron errores en el procedimiento de la corrección de la fase. De esta figura se calcula la tasa de éxito. La figura 3.9 muestra la variación de la tasa de éxito tomando como variable horizontal modulación, σ , SNR^1 , donde la modulación se define como [13]:

$$M = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}. \quad (3.1)$$

y SNR se define: $SNR = \frac{M}{\sigma}$.

¹signal-to-noise-ratio

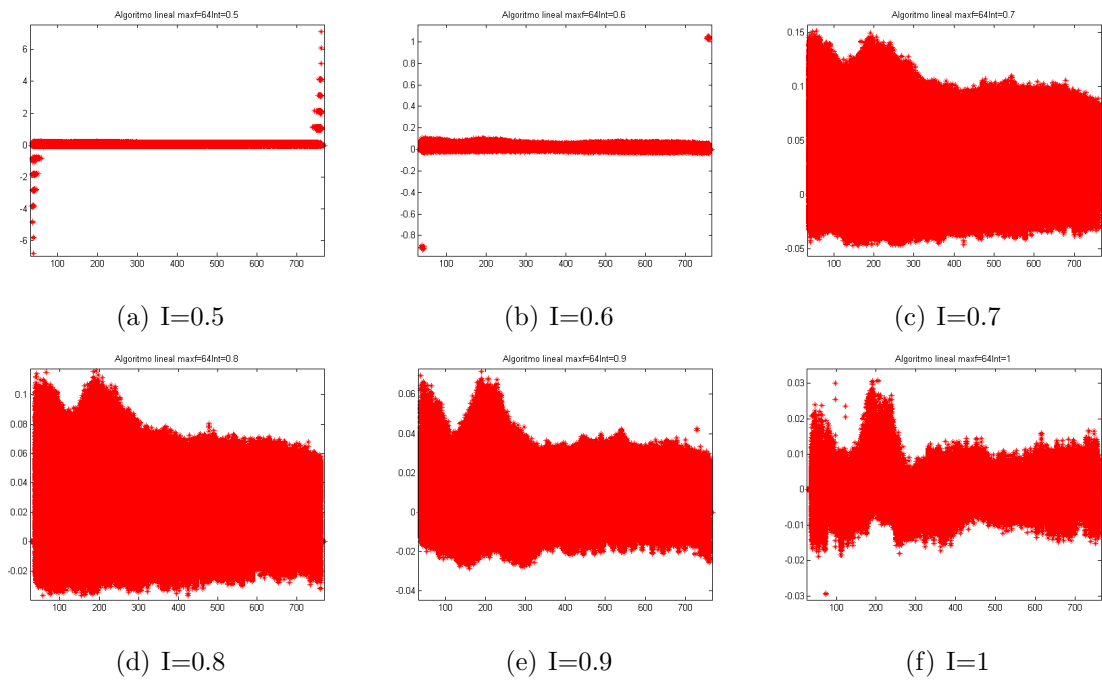


Figura 3.8: Errores experimentales del algoritmo lineal.

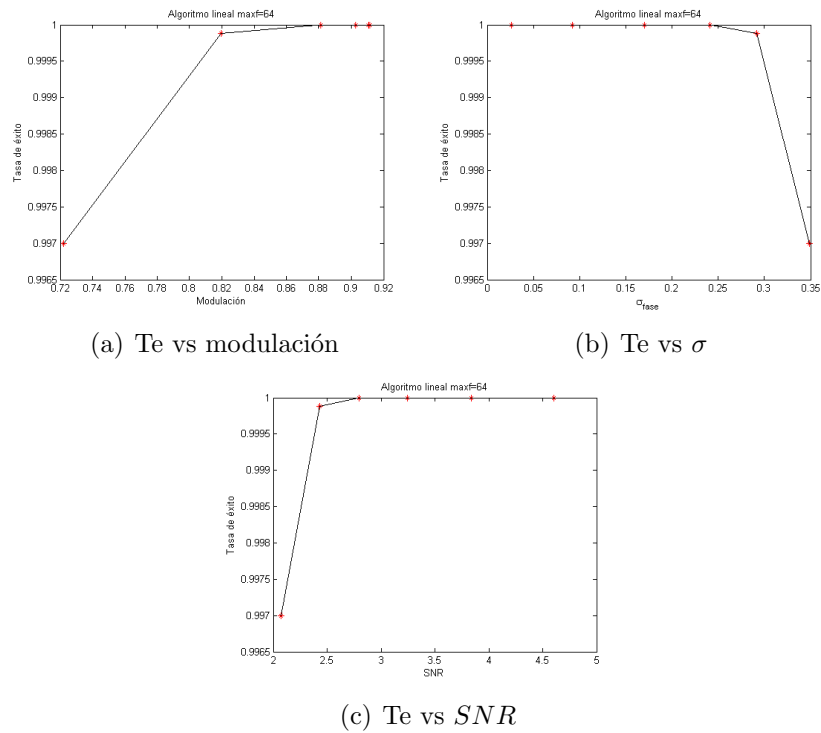


Figura 3.9: Gráficas para el análisis del error del algoritmo lineal

La figura 3.10 muestra la gráfica de σ vs modulación. De esta gráfica se concluye que al disminuir la modulación de las franjas, variando el nivel de intensidad luminosa del proyector, se incrementa el nivel de ruido en la fase final del sistema de franjas. De igual forma al incrementarse el nivel de intensidad se reduce la tasa de éxito.

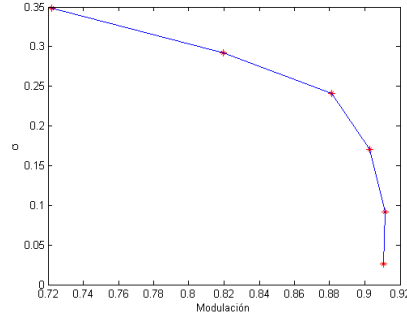


Figura 3.10: Desviación estándar versus modulación.

3.4. ALGORITMO EXPONENCIAL BASE 2

En el algoritmo exponencial base 2 nuevamente se tomó como número máximo de franjas $s=64$ tomando las franjas 1, 2, 4, 8, 16, 32, 64; para cada patrón de franjas se tomaron 4 imágenes con corrimiento $\pi/2$ y se calculó la fase discontinua. Para cada fase final obtenida se calculó el nivel de ruido, figura 3.11.

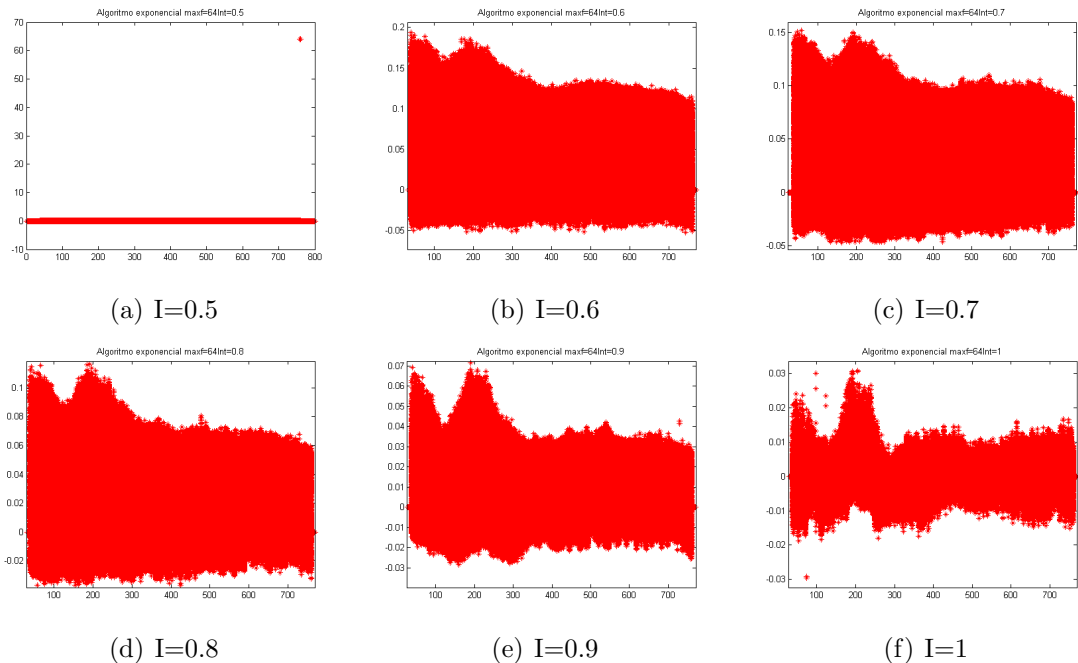


Figura 3.11: Errores experimentales del algoritmo exponencial base 2.

Con los parámetros obtenidos de desviación estándar, modulación y relación señal a ruido es posible construir nuevamente las gráficas de tasa de éxito las cuales presentan comportamiento similar al lineal, a medida que la modulación aumenta la tasa de éxito también lo hace, cuando la desviación estándar aumenta la tasa de éxito disminuye comportamiento que era esperado.

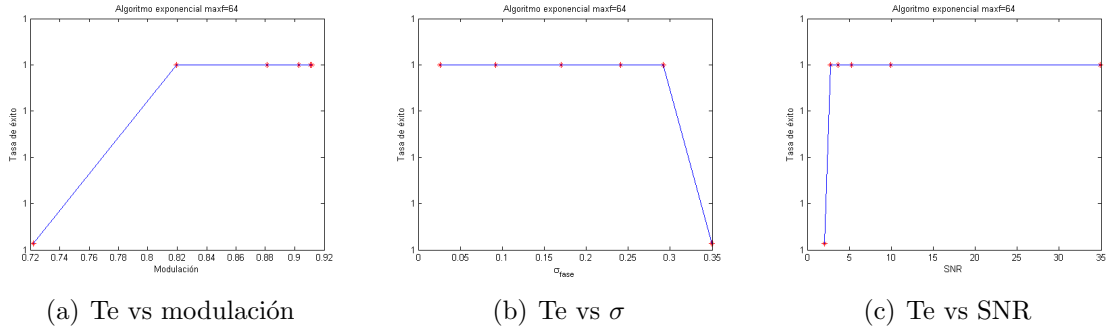


Figura 3.12: Gráficas para el análisis del error del algoritmo exponencial

3.5. ALGORITMO INVERSO BASE 2

En el algoritmo inverso base 2 se tomaron las franjas 64, 63, 62, 60, 56, 48, 32, 0; para cada patrón de franjas se tomaron 4 imágenes con corrimiento $\pi/2$ y se cálculo la fase discontinua. Para cada fase final obtenida se cálculo el nivel de ruido, figura 3.13.

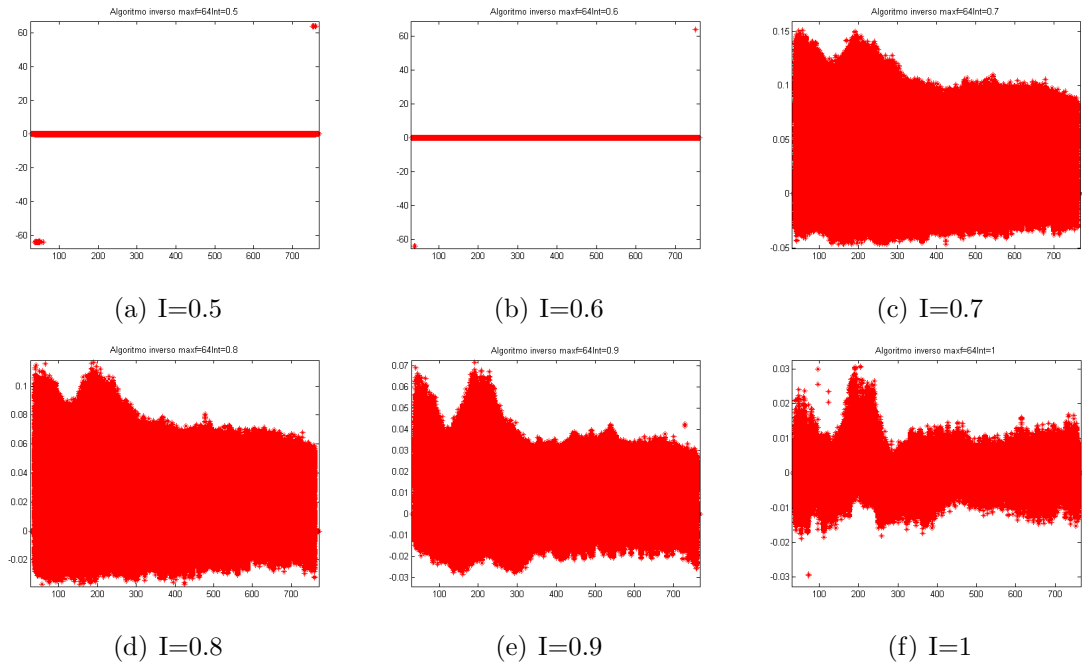


Figura 3.13: Errores experimentales del algoritmo inverso base 2.

La figura 3.14 muestra la variación de la tasa de éxito tomando como variable horizontal modulación, desviación estándar y relación señal a ruido (SNR).

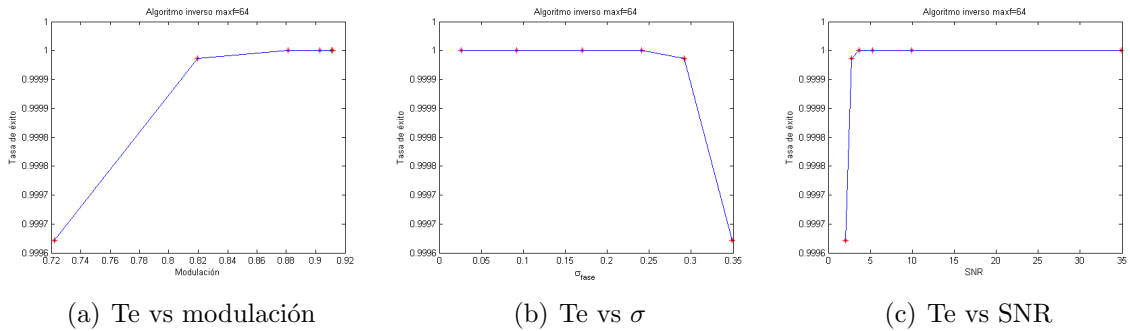


Figura 3.14: Gráficas para el análisis del error del algoritmo inverso

3.6. ALGORITMO GENERALIZADO

Finalmente se realizó el análisis para el algoritmo generalizado, debido a que en este se pueden tomar valores aleatorios hasta llegar a la fase final que se desea, en este caso $s = 64$ para realizar el proceso experimental se tomaron los patrones 1,12,33,64, los cuales arrojaron los siguientes errores en la fase para cada valor de intensidad.

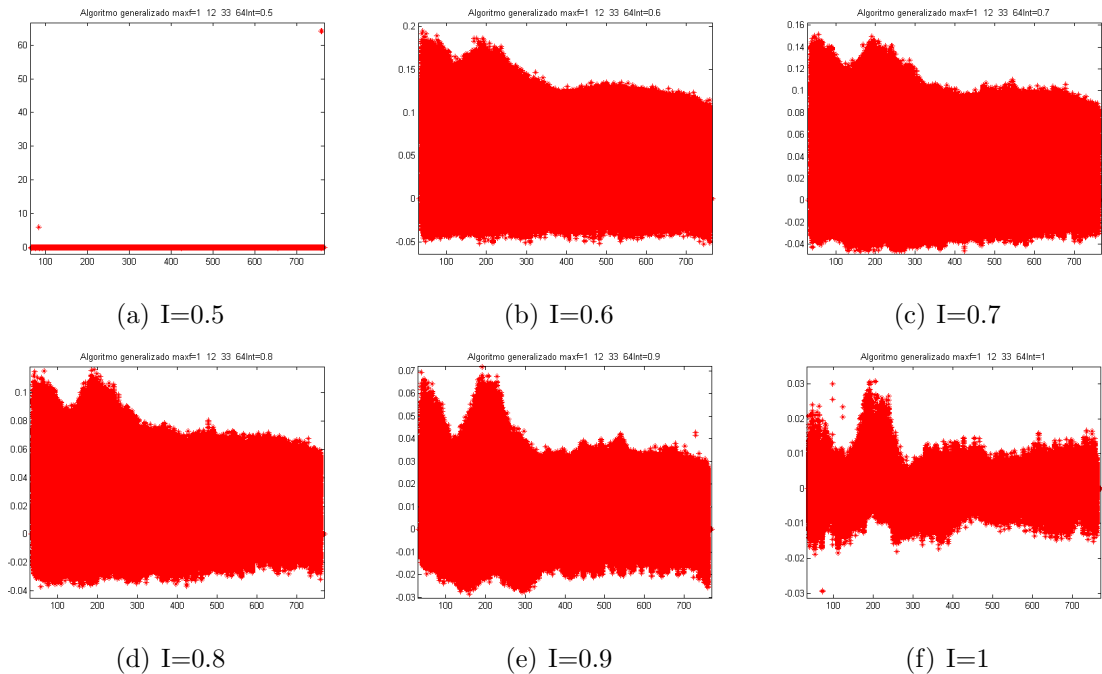


Figura 3.15: Errores experimentales del algoritmo generalizado.

Nuevamente se realizan las gráficas de tasa de éxito en función de la desviación estándar, modulación y relación señal a ruido, ya que estas son las que permiten determinar el mejor algoritmo.

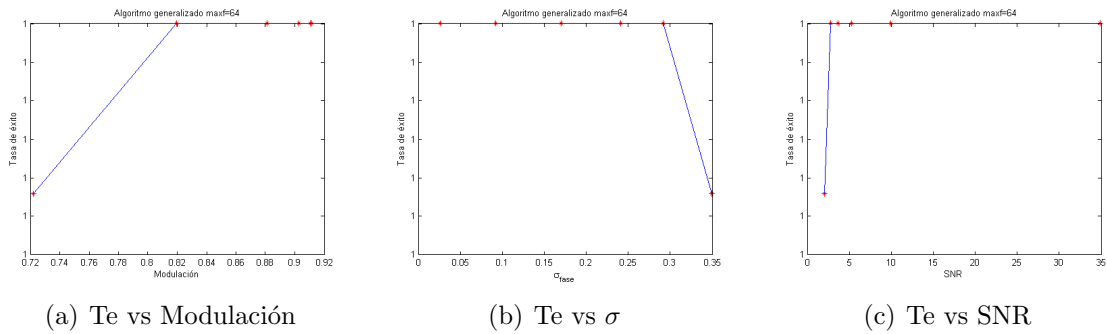


Figura 3.16: Gráficas del para el análisis del error del algoritmo generalizado.

3.7. ANÁLISIS DE RESULTADOS

Con las gráficas mostradas anteriormente para cada uno de los algoritmos es posible determinar cual sería el más indicado implementar en el sistema del laboratorio y que condiciones deben cumplirse para que este funcione en función del objeto a reconstruir. Por esta razón se presentaran las gráficas de cada uno de los algoritmos en una sola, ya que permitirá comparar más fácilmente el comportamiento de los mismos.

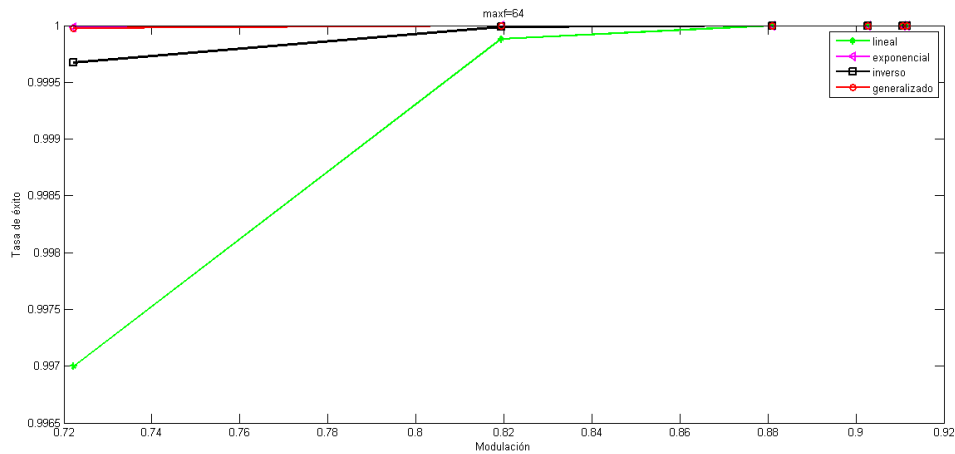


Figura 3.17: Tasa de éxito vs modulación de todos los algoritmos.

La figura 3.17 muestra la tasa de éxito vs la modulación donde se observa que en todos los algoritmos a medida en que la modulación aumenta tasa de éxito es mayor; Según esta gráfica se puede concluir que si se utiliza una modulación mayor o igual de 0.82 cualquiera de los algoritmos puede ser implementado ya que todos presentan tasa de

éxito de 1. Se observa que el algoritmo lineal es el que decae más rápidamente seguido del inverso mientras el generalizado y el exponencial son los mas estables.

En la gráfica 3.18 se observa que la tasa de éxito disminuye a medida que la desviación estándar del sistema aumenta, comportamiento esperado a partir de las simulaciones realizadas en el capítulo anterior. Además se observa que el algoritmo lineal nuevamente decae mas rápidamente, esto ya nos da una luz de que el algoritmo lineal no es una buena alternativa ya que no es muy estable.

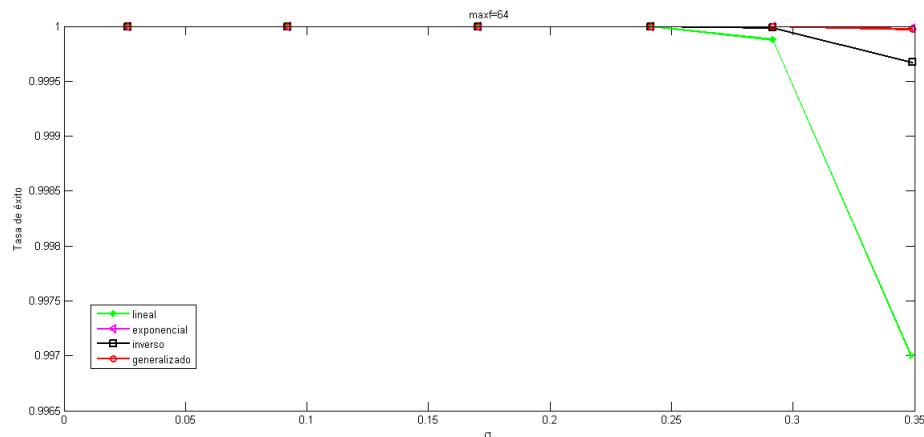


Figura 3.18: Tasa de éxito vs σ de todos los algoritmos.

Finalmente se muestran las gráficas de tasa de éxito versus la relación señal a ruido (SNR), esta última se define como el margen que hay entre la modulación de las franjas adquiridas y el ruido que la corrompe. Donde se observa que a medida que SNR aumenta la tasa de éxito también lo hace, esto se debe a que la desviación estándar se hace cada vez mas pequeña. Analizando las gráficas se observa que nuevamente el algoritmo lineal decae más que los demás algoritmos, es decir que este es el más sensible al ruido.

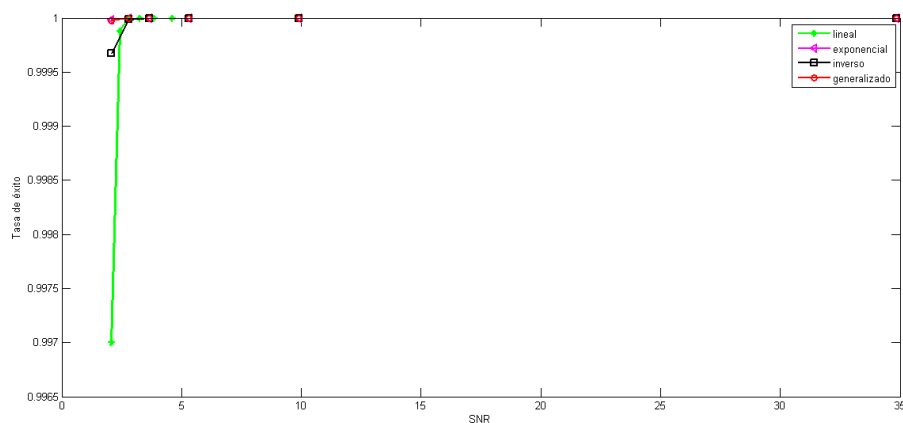


Figura 3.19: Tasa de éxito vs SNR de todos los algoritmos.

A partir de estas tres últimas gráficas se puede concluir que el algoritmo lineal es el más sensible al ruido, ya que como son necesarios tantos mapas para realizar el desenvolvimiento existe mayor probabilidad de que el error se propague, mientras el exponencial es el algoritmo más robusto al ruido. Estos dos resultados coinciden exactamente con lo obtenido en las simulaciones anteriores, el algoritmo generalizado presenta variaciones respecto a los resultados de la simulación, la parte experimental reporta tasa de éxito muy estable en cada uno de los diferentes parámetros mientras en la simulación los resultados no eran tan buenos, esto se debe a que el buen desenvolvimiento en el algoritmo depende de los valores de k que se tomen, como se explicó en el capítulo anterior.

3.8. SELECCIÓN DEL ALGORITMO TEMPORAL EN FUNCIÓN DEL OBJETO

Las figuras anteriores establecen un procedimiento que permite definir el algoritmo temporal en función del objeto. Es decir, si el objeto al evaluarle la modulación posee un rango de valores mayores de 0.86, cualquier algoritmo sirve para corregir la fase discontinua. Sin embargo valores entre 0.79 a 0.86 indica que los métodos lineal e inverso no son apropiados.

Para ilustrar el proceso experimental de selección del algoritmo temporal de corrección de fase se emplean dos objetos mostrados en la figura 3.20. En la figura 3.21 se muestra la imagen del sistema de franjas para los objetos de la figura 3.20, donde se observa que debido a la topografía de los mismos estas presentan deformación.

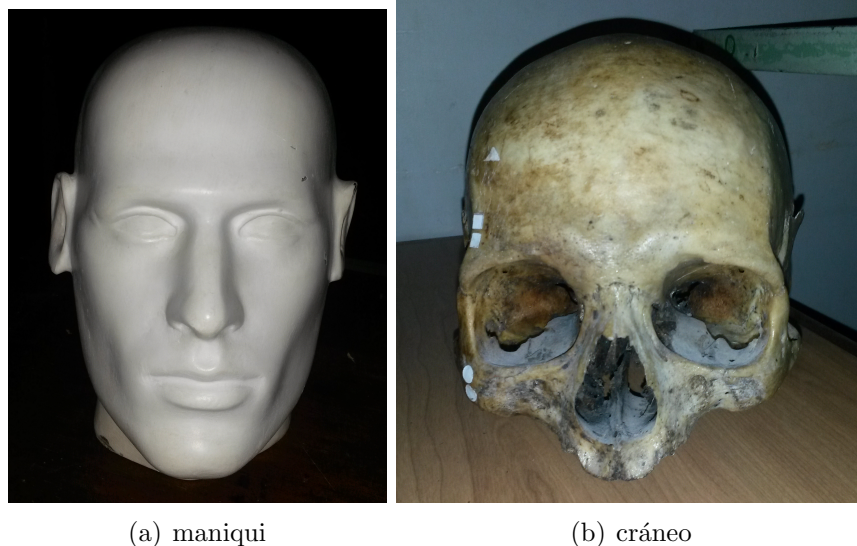


Figura 3.20: Objetos utilizados para la obtención de la fase

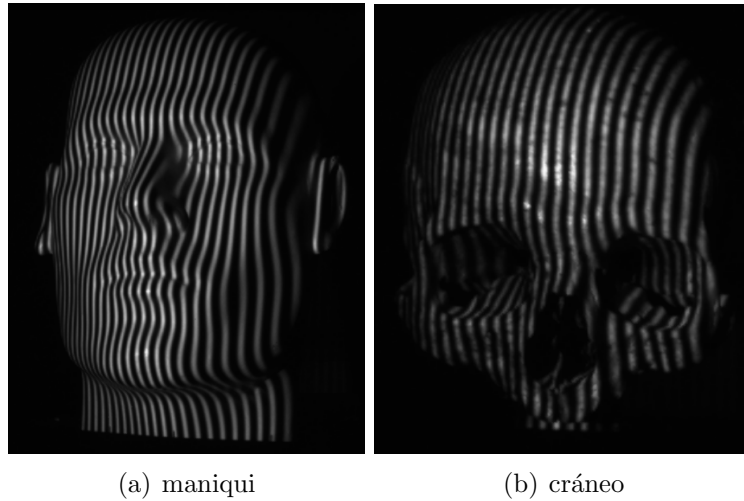


Figura 3.21: Franjas sobre los objetos de estudio

Del algoritmo de corrimiento de fase se puede calcular la amplitud local de la intensidad de las franjas, B en la ecuación 1.8:

$$B = \sqrt{(I_1 - I_3)^2 + (I_2 - I_4)^2} \quad (3.2)$$

Valores cercanos a cero indican que no hay franjas presentes en ese punto de la imagen. De esta manera, se puede obtener una mascara que valida puntos del objeto en la imagen. La figura 3.22 muestra las mascaras obtenidas al binarizar las distribuciones de amplitud local de la intensidad de franjas.

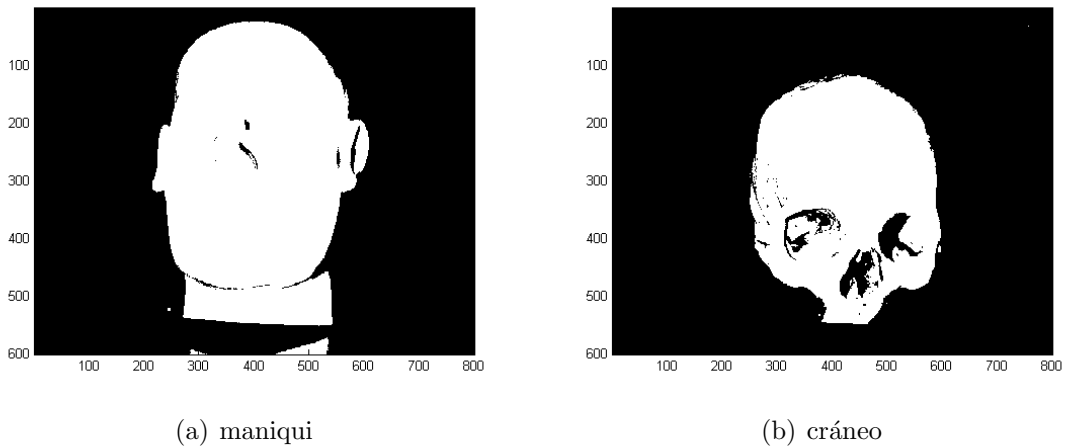


Figura 3.22: Mascara de los objetos en estudio

Usando un proceso de convolución con una ventana de ancho de 15 píxeles, se cálculo la modulación local de las franjas usando las mascaras y determinando valores locales de máximos y mínimos de intensidad. La figura 3.23 muestra los valores de modulación para los objetos usados.

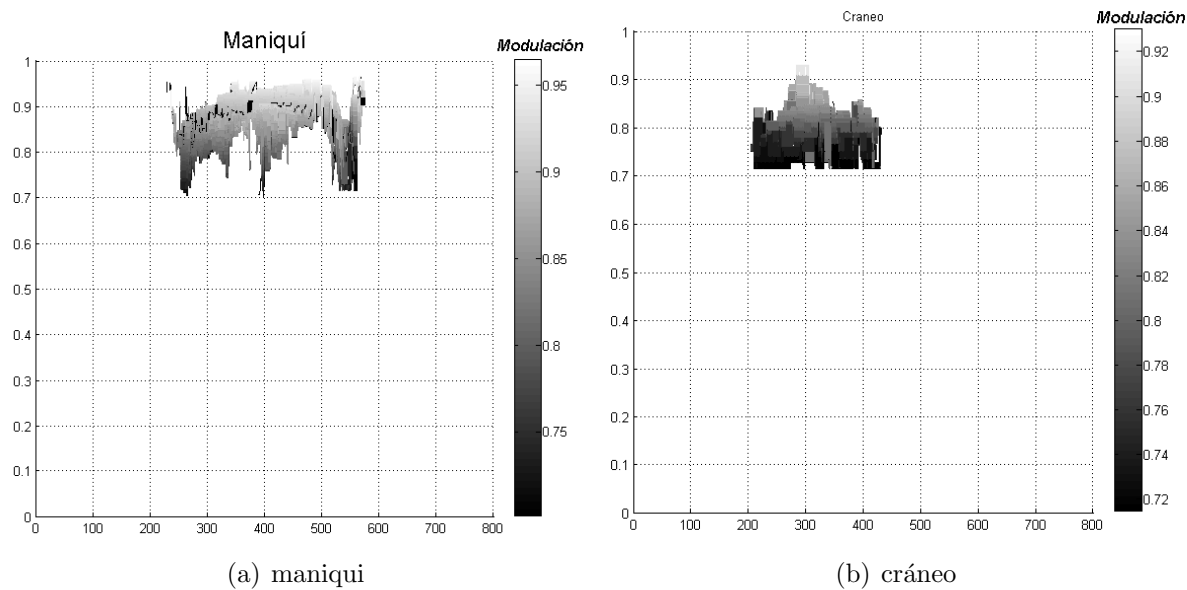


Figura 3.23: Modulaciones de los objetos en estudio

De esta manera se puede determinar que para el objeto 1: maniquí, el rango de valores centrales de la modulación oscila alrededor de 0.85 y para el objeto 2: cráneo, el rango oscila alrededor de 0.78. Esto implica que para el objeto 1 se puede implementar cualquier algoritmo, mientras que para el objeto 2 los algoritmos inverso y lineal presentan errores en la corrección de la fase.

La figura 3.24 muestra la fase continua para el objeto 1 empleando el algoritmo lineal y la figura 3.25 la fase continua del objeto 2 empleando el algoritmo exponencial base 2.

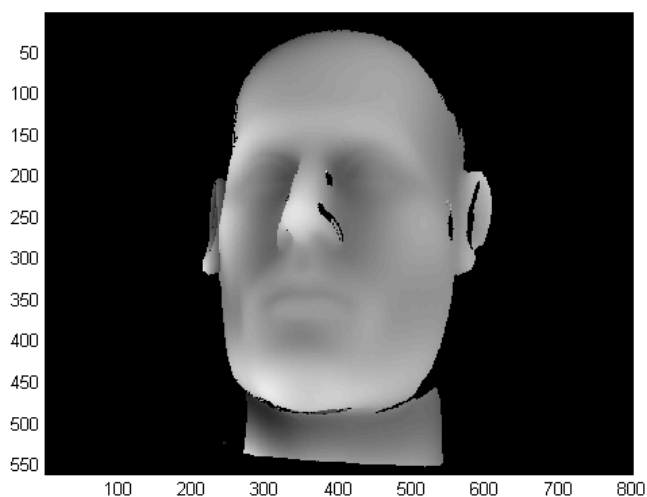


Figura 3.24: Fase continua del maniquí con algoritmo lineal

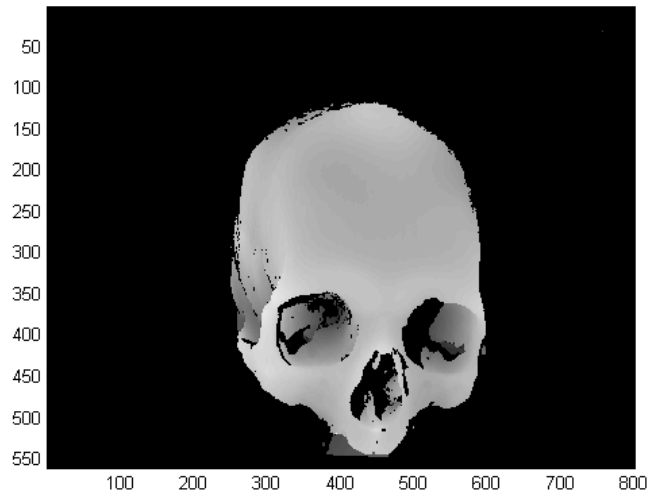


Figura 3.25: Fase continua del cráneo con algoritmo exponencial

En las imágenes anteriores se observa que a pesar de la topografía compleja que presentan los objetos fue posible obtener las fases continuas de ellos de manera automática, lo cual implica que los algoritmos de tipo temporal fueron una buena elección. Al observar los mesh de las figuras 3.24 y 3.25 se aprecia que existen algunos picos, estos pueden ser debido a que el algoritmo calculó valores de las 2π discontinuidades "d" de manera incorrecta.

4. *CONCLUSIONES*

Se demostró que las ventajas que presenta el algoritmo de desenvolvimiento temporal son: la fase de un mapa de franjas es corregida a partir de la información de mapas anteriores evitando la propagación espacial de errores, además permite obtener la fase absoluta del sistema y facilita un proceso automático de reconstrucción tridimensional.

Procesos de simulación y montaje experimental demostraron que el nivel de ruido en el sistema de franjas afecta el comportamiento de los algoritmos; mostrando que el algoritmo exponencial a base 2 presenta el mejor desempeño, usando como criterio de comparación la tasa de éxito.

En todos los algoritmos de corrección de fase temporal se utiliza el redondeo al entero más cercano para calcular las 2π discontinuidades, pero esta función es afectada por el nivel de ruido de la imagen. La probabilidad de error se incrementa al incrementarse el número de franjas proyectadas y al utilizar un factor multiplicativo alto en el proceso de corrección de la fase.

En todos los algoritmos de desenvolvimiento temporal se observó que los puntos con fase mal calculada se encuentran en los extremos del campo de proyección. Esto es debido al hecho de que las diferencias de fase entre dos patrones proyectados es más grande y cercana a π en los extremos que en el centro de proyección.

Con el desarrollo del presente trabajo de grado se demostró de manera cuantitativa que la modulación local de las franjas afecta el nivel de ruido de la fase obteniéndose reconstrucciones tridimensionales ruidosas y mayor probabilidad de error en el algoritmo de corrección de fase. Por lo tanto, antes de realizar la reconstrucción tridimensional se debe establecer un proceso preliminar para evaluar la modulación de las franjas y definir el algoritmo de corrección apropiado para el montaje experimental utilizado.

Bibliografía

- [1] ARCINIEGAS, J. Exploración superficial de grietas y deformaciones en tubería flexible con refuerzo no metálico para el transporte de hidrocarburos a partir de la metodología de proyección de franjas. Tesis de maestría, Universidad Industrial de Santander, 2014.
- [2] BING, P., HUI-MIN, X., BO-QIN, X., AND FU-LONG, D. Performance of sub-pixel registration algorithms in digital image correlation. *Measurement Science and Technology* 17, 6 (2006), 1615.
- [3] CASTILLO, D. A., AND DURÁN, E. V. Extracción de las características topográficas de la piel humana por medio del método de proyección de franjas. Tesis de pregrado, Universidad Industrial de Santander, 2012.
- [4] CHEN, L., QUAN, C., TAY, C. J., AND FU, Y. Shape measurement using one frame projected sawtooth fringe pattern. *Optics communications* 246, 4 (2005), 275–284.
- [5] COGGRAVE, C. R. *Temporal Phase Unwrapping: Development and Application of Real-Time Systems for Surface Profile and Surface Displacement Measurement*. PhD thesis, Loughborough University, 2001.
- [6] CONTRERAS, C. R. *Análisis tridimensional de objetos a 360° de observación y campo amplio con multiple projector-cámara a partir de la iluminación estructurada*. Tesis doctoral, Universidad Industrial de Santander, 8 2014.
- [7] FAVARO, P., AND SOATTO, S. A geometric approach to shape from defocus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, 3 (2005), 406–417.
- [8] FU, Y., WANG, W., AND XIAO, H. Three-dimensional profile measurement based on modified temporal phase unwrapping algorithm. *Optik-International Journal for Light and Electron Optics* 124, 6 (2013), 557–560.
- [9] GASVIK, K. J. *Optical Metrology*, 3 ed. John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2002.
- [10] GÓMEZ, A., AND RODRÍGUEZ, O. Diseño y reconstrucción de un modelo funcional para la obtención de información topográfica de números de identificación vehicular. Tesis de pregrado, Universidad Industrial de Santander, 2012.

- [11] GONZÁLEZ, A. L. Reconstrucción 3D del torso humano a 360° de observación mediante la técnica de proyección de franjas. Tesis de maestría, Universidad Autónoma de Bucaramanga, 2014.
- [12] GORTHI, S. S., AND RASTOGI, P. Fringe projection techniques: whither we are? *Optics and lasers in engineering* 48, IMAC-REVIEW-2009-001 (2010), 133–140.
- [13] HECHT, E., AND ZAJAC, A. *Optics Addison-Wesley*, 4 ed. Adisson Wesley, 2002.
- [14] HUANG, P. S., AND ZHANG, S. Fast three-step phase-shifting algorithm. *Applied optics* 45, 21 (2006), 5086–5091.
- [15] HUANG, P. S., ZHANG, S., AND CHIANG, F.-P. Trapezoidal phase-shifting method for three-dimensional shape measurement. *Optical Engineering* 44, 12 (2005), 123601–123601.
- [16] HUNTLEY, J. M., AND SALDNER, H. Temporal phase-unwrapping algorithm for automated interferogram analysis. *Applied Optics* 32, 17 (1993), 3047–3052.
- [17] HUNTLEY, J. M., AND SALDNER, H. O. Error-reduction methods for shape measurement by temporal phase unwrapping. *JOSA A* 14, 12 (1997), 3188–3196.
- [18] HUNTLEY, J. M., AND SALDNER, H. O. Shape measurement by temporal phase unwrapping: comparison of unwrapping algorithms. *Measurement Science and Technology* 8, 9 (1997), 986.
- [19] JIA, P., KOFMAN, J., AND CHAD, E. Two-step triangular-pattern phase-shifting method for three-dimensional object-shape measurement. *Optical Engineering* 46, 8 (2007), 083201–083201.
- [20] ROJAS, L. J. Exploración topográfica del rostro humano a multiples campos con registro unificado tridimensional. Tesis de pregrado, Universidad Industrial de Santander, 2012.
- [21] SALVI, J., PAGES, J., AND BATLLE, J. Pattern codification strategies in structured light systems. *Pattern Recognition* 37, 4 (2004), 827–849.
- [22] SCHECHNER, Y. Y., AND KIRYATI, N. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision* 39, 2 (2000), 141–162.
- [23] SU, X., AND CHEN, W. Fourier transform profilometry:: a review. *Optics and lasers in Engineering* 35, 5 (2001), 263–284.
- [24] SU, X.-Y., ZHOU, W.-S., VON BALLY, G., AND VUKICEVIC, D. Automated phase-measuring profilometry using defocused projection of a ronchi grating. *Optics Communications* 94, 6 (1992), 561–573.
- [25] SUBBARAO, M., AND SURYA, G. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision* 13, 3 (1994), 271–294.

- [26] TIAN, J., PENG, X., AND ZHAO, X. A generalized temporal phase unwrapping algorithm for three-dimensional profilometry. *Optics and Lasers in Engineering* 46, 4 (2008), 336–342.
- [27] ZHANG, S., VAN DER WEIDE, D., AND OLIVER, J. Superfast phase-shifting method for 3-d shape measurement. *Optics Express* 18, 9 (2010), 9684–9689.
- [28] ZHANG, Z., ZHANG, D., AND PENG, X. Performance analysis of a 3d full-field sensor based on fringe projection. *Optics and lasers in engineering* 42, 3 (2004), 341–353.
- [29] ZHAO, H., CHEN, W., AND TAN, Y. Phase-unwrapping algorithm for the measurement of three-dimensional object shapes. *Applied Optics* 33, 20 (1994), 4497–4500.