

**ANÁLISIS PRELIMINAR DE UN SISTEMA DE RECONSTRUCCIÓN
TRIDIMENSIONAL MULTI-CAPTOR: APLICACIONES MÉDICAS E
INDUSTRIALES**

**MANUEL JULIÁN BARRERA AMOROCHO
GONZALO AUGUSTO TORRES QUINTERO**

**ESCUELA DE INGENIERÍA MECÁNICA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA**

2011

**ANÁLISIS PRELIMINAR DE UN SISTEMA DE RECONSTRUCCIÓN
TRIDIMENSIONAL MULTI-CAPTOR: APLICACIONES MÉDICAS E
INDUSTRIALES**

**MANUEL JULIÁN BARRERA AMOROCHO
GONZALO AUGUSTO TORRES QUINTERO**

**Trabajo de grado para optar al título de
Ingeniero mecánico**

Director

**CARLOS RICARDO CONTRERAS PICO
Candidato a Doctor, Físico**

Codirector:

**OMAR ARMANDO GELVEZ AROCHA
Ingeniero Mecánico**

**ESCUELA DE INGENIERÍA MECÁNICA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA**

2011

AGRADECIMIENTOS

A Carlos Ricardo Contreras, por su apoyo y colaboración durante el desarrollo de esta investigación.

Al doctor Jaime Meneses, por el apoyo brindado y el tiempo dedicado.

CONTENIDO

	Pág.
INTRODUCCIÓN	18
1. ANTECEDENTES	20
1.1 RECONSTRUCCIÓN TRIDIMENSIONAL	20
1.1.1 Conceptos generales de la reconstrucción tridimensional	20
1.1.2 Clasificación de los métodos ópticos	22
1.2 RECONSTRUCCIÓN 3D POR MEDIO DE PROYECCIÓN DE FRANJAS	28
1.2.1 Codificación de la información	28
1.2.2 Extracción de la fase de un sistema de proyección de franjas	31
1.3 CORRIMIENTO DE FASE A INTERVALOS REGULARES	32
1.4 CORRIMIENTO DE FASE A INTERVALOS NO REGULARES.	34
2. FORMULACIÓN DEL PROBLEMA	37
2.1 IDENTIFICACIÓN DEL PROBLEMA	37
2.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA	37
2.3 OBJETIVOS DEL TRABAJO DE GRADO	38
2.3.1 Objetivo General	38
2.3.2 Objetivos Específicos	38
3. DISEÑO DEL SOFTWARE DEL SISTEMA DE RECONSTRUCCIÓN TRIDIMIENSIONAL MULTI-CAPTOR	40
3.1 RECONOCIMIENTO DEL SISTEMA EN EL ENTORNO DE MATLAB	41
3.1.2 Reconocimiento de las cámaras en el entorno MATLAB.	44
3.1.3 Control de los sistemas de Observación	44
3.1.3.1 Inicio	45

1.3.1.2 Configuración	47
3.1.3.3 Congelar	48
3.1.3.4 En vivo	48
3.1.3.5 Guardar	49
3.2.1 Posicionamiento preliminar	51
4. CALIBRACIÓN DE LOS SISTEMAS INDEPENDIENTES	53
4.1 PASOS PARA LA CALIBRACIÓN INDEPENDIENTE	56
4.1.1 Matrices de Calibración.	60
5. CALIBRACIÓN ESTÉREO DEL SISTEMA	62
5.1 EXTRACCIÓN DE LAS ESQUINAS, CALIBRACIÓN Y HERRAMIENTAS ADICIONALES.	62
5.1.1 Lectura de las imágenes	65
5.1.2 Extracción de las esquinas de la cuadrícula	66
6. TRATAMIENTO DE LAS IMÁGENES	88
6.1 OBTENCIÓN DE LAS IMÁGENES DEL OBJETO A RECONSTRUIR	88
6.2 OBTENCIÓN DEL X, Y, Z DEL OBJETO A RECONSTRUIR.	90
6.3 EXTRACCIÓN DE LA FASE	96
6.4 UMBRAL INICIAL	97
6.5 RECONSTRUCCIÓN DE LOS SISTEMAS INDEPENDIENTES	99
6.6 RECONSTRUCCIÓN ESTÉREO (SISTEMA GLOBAL)	102
7. ANÁLISIS DE RESULTADOS	105
7.1 ANÁLISIS DEL OBJETO RECONSTRUIDO	105
7.1.1 Caracterización del objeto a reconstruir.	105
7.1.2 Recopilación de información del modelo	106
7.1.3 Procesamiento de la información obtenida	108
7.1.4 Integración y tratamiento de las imágenes	111
7.2 RESOLUCION Y PRECISIÓN DEL SISTEMA	116

7.2.1 Resolución del sistema	116
7.2.1.1 Resolución cualitativa	117
7.2.1.2 Resolución cuantitativa de los sistemas independientes	118
7.2.2 Precisión del sistema	121
8. CONCLUSIONES	127
9. RECOMENDACIONES	129
BIBLIOGRAFÍA	130
ANEXOS	132

LISTA DE TABLAS

	Pág.
Tabla 1. Clasificación de los métodos ópticos más comunes.	25
Tabla 2. Métodos ópticos de reconstrucción 3D en función de las dimensiones transversales del objeto, resolución, precisión y rango de medida.	26

LISTA DE FIGURAS

	Pág.
Figura 1. Procedimiento de reconstrucción 3D por vía óptica.	22
Figura 2. Composición básica de un sistema óptico	22
Figura 3. Reconstrucción 3D	24
Figura 4. Sistema de franjas proyectado	29
Figura 5. Corrimiento del sistema de franjas en función de Z	30
Figura 6. Interfaz del software de reconstrucción 3D	40
Figura 7. Error en la configuración de pantalla.	42
Figura 8. Extensión de la Pantalla	42
Figura 9. Franjas Proyectadas	43
Figura 10. Visualización de las cámaras dentro de la Interfaz	44
Figura 11. Controles de las Cámaras	45
Figura 12. Control para la Inicialización de las cámaras	46
Figura 13. Control para configuración de las cámaras	47
Figura 14. Control Congelar Imagen	48
Figura 15. Control para Reanudación captura de imágenes	49
Figura 16. Adquirir y Guardar Fotografías.	50
Figura 17. Dispositivos de sujeción	51
Figura 18. Posicionamiento premilimar del objeto a reconstruir	52
Figura 19. Placas de calibración	53
Figura 20. Tabla con cuadrícula ajedrezada	54
Figura 21. Soportes de cámaras y proyectores	55
Figura 22. Ubicación de la cuadrícula en las placas de calibración	56
Figura 23. Control para la calibración independiente	57
Figura 24. Franja proyectada capturada por las cámaras	57

Figura 25. Cuadro de diálogo para nombrar la calibración independiere	58
Figura 26. Advertencia para correr la cuadrícula en dirección Z	59
Figura 27. Dirección del eje Z	59
Figura 28. Posiciones para el cálculo de la fase	61
Figura 29. Adquisición de imágenes de la cuadrícula	63
Figura 30. Control para capturar imágenes simultaneas	64
Figura 31. Control para invocar el Software Camera Calibration Toolbox	64
Figura 32. Interfaz del software <i>Camera Calibration Toolbox</i>	65
Figura 33. Imágenes para la calibración en el ambiente Matlab	66
Figura 34. Selección de puntos para el área común	67
Figura 35. Secuencia de selección de puntos	68
Figura 36. Cuadro común seleccionado	69
Figura 37. Selección de los elementos diferenciales del área de trabajo	70
Figura 38. Extracción de las esquinas	71
Figura 39. Extracción de las esquinas para todas las imágenes	72
Figura 40. Calculo de los elementos diferenciales y coeficiente de distorsión	73
Figura 41. Imágenes tratadas	75
Figura 42. Error de reproyección	76
Figura 43. Parámetros extrínsecos	76
Figura 44. Orientación de la cámara	77
Figura 45. Error de reproyección modificado	78
Figura 46. Mala selección de una esquina	79
Figura 47. Modificación de la esquina seleccionada	79
Figura 48. Almacenaje de las imágenes de calibración	80
Figura 49. Modelo de distorsión completo	81
Figura 50. Modelo de distorsión tangencial	82
Figura 51. Modelo de distorsión radial	83

Figura 52. Control para invocar interfaz de <i>stereo_gui</i>	85
Figura 53. Interfaz de <i>stereo_gui</i>	85
Figura 54. Parámetros extrínsecos sistema estéreo	86
Figura 55. Visualización del modelo a reconstruir	89
Figura 56. Control para la capturar y guardar imágenes del modelo	89
Figura 57. Control para Tratamiento de las imágenes	90
Figura 58. Zona de superposición de los perfiles.	91
Figura 59. Superposición de los perfiles.	91
Figura 60. Pausa iniciar proceso de selección del punto de intersección	92
Figura 61. Aumento de la visualización de los perfiles	93
Figura 62. Advertencia para seleccionar el punto de intersección.	93
Figura 63. Selección del punto de intersección.	94
Figura 64. Cuadro de dialogo para ingresar posición en Z	94
Figura 65. Determinación de la posición de Z	95
Figura 66. Selección de la Región blanca dentro de la imagen a reconstruir	98
Figura 67. Advertencia para modificar el Umbral	98
Figura 68. Imagen de Confirmación del Umbral	99
Figura 69. Zona de visualización de la reconstrucción independiente	100
Figura 70. Reconstrucción independiente	102
Figura 71. Control para el empalme de las imágenes de reconstrucción	103
Figura 72. Objeto reconstruido	104
Figura 73. Objeto a reconstruir	106
Figura 74. Imágenes tomadas por el sistema independiente derecho	107
Figura 75. Imágenes tomadas por el sistema independiente izquierdo	107
Figura 76. Perfiles del Objeto a reconstruir	108
Figura 77. Reconstrucción del sistema independiente derecho, plano YZ	109
Figura 78. Aberraciones del objeto reconstruido	110

Figura 79. Integración de las reconstrucciones independientes (ejes coordenados en [mm])	111
Figura 80. Zona oculta para el sistema independiente derecho	112
Figura 81. Zona oculta para el sistema independiente izquierdo	113
Figura 82. Imagen de la reconstrucción tratada por medio de un mallado	114
Figura 83. Figura virtual	115
Figura 84. Figura real	116
Figura 85. Extracción de la profundidad de la ranura.	117
Figura 86. Tabla con tiras de cartulina superpuestas	118
Figura 87. Resolución sistema derecho	119
Figura 88. Resolución sistema izquierdo	121
Figura 89. Muestreo de datos para la determinación de la precisión del sistema	122
Figura 90. Reconstrucción 3D sección cilíndrica y su imagen real	123
Figura 91. Reconstrucción 3D en nubes de puntos y su imagen real	124
Figura 92. Reconstrucción 3D por secciones en nubes de puntos y su imagen real	125

LISTA DE ANEXOS

	Pág.
Anexo A. Aplicaciones Médicas	133
Anexo B. Aplicaciones Industriales	141
Anexo C. Herramientas de Matlab	147

RESUMEN

TITULO: ANÁLISIS PRELIMINAR DE UN SISTEMA DE RECONSTRUCCIÓN TRIDIMENSIONAL MULTI-CAPTOR: APLICACIONES MEDICAS E INDUSTRIALES.

AUTORES: MANUEL JULIÁN BARRERA AMOROCHO**
GONZALO AUGUSTO TORRES QUINTERO**

PALABRAS CLAVES: Reconstrucción tridimensional (3D), Proyección de franjas, Corrimiento de fase, Calibración, Sistema Multi-captor.

DESCRIPCIÓN:

La reconstrucción tridimensional es una técnica cuyas aplicaciones van desde el campo industrial hasta la medicina estética, estudios para la implementación y mejoramiento de ésta han sido analizados en todo el mundo, pero infortunadamente, los resultados de dichos estudios están altamente protegidos y son de difícil acceso.

El análisis desarrollado en esta investigación trata específicamente de la reconstrucción tridimensional por medio de la técnica óptica de proyección de franjas. En él se implementó un sistema multi-captor que consta de dos sistemas de proyección-observación y para el cual se desarrollaron algoritmos en el entorno MATLAB que facilita y agiliza la adquisición de los datos, el registro, la integración, la segmentación y el ajuste de superficies para la reconstrucción tridimensional de diversos objetos. Adicionalmente en este proyecto de investigación y como metodología innovadora, se realiza el empalme de dos reconstrucciones obtenidas independientemente por cada uno de los sistemas de proyección-observación, de esta manera se obtiene información metrológica de una superficie completa de los objetos que se deseen reconstruir, eliminando así zonas de difícil acceso de las cuales con un solo sistema captor sería imposible extraer dicha información metrológica. Este proceso se lleva a cabo partiendo de referencias obtenidas mediante la realización de calibraciones rigurosas de los sistemas.

* Trabajo de grado

**Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería Mecánica. Director: M.Sc. Carlos Ricardo Contreras Pico. Codirector. Ing. Omar Armando Gelvez Arocha.

ABSTRACT

TITLE: PRELIMINARY ANALYSIS OF A MULTI-CAPTOR THREE-DIMENSIONAL RECONSTRUCTION SYSTEM: MEDICAL AND INDUSTRIAL APPLICATIONS *

AUTHORS: MANUEL JULIÁN BARRERA AMOROCHO**
GONZALO AUGUSTO TORRES QUINTERO**

KEY WORDS: Three-dimensional Reconstruction (3D), Fringe Projection, Phase Stepping, Calibration, Multi-captor System.

DESCRIPTION:

Three-dimensional reconstruction is a technique which applications range from industrial to cosmetic medicine fields, studies for the implementation and refinement of this technique have been tested worldwide, but unfortunately, the results of these studies are highly protected and are difficult to get.

The analysis developed in this research specifically aboard three-dimensional reconstruction using the optical technique of fringe projection. In that analysis was implemented a multi-captor system consisting of two sets of projection-observation and which for are developed algorithms in MATLAB environment speeding up data acquisition, registration, integration, segmentation and adjustment reconstruction of three-dimensional surfaces for different objects. Additionally in this research project and as innovative methodology, is made the joint of two reconstructions obtained independently by each of the projection-observation systems, with that is possible to obtain metrology information from a complete surface of objects to reconstruct, eliminating areas of difficult access each with a single captor system would be unable to extract that information metrological. This process is carried out based on references obtained with rigorous calibration process of the systems.

* Graduate project

** Faculty of Physical-Mechanical Engineering, School of Mechanical Engineering. Director: M.Sc. Carlos Ricardo Contreras Pico. Co-Director. Eng. Omar Armando Gelvez Arocha.

INTRODUCCIÓN

En la actualidad el proceso de digitalización de la información toma gran fuerza, desde la medicina hasta la arqueología, pasando por un sinnúmero de campos en los cuales la información requiere ser tratada de manera no convencional. La extracción de información topográfica de un objeto; área, volumen, contorno, regiones cóncavas y convexas, detalles específicos en general, es un proceso requerido en ámbitos diversos para su análisis y estudio posterior, debido a esta necesidad, el auge de la reconstrucción tridimensional ha venido aumentando desde hace ya varios años, a manera de ejemplo podemos enfocarnos un poco en la medicina¹, observando trabajos como: mediciones intra-orales 3D, imágenes 3D y monitoreo no invasivo de las paredes vasculares, detección y monitoreo de la escoliosis, inspección de heridas, medición de la topografía de la piel para el uso en la cosmetología, por nombrar algunos de los trabajos más conocidos; además podemos dar un vistazo a las aplicaciones industriales², como: análisis de vibraciones, refractometría, medición global de deformación en superficies libres, medición de espesores de láminas metálicas forzadas, análisis de corrosión, medición de la rugosidad superficial e ingeniería inversa, por nombrar algunas de las aplicaciones de mayor potencial.

Sin embargo, muchas de estas aplicaciones se realizan con dispositivos robustos, voluminosos y por supuesto, con codificación de alta complejidad, haciendo difícil su programación y posterior uso, de tal manera, que se ve cerrado a unas cuantas aplicaciones, limitando su capacidad, como bien se sabe, la compra de un dispositivo de dichas características, representa una gran inversión a la entidad que quiere prestar servicios de alta calidad y por ende, su utilización deberá suplir dicha inversión.

¹Ver anexo: Aplicaciones médicas

² Ver anexo: Aplicaciones industriales

Para la ampliación de la gama de usos de un dispositivo de reconstrucción tridimensional, es necesario el desarrollo de un elemento versátil, que no se cierre a grandes áreas o ángulos de observación limitados que vayan a dificultar la recopilación de información, además, como hecho importante, cabe resaltar que la modificación del código usado para la reconstrucción se debe acoplar a las necesidades del usuario, ya que no es lo mismo tratar imágenes en escala de grises que tratar imágenes policromáticas, de tal manera que su utilización no requiera de una persona experta que deba estar supervisando siempre su buen uso.

El software trabajado en ambiente MATLAB y que es objeto del presente proyecto de grado, denominado: ANÁLISIS PRELIMINAR DE UN SISTEMA DE RECONSTRUCCIÓN TRIDIMENSIONAL MULTI-CAPTOR: APLICACIONES MÉDICAS E INDUSTRIALES, es capaz de adquirir información bidimensional por medio de dos sistemas de proyección – observación, la cual será tratada y con la cual posteriormente se obtendrá información tridimensional del objeto a reconstruir, dando así inicio al tratamiento de imágenes 3D, con su respectiva visualización y posibilidad de manipulación. Este software no sólo cumplirá labores de reconstrucción tridimensional, también, permitirá el análisis posterior de dichas figuras reconstruidas y cabe la posibilidad de realizar (en estudios posteriores) la interfaz de nuestro software con distintos programas de tópicos varios, para su posterior trabajo o manipulación.

1. ANTECEDENTES

1.1 RECONSTRUCCIÓN TRIDIMENSIONAL

1.1.1 Conceptos generales de la reconstrucción tridimensional

La reconstrucción tridimensional por métodos ópticos, es el proceso mediante el cual, la topografía de un objeto es digitalizada. Es decir, a partir de un sistema de referencia (X, Y, Z) se calculan digitalmente las coordenadas de los puntos observados en la superficie de un cuerpo, empleando una técnica óptica de reconstrucción 3D. Debido a la naturaleza bidimensional de los dispositivos de digitalización de imágenes, los métodos ópticos de reconstrucción 3D permiten codificar la información tridimensional en diferentes alternativas tales como: deformación de una línea proyectada a través de un cuerpo, variación bidimensional del paso y orientación de franjas proyectadas o retardo temporal de pulsos enviados a la superficie, por ejemplo. De esta manera, una variable de naturaleza óptica es alterada en función de la topografía de la superficie del cuerpo. El tratamiento digital para la reconstrucción 3D, permite inicialmente:

- a) Calcular el valor de la variable óptica utilizada como codificación para todos los puntos de la superficie digitalizada.
- b) Calcular las alturas correspondientes a partir de la información obtenida anteriormente.

El proceso de convertir digitalmente la variable óptica en altura puede realizarse de dos maneras. Por un lado, conociendo la ecuación fundamental que relaciona la altura de un punto con el valor de la variable óptica asociada en función de los

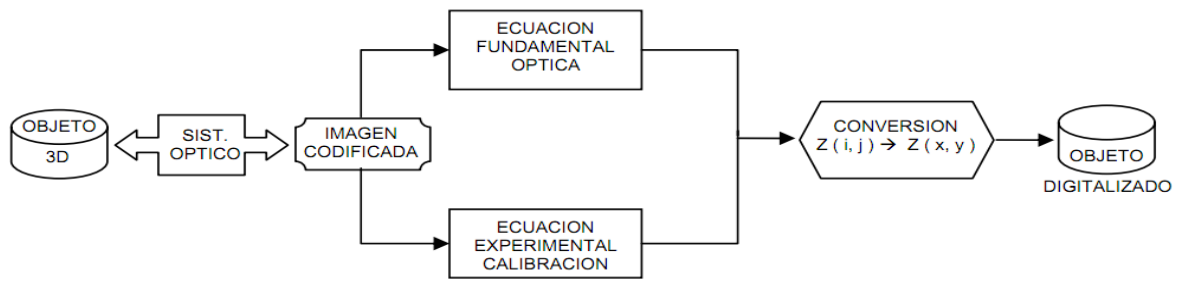
parámetros del sistema. La precisión para medir los parámetros del sistema y las aproximaciones utilizadas para obtener la ecuación fundamental, determinan principalmente la precisión y resolución del sistema de medida 3D implementado. Por otro lado, una etapa de calibración permite obtener una ecuación experimental de conversión que tiene en cuenta todos los parámetros del sistema.

Usando una de estas dos alternativas, el tratamiento digital permite obtener una imagen, donde cada posición (i, j) digitalizada de la superficie, posee la altura correspondiente medida en función de las características metrológicas del sistema. Por lo tanto, otro proceso de conversión es requerido para transformar posiciones del sistema de adquisición $Z(i, j)$, en posiciones en el espacio coordinado del objeto $Z(x, y)$. De igual manera, existen 2 alternativas para realizar esta conversión.

Conociendo el factor de ampliación del sistema óptico de adquisición y asumiendo un sistema libre de aberraciones geométricas, se puede realizar un proceso lineal de conversión de coordenadas (i, j) del espacio imagen sobre la CCD a coordenadas (x, y) del espacio objeto. Por otro lado, un procedimiento de calibración del sistema formador de imágenes, permite obtener una ecuación experimental de conversión del espacio (i, j) al espacio (x, y) que tenga en cuenta las influencias de las aberraciones, si son considerables. La figura 1.1 nos muestra el procedimiento para la reconstrucción tridimensional por medio de la vía óptica.³

³ PEREZ, Zayra y; ROMERO, Lenny. Sistema óptico de reconstrucción 3D para la detección de ampollas en recubrimientos. Proyecto de grado. Universidad industrial de Santander. 2004. p12-13

Figura 1. Procedimiento de reconstrucción 3D por vía óptica.

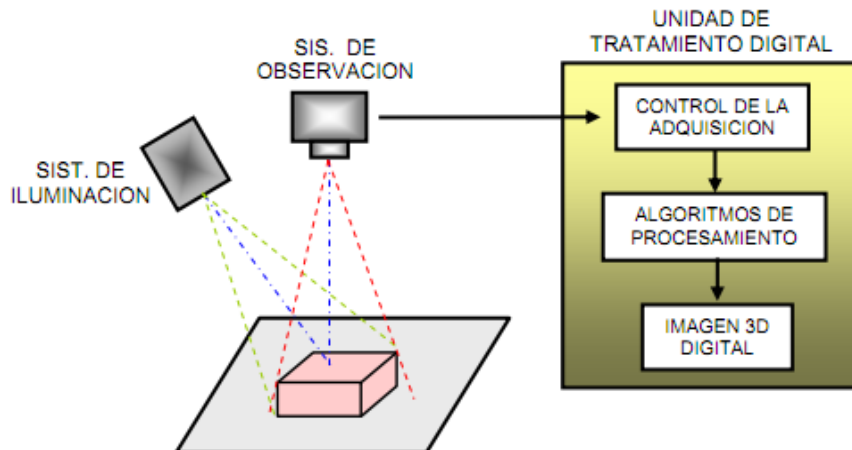


Fuente. [3]

1.1.2 Clasificación de los métodos ópticos

Un método óptico de reconstrucción 3D hace referencia a una técnica que a partir de un sistema óptico codifica la topografía del cuerpo en función de los parámetros del sistema. Generalmente un sistema óptico de reconstrucción 3D se compone de tres unidades.

Figura 2. Composición básica de un sistema óptico



Fuente. [5]

Como podemos observar en la figura 1.2, un sistema óptico consta de tres elementos fundamentales:

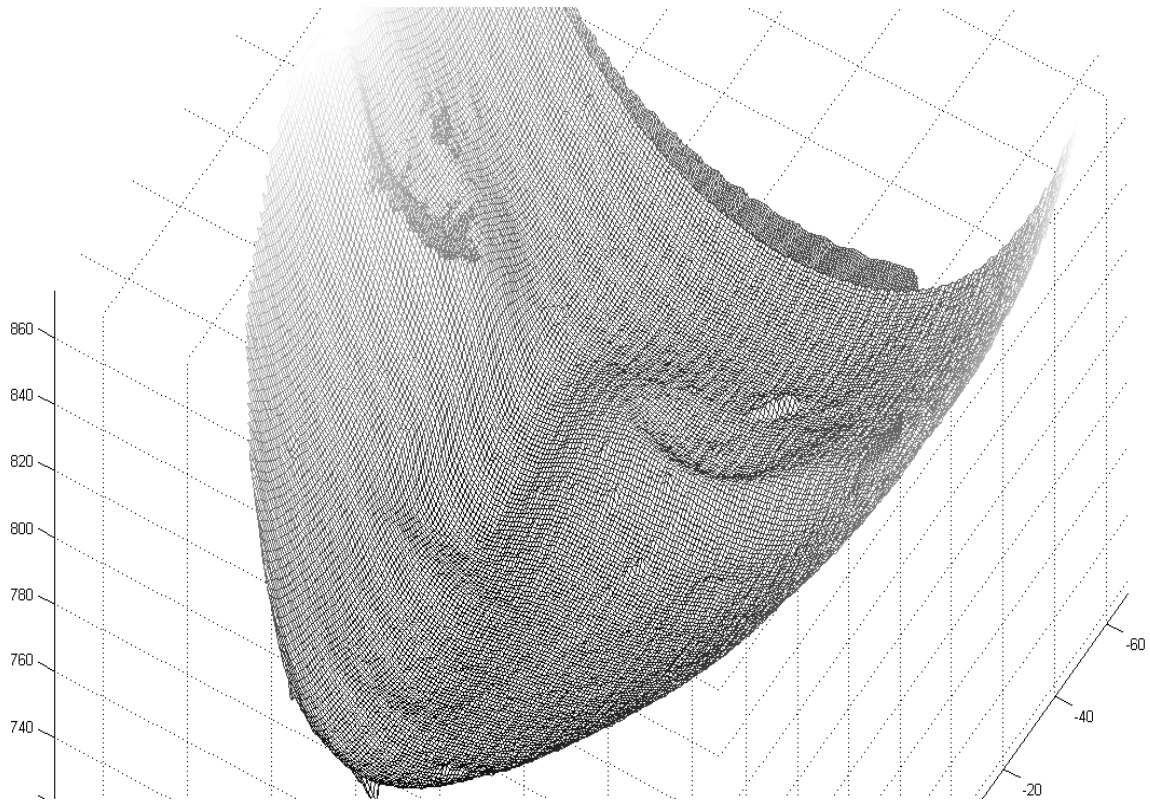
- Sistema de iluminación o cabeza óptica.
- Sistema de adquisición.
- Unidad de control y tratamiento digital.

El sistema de iluminación es diseñado en función del método óptico y las condiciones metrológicas de reconstrucción: campo de observación, resolución, profundidades de campo, entre otras.

La utilización de dispositivos telecéntricos o no telecéntricos formadores de imágenes, la proyección de un punto línea laser, la proyección de un patrón de intensidad sobre la superficie del cuerpo son algunos ejemplos que permiten definir la óptica utilizada en el sistema de iluminación. De igual manera, existen otros métodos de los cuales no hay control directo sobre la fuente de iluminación. En este caso solo es necesario adaptar el sistema de adquisición en función de las alternativas de codificación. Por ejemplo en el uso de visión estéreo se requieren al menos dos imágenes con diferentes ángulos de observación de la misma superficie.

El sistema de adquisición de la información, se diseña en función de la variable óptica que codifica la topografía. Generalmente se utilizan transductores de señal luminosa: Puntual (Foto-detector), lineal (Línea CCD) o bidimensional (Matriz CCD). La interface de digitalización de la información se sincroniza con el sistema de iluminación para adquirir el número de datos necesarios en la digitalización de la superficie del cuerpo. En cada posición (x, y) del sistema de barrido, se debe digitalizar la imagen del punto laser, desplazado por la topografía del cuerpo. Sin embargo, proyectando un patrón de líneas sobre la superficie, no son necesarios los sistemas adicionales de barrido y se requiere al menos una imagen, dependiendo del método óptico de proyección estructurado diseñado.

Figura 3. Reconstrucción 3D



Fuente. Los autores

La unidad de tratamiento digital de la información, permite realizar las transformaciones digitales necesarias para el proceso inverso de reconstrucción 3D. Etapas intermedias como almacenamiento de la información, pre-tratamiento de datos, etapas de calibración experimental, etc., son requisitos en el tratamiento de imágenes.

Aunque los métodos ópticos pueden clasificarse de diferentes maneras; bibliográficamente, se puede agrupar en función del sistema de iluminación como métodos pasivos y activos. En los métodos pasivos no es necesario controlar la fuente de iluminación, pero el tratamiento digital para obtener información topográfica requiere un alto esfuerzo computacional. En los métodos activos, el uso de un patrón de radiación simplifica el problema de la medida de la topografía.

Tabla 1. Clasificación de los métodos ópticos más comunes.

Clasificación		Método	Principio de Codificación	
A C T I V O S	Luz Estructurada	láser	El patrón puede ser un punto, una línea, múltiples líneas o un patrón definido.	Se basa en triangulación, la altura para cada punto se codifica en el corrimiento del patrón con respecto a una referencia debido a la topografía del cuerpo.
		Binaria	Se utilizan patrones binarios bidimensionales que codifican la información en uno ó en dos ejes	
		Nivel de gris	Usa un patrón de iluminación en niveles de gris.	
		Cromática	El patrón de iluminación está asociado a una matriz de color	
	Métodos a medida de fase	Proyección de franjas	La altura del objeto se codifica en la deformación local del paso y orientación de las franjas proyectadas que están contenidas en un término de fase, que se debe recuperar.	
		Moiré	La altura del cuerpo se codifica en curvas de Nivel, obtenidas por efecto Moiré debido a la proyección de sistemas de franjas de paso similar.	
		Interferometría	Consiste en variar el camino óptico y obtener información de fase de las franjas de interferencia. Puede ser a incidencia normal, oblicua o Interferometría a luz blanca	
Tiempo de Vuelo		Para calcular la distancia absoluta de un punto del objeto se envía un pulso de luz (usualmente láser), y se mide el retardo de tiempo del haz reflejado,		
	Microscopía Confocal	El cuerpo se ubica en un sistema de desplazamiento axial y la altura se determina identificando la posición donde se encuentra el máximo de intensidad del sistema Confocal.		
P A S I V O S	Visión estéreo	A partir de dos imágenes tomadas en diferentes ángulos, se busca la correspondencia entre características y se halla la profundidad, esto es semejante al modelo binocular humano.		
	Shape from Shading	La información 3D se encuentra a través del manejo de sombras		

Fuente. [5]

En general, la elección de una técnica óptica de reconstrucción 3D depende de varias condiciones que impone el objeto en estudio. Se destacan principalmente:

- a) Medición absoluta de la topografía con respecto a un sistema de referencia o medición relativa con respecto a la posición relativa en el sistema de adquisición y el cuerpo.
- b) Medición para el estudio estático, sin movimiento en el intervalo de adquisición o dinámicos con objetos en movimiento o deformación superficial.
- c) Características del cuerpo como grado de absorción luminosa, discontinuidades en la topografía, sombras, etc.

Tabla 2. Métodos ópticos de reconstrucción 3D en función de las dimensiones transversales del objeto, resolución, precisión y rango de medida⁴.

MÉTODO	RANGO DE MEDIDA	DIM. TRANSV. DEL OBJETO	PRECISIÓN $\pm \Delta Z$	RESOLUCIÓN RELATIVA
Proyección de Franjas	1 mm : 300 mm	1000 mm x 1000 mm	0.5 mm	1/1000 : 1/5000
Proyección de Franjas	0.1 mm : 50 mm	100 mm x 100 mm	0.05 mm	1/1000 : 1/5000
Proyección de Franjas	0.05 mm : 1 mm	50 mm x 50 mm	0.001 mm	1/500 : 1/5000
Microscopía interferométrica	0.001 mm : 0.05 mm	15 mm x 10 mm	0.0001 mm	1/100 : 1/5000
Interferometría con luz Blanca	2 nm : 400 μm	5 mm x 5 mm	2 nm	1/100000
Microscopía Confocal	0.001 mm : 0.056 mm	1 mm x 1 mm	50 nm	1/100 : 1/5000
Interferometría UV	0.001 mm : 0.01 mm	0.3 mm x 0.3 mm	0.5 nm	1/2000
AFM	0.1 nm : 2000 nm	0.1 mm x 0.1 mm	0.01 nm	1/5 : 1/10000

Fuente. [4]

⁴ MENESES, Jaime. Systeme optique pour la mesure de deformations de la voute plantaire sous charge statique. Tesis Doctoral. Université de Franche-Comté. 1998. p. 28

De igual manera, la elección de un método óptico de reconstrucción 3D depende de las condiciones metrológicas exigidas en la reconstrucción. Así un sistema de proyección es adecuado para campos de observación grandes en el rango de milímetros a metros, con resolución del orden de decenas de milímetros. Para resolución del orden de varias decenas de nanómetros a micras, las técnicas interferométricas son adecuadas; aunque el campo de observación es limitado a algunas décimas de milímetros.

Las técnicas ópticas de reconstrucción 3D han tenido gran acogida, ya que es una tecnología no invasiva, no requiere altos costos de implementación o supervisión, siendo cuantitativamente más objetiva, fiable y consistente que la inspección humana. Pueden alcanzar excelentes resoluciones, son de fácil automatización, evitando trabajos visuales engorrosos y altamente repetitivos. Actualmente se aplican en diversos procesos científicos y militares, extendiéndose su uso además, en un amplio rango de sectores industriales para la automatización de tareas anteriormente reservadas a la inspección visual humana. El uso de técnicas ópticas de reconstrucción 3D, se está extendiendo cada día más entre los diferentes sectores industriales, tanto en procesos de control de los productos, como en el control de los procesos de producción. La utilización de esta tecnología, y los beneficios que conlleva, se centra principalmente en todas aquellas áreas en las que es importante determinada información metrológica, como por ejemplo, en los sectores industriales del metal, alimentación (envasados), medicina, computación y electrónica, en el sector textil para la medición del ancho de los tejidos, medición de parámetros en ensayos de control de calidad, como la medición del número de fibras que componen un hilo a partir de imágenes extraídas del microscopio, la capacidad de detectar personas, huellas dactilares, matriculas, etc.

1.2 RECONSTRUCCIÓN 3D POR MEDIO DE PROYECCIÓN DE FRANJAS

Conscientes de las necesidades que existen actualmente en el medio científico-industrial, se fundamenta esta investigación con el objetivo de realizar un análisis topográfico de un objeto de estudio, basado en el procesamiento de datos mediante algoritmos computacionales. La técnica de proyección de franjas es el método óptico utilizado para el desarrollo de este proyecto de grado, fundamentado en la deformación que sufren dichas franjas al ser proyectadas sobre la topografía de cualquier tipo de objeto.

El sistema óptico de proyección de franjas consta en su forma más común de dos partes principales: un sistema de proyección, el cual proyecta un patrón de franjas sobre una superficie; y un sistema de observación que adquiere las imágenes con su respectivo patrón de franjas anteriormente proyectado. La deformación sufrida por el patrón de franjas proyectado es debida a la topografía del objeto. Como es lógico suponerlo, si tenemos una superficie plana, el patrón de franjas proyectado no sufrirá deformación alguna, manteniendo su paso constante, mientras que si se introduce un objeto, cuya superficie no es plana, el patrón proyectado sufrirá una deformación, debido a su topografía.

1.2.1 Codificación de la información

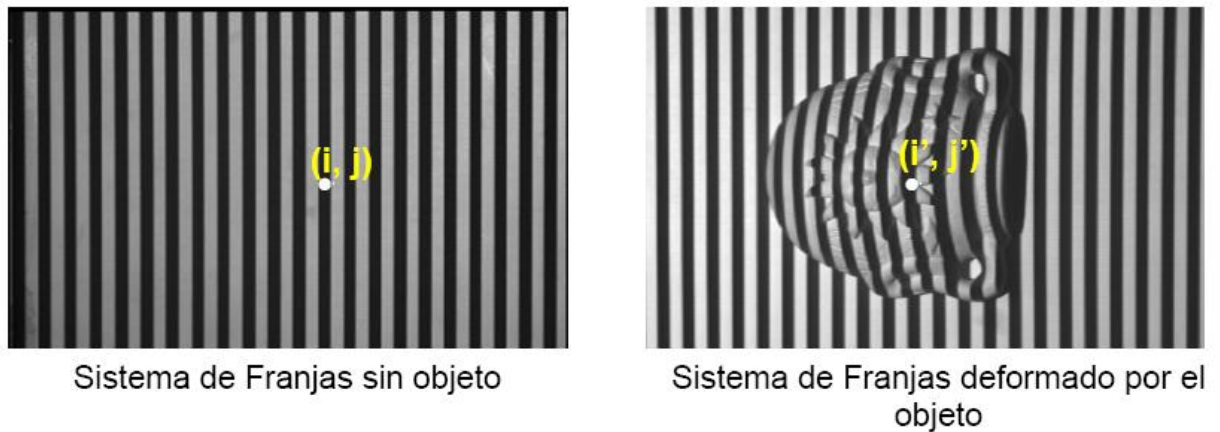
La intensidad luminosa que registra la cámara CCD se puede interpretar como:

$$I(x, y) = I_o(x, y) + A(x, y) \cos[\phi(x, y)]. \quad (1)$$

Donde $I_o(x, y)$ nos indica la intensidad luminosa del fondo continuo, $A(x, y)$ es el factor de contraste y donde $\phi(x, y)$ corresponde a la función fase, la cual nos da un indicio de la modificación del paso de las franjas al colocar el cuerpo. Es decir, para la n -ésima franja ubicada sobre el plano de referencia en el pixel (i, j) , se

introduce un desplazamiento al pixel (i' , j'), lo que es correspondiente a la altura del cuerpo. (Ver figura 1.4).

Figura 4. Sistema de franjas proyectado

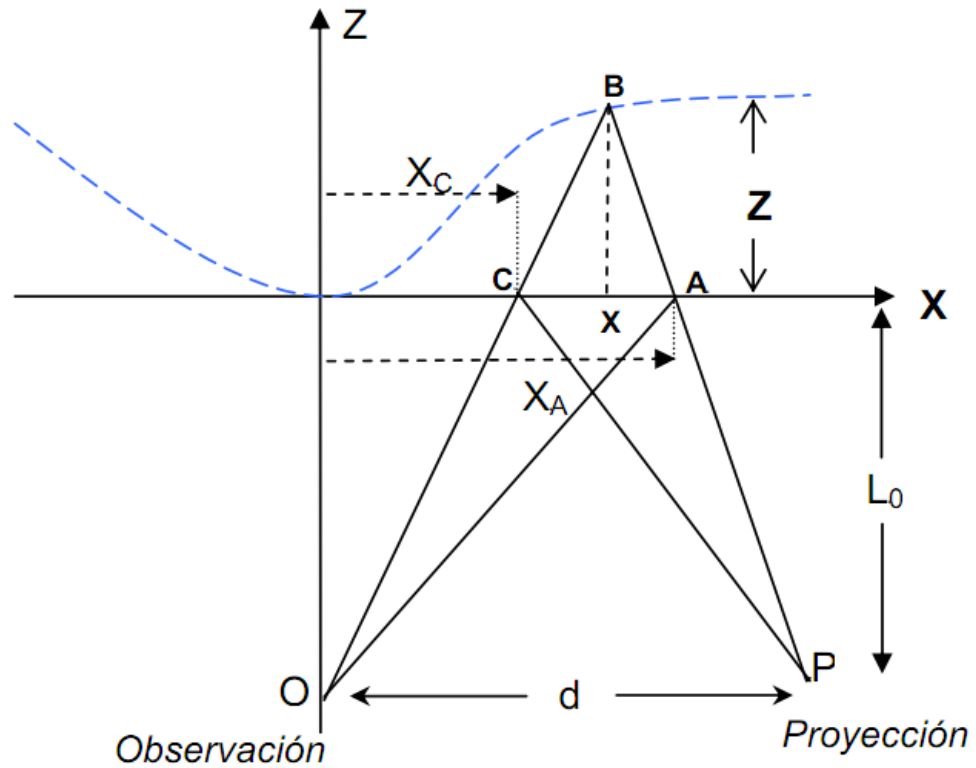


Fuente. [1]

De esta manera el corrimiento producido por el cuerpo, genera una alteración en la función fase $\phi(x, y)$ de la ecuación (1). De acuerdo a esto, matemáticamente el proceso de codificación consiste en deformar la función $\phi(x, y)$ del sistema de franjas y el proceso de reconstrucción 3D consiste en recuperar la función $\phi(x, y)$ y realizar la conversión inversa en altura.

Sobre el plano de referencia existe una fase inicial que es característica del sistema y ésta depende del paso de proyección y de los parámetros del sistema de proyección e inclusive de las aberraciones geométricas. La figura 1.5 nos muestra un sistema óptico de proyección y observación. P y O son las pupilas de entrada y salida de los sistemas formadores de imágenes, el plano XY define el plano de referencia donde se enfoca el sistema de observación. El eje óptico del sistema de observación es perpendicular al plano de referencia, la altura se mide, en dirección al eje Z . la franja que llega a A tiene una fase inicial ϕ_A , como se muestra en la figura 1.5.

Figura 5. Corrimiento del sistema de franjas en función de Z



Fuente. [1]

Cuando se ubica el objeto, la franja es interceptada por la superficie del cuerpo en B, usando un modelo no tele-céntrico de observación el punto B es proyectado en C sobre el plano de referencia, luego se debe cumplir que:

$$\phi_{obj}(x_c) = \phi(x_A, 0) = \phi(x_c + \Delta), \quad (2)$$

donde $\Delta = \overline{AC}$.

La codificación de la información, será entonces, la deformación de la fase inicial en función del corrimiento Δ , que a su vez es función de Z.

Usando el modelo de observación no tele-céntrico, el corrimiento Δ se puede calcular a partir de relaciones geométricas como:

$$\Delta = \frac{z d}{z - L_o} . \quad (3)$$

definiendo:

$$\Delta\phi_{obj} = \phi_{obj}(x_C) - \phi(x_C, 0), \quad (4)$$

obtenemos:

$$\Delta\phi_{obj} = \frac{2 * \pi * \Delta}{P_o} = \frac{2 * \pi * d}{P_o * (z - L_o)}, \quad (5)$$

Donde P_o es el paso de las franjas medidas sobre el plano de referencia. De este modo, extrayendo $\Delta\phi_{obj}$ del sistema de franjas sobre el cuerpo y ϕ del sistema de franjas sobre el plano de referencia, se puede utilizar la ecuación fundamental (5) para calcular $z(x, y)$.

1.2.2 Extracción de la fase de un sistema de proyección de franjas

Existen varias técnicas de extracción de fase donde se utilizan uno o varios diagramas de intensidad. Para obtener nuestro parámetro $\phi(x, y)$, los métodos más utilizados son: transformada de Fourier (TF)⁵, regresión no lineal⁶ y corrimiento de fase.

La tarea principal de un método de extracción de fase es la de obtener $\phi(x, y)$ a partir de imágenes digitalizadas

⁵ Ver <<http://upcommons.upc.edu/ocw/diposit/material/33393/44180.pdf>>

⁶ Ver <<http://www.eumed.net/cursecon/medir/no-lineal.htm>>

1.3 CORRIMIENTO DE FASE A INTERVALOS REGULARES

La ecuación (1), nos indica la existencia de tres funciones incógnitas; $I_o(x, y)$, $A(x, y)$ y $\phi(x, y)$. Para calcular $\phi(x, y)$ en cada punto de la imagen del sistema de franjas, se requieren al menos tres datos en intensidad linealmente independientes. Éstos datos se obtienen experimentalmente desplazando las franjas con respecto al cuerpo introduciendo una fase adicional φ_i en un rango de $0 - 2\pi$. En éste rango se adquieren N sistemas de franjas desplazadas una fase φ_i definida por:

$$\varphi_i = \frac{2\pi}{N}(i - 1) \quad i = 1, 2, 3 \dots N. \quad (6)$$

De ésta manera, la ecuación (1) puede ser escrita como:

$$I(x, y) = I_o(x, y) + A(x, y) \cos[\phi(x, y) - \varphi_i]. \quad (7)$$

Utilizando las propiedades ortogonales de las funciones sinusoidales se encuentra:

$$\sum_{i=1}^N I_i \cos(\varphi_i) = \frac{NA(x, y)}{2} * \cos[\phi(x, y)], \quad (8)$$

$$\sum_{i=1}^N I_i \sin(\varphi_i) = \frac{NA(x, y)}{2} \sin[\phi(x, y)], \quad (9)$$

$$\sum_{i=1}^N I_i = NI_o. \quad (10)$$

A partir de estas ecuaciones se calcula $\phi(x, y)$ y $A(x, y)$, usando:

$$tg(\phi) = \frac{\sum_{i=1}^N I_i \sin(\phi_i)}{\sum_{i=1}^N I_i \cos(\phi_i)}, \quad (11)$$

$$A(x, y) = \frac{2}{N} \sqrt{\left(\sum_{i=1}^N I_i \sin(\phi_i)\right)^2 + \left(\sum_{i=1}^N I_i \cos(\phi_i)\right)^2}. \quad (12)$$

Para el algoritmo a 4 imágenes, es decir, $N=4$. Las ecuaciones generales para el corrimiento de fase serán:

$$tg(\phi) = \frac{I_2 - I_4}{I_1 - I_3}, \quad (13)$$

$$2A(x, y) = \sqrt{(I_2 - I_4)^2 + (I_1 - I_3)^2}. \quad (14)$$

Donde I_1 equivale a un corrimiento de fase $\phi_1 = 0$, I_2 corresponde a un corrimiento de fase $\phi_2 = \frac{\pi}{2}$; I_3 corresponde a un corrimiento de fase $\phi_3 = \pi$ y por último I_4 corresponde a un corrimiento de fase $\phi_4 = \frac{3\pi}{2}$.

De esta manera, conociendo ϕ_i y los valores de intensidad I_i se calcula la función $\phi(x, y)$.

La ventaja de esta técnica es que permite obtener $\phi(x, y)$ para cada punto independiente de la influencia de los puntos vecinos. Regiones sin información de franjas (regiones sombreadas), no afectan regiones útiles, a diferencia del método de la transformada de Fourier.

1.4 CORRIMIENTO DE FASE A INTERVALOS NO REGULARES.

En el método de corrimiento de fase a intervalos no regulares, la fase se calcula utilizando varios sistema de franjas deformadas por el cuerpo pero desplazadas a intervalos no regulares de ψ_k . Introduciendo un corrimiento no regular, al sistema de franjas deformado y resolviendo un sistema de ecuaciones lineales, se calcula la fase para cada punto.

De igual manera asignamos el valor de la intensidad luminosa como:

$$I_k(x, y) = I_o(x, y) + A(x, y) \cos[\phi(x, y) + \psi_k]. \quad (15)$$

Como bien sabemos, la ecuación (15) nos indica la intensidad luminosa registrada por la cámara CCD del sistema de franjas deformadas. $I_o(x, y)$ representa la intensidad luminosa del fondo continuo, $A(x, y)$ es el factor contraste, $\phi(x, y)$ es la fase del sistema de franjas y ψ_k representa el factor de fase no regular, introducido para cada sistema de franjas.

A partir de la ecuación (15) para cada punto de la imagen se obtiene:

$$I_k = a_o + a_1 \cos(\psi_k) + a_2 \sin(\psi_k). \quad (16)$$

Conociendo ψ_k e I_k de la ecuación (16), permite definir un sistema lineal de k ecuaciones y tres incógnitas. Utilizando una función de error E y usando el método de mínimos cuadrados se pueden calcular a_o , a_1 y a_2 :

$$E = \sum_{k=0}^{N-1} (I_k - I'_k)^2 = \sum_{k=0}^{N-1} (a_o + a_1 \cos(\psi_k) + a_2 \sin(\psi_k) - I'_k)^2, \quad (17)$$

donde I'_k son los valores de intensidad experimentales.

Calculando $\frac{\partial E}{\partial a_0} = 0$, $\frac{\partial E}{\partial a_1} = 0$ y $\frac{\partial E}{\partial a_2} = 0$, se obtiene un sistema lineal de ecuaciones, que se puede escribir matricialmente como:

$$A(\psi_k)a = b(\psi_k), \quad (18)$$

donde:

$$A(\psi_k) = \begin{pmatrix} N & \sum_{k=0}^{N-1} \cos(\psi_k) & \sum_{k=0}^{N-1} \sin(\psi_k) \\ \sum_{k=0}^{N-1} \cos(\psi_k) & \sum_{k=0}^{N-1} \cos^2(\psi_k) & \sum_{k=0}^{N-1} \cos(\psi_k) * \sin(\psi_k) \\ \sum_{k=0}^{N-1} \sin(\psi_k) & \sum_{k=0}^{N-1} \cos(\psi_k) * \sin(\psi_k) & \sum_{k=0}^{N-1} \sin^2(\psi_k) \end{pmatrix}$$

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} ; \quad b(\psi_k) = \begin{pmatrix} \sum_{k=0}^{N-1} I'_k \\ \sum_{k=0}^{N-1} I'_k * \cos(\psi_k) \\ \sum_{k=0}^{N-1} I'_k * \sin(\psi_k) \end{pmatrix}$$

De la ecuación (18) se obtiene:

$$a = A^{-1}(\psi_k) b(\psi_k). \quad (19)$$

Finalmente a partir de a_1 y a_2 se calcula $\phi(x, y)$, según la siguiente ecuación:

$$\phi = -tg^{-1} \left(\frac{a_2}{a_1} \right). \quad (20)$$

Así, la matriz A de (3×3) es conocida a partir de los valores ψ_k e I_k , para cada punto del sistema de franjas. El vector $b(\psi_k)$ se conoce para cada punto de la imagen del sistema de franjas usando los valores de ψ_k e I_k . Calculando A^{-1} y multiplicándolo por $b(\psi_k)$ se obtiene el valor de a donde sus elementos corresponden a a_0, a_1 y a_2 . La fase $\phi(x, y)$ se obtiene según la ecuación (20). Esta técnica es utilizada en sistemas ópticos donde no se puede introducir un corrimiento regular de fase.

2. FORMULACIÓN DEL PROBLEMA

2.1 IDENTIFICACIÓN DEL PROBLEMA

Aunque resulta fácil encontrar en el mercado ciertos dispositivos de reconstrucción tridimensional, la mayoría de ellos no son versátiles, no están disponibles para adquisición de información de la superficie de un objeto con alta resolución, y normalmente son sistemas con software protegido los cuales no se pueden modificar, ni adaptar a necesidades particulares.

Debido a esto surge la necesidad de construir un dispositivo de reconstrucción tridimensional que permita extraer información topográfica de cualquier objeto y que además conste de dos módulos de observación eliminando así las regiones muertas u ocultas, es decir aquellas zonas que serían de difícil acceso o difíciles de explorar con un solo módulo de proyección-observación. Adicionalmente este dispositivo debe contar con una resolución apropiada (500 micras para el presente trabajo), estar abierto a futuras mejoras, y enfocarse en la adaptabilidad que pueda tener en la industria y particularmente en la medicina estética.

2.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA

El diseño y construcción de un sistema de reconstrucción tridimensional con dos módulos de proyección-observación independientes y con alta resolución, permitirá establecer bases sólidas, con las cuales se podrá dar fin a la fase 1 del proyecto de investigación que se adelanta entre la Universidad Industrial de Santander (UIS) y la empresa IDECA S.A., la cual consiste en desarrollar un

dispositivo de reconstrucción tridimensional a 360° de observación y con alta resolución que permita realizar aplicaciones en medicina estética.

Adicionalmente la robustez con la que contará el software que se pretende desarrollar, permitirá adaptarlo para realizar aplicaciones específicas en los campos investigativos, industriales y médicos.

2.3 OBJETIVOS DEL TRABAJO DE GRADO

2.3.1 Objetivo General

- ✓ Complementar la misión de la Universidad Industrial de Santander (UIS) de crear excelentes relaciones con la industria, desarrollando un dispositivo preliminar de exploración 3D que permita extraer información metrológica de la superficie de un objeto a partir de múltiples vistas.

2.3.2 Objetivos Específicos

- ✓ Diseñar y construir un sistema de reconstrucción tridimensional mediante la utilización de dos módulos independientes de proyección-observación, con una resolución de 500 [μm].
- ✓ Ensamblar el sistema de reconstrucción tridimensional con dispositivos de sujeción livianos de tal manera que su movilización no sea complicada.
- ✓ Crear una interfaz en el ambiente MATLAB que permita agilizar y facilitar la adquisición de los datos, el registro, la integración la segmentación y el

ajuste de superficies para la reconstrucción tridimensional de superficie de diferentes objetos.

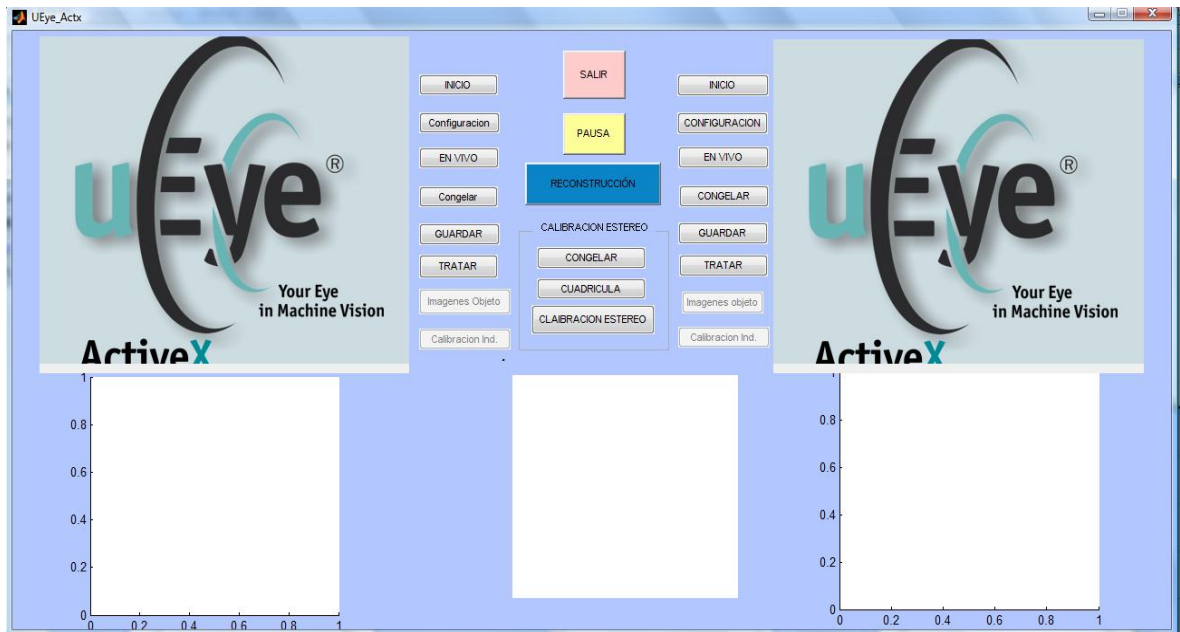
- ✓ Realizar el proceso de integración de las secciones tridimensionales obtenidas a partir de cada uno de los módulos de proyección-observación eliminando el problema de las denominadas regiones ocultas.
- ✓ Determinar experimentalmente la resolución y la precisión del sistema desarrollado a partir de la reconstrucción tridimensional de objetos con dimensiones conocidas.
- ✓ Garantizar que el software desarrollado esté disponible para futuras mejoras y además, esté en capacidad de adaptarse a necesidades requeridas en los campos investigativos, industriales o médicos.

3. DISEÑO DEL SOFTWARE DEL SISTEMA DE RECONSTRUCCIÓN TRIDIMENSIONAL MULTI-CAPTOR

Este trabajo consta de dos sistemas de proyección-observación, conformado por dos Video Proyectoros (el primero Mitsubishi SL-4U y el segundo Aigo-PT6216) y dos cámaras CCD (UI-2310-M). Los cuáles serán reconocidos dentro del ambiente MATLAB.

En la figura 6. se puede apreciar la interfaz del software diseñado.

Figura 6. Interfaz del software de reconstrucción 3D



Fuente: Los autores

Durante el desarrollo de este capítulo se describirá el proceso de reconstrucción tridimensional explicando la interfaz del software y cada uno de los controles implementados en la misma, así como su función y la manera correcta de utilizarlos.

3.1 RECONOCIMIENTO DEL SISTEMA EN EL ENTORNO DE MATLAB

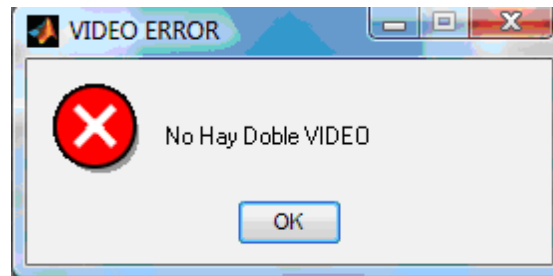
Para llevar a cabo todos los procesos de reconstrucción tridimensional, es necesario que los sistemas de proyección y observación sean reconocidos por el entorno MATLAB, adicionalmente resulta imprescindible que a través de los sistemas de proyección (Video Proyector) se proyecten las franjas, las cuales como ya se indicó son la base para la metodología implementada en esta investigación.

3.1.1 Proyección de las franjas

Las franjas proyectadas a través de los Video Proyectores sobre la superficie del objeto a reconstruir no pueden ser franjas aleatorias, es decir no pueden ser cualquier tipo de franjas, es necesario que todas las franjas tengan el mismo grosor, que su intensidad este entre 0 y 255, y que su enfoque sea el adecuado. Adicionalmente se requiere trabajar a lo largo de los dos monitores del ordenador utilizado, es decir durante la conexión del Video Beam al ordenador, hay que configurar la pantalla y ajustar la resolución para el monitor uno en la máxima permitida (1366x768) y para el monitor dos en 800x600 píxeles.

Dentro de la codificación desarrollada se evalúa si está configurada correctamente la extensión de los dos monitores de la pantalla, de no estarlo, se producirá un error como el que se muestra en la siguiente figura.

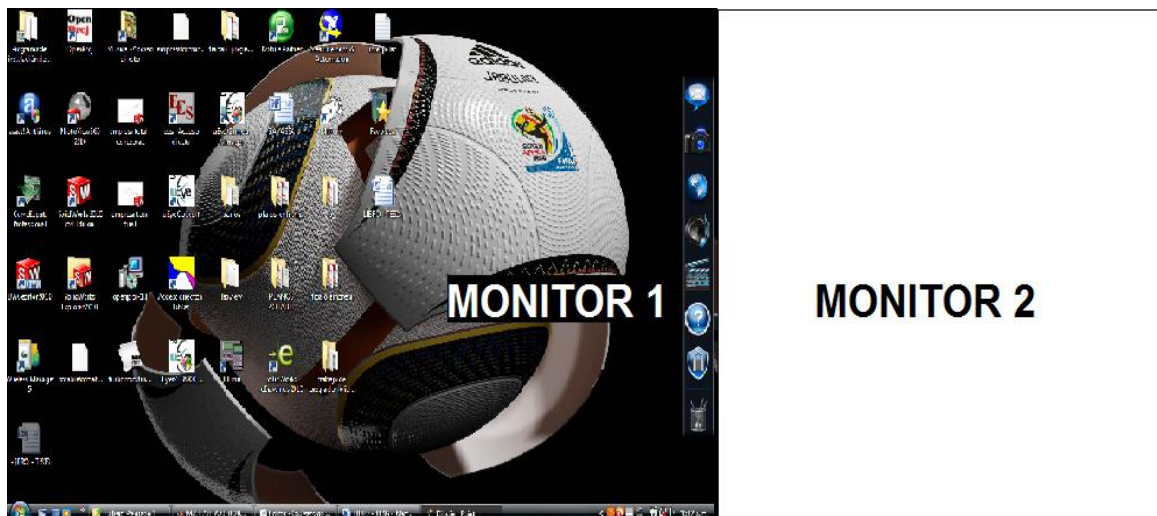
Figura 7. Error en la configuración de pantalla.



Fuente. Los autores

Con esto se asegura que los patrones que se ubiquen dentro de la zona del monitor dos (a la derecha de la pantalla Figura 3.3.) son los que se van proyectar por medio del Video Beam, dejando así como zona de trabajo libre al monitor uno. Además se comprueba que la resolución del monitor dos no sea diferente a 800x600 píxeles, la cual es la óptima para la proyección de las franjas.

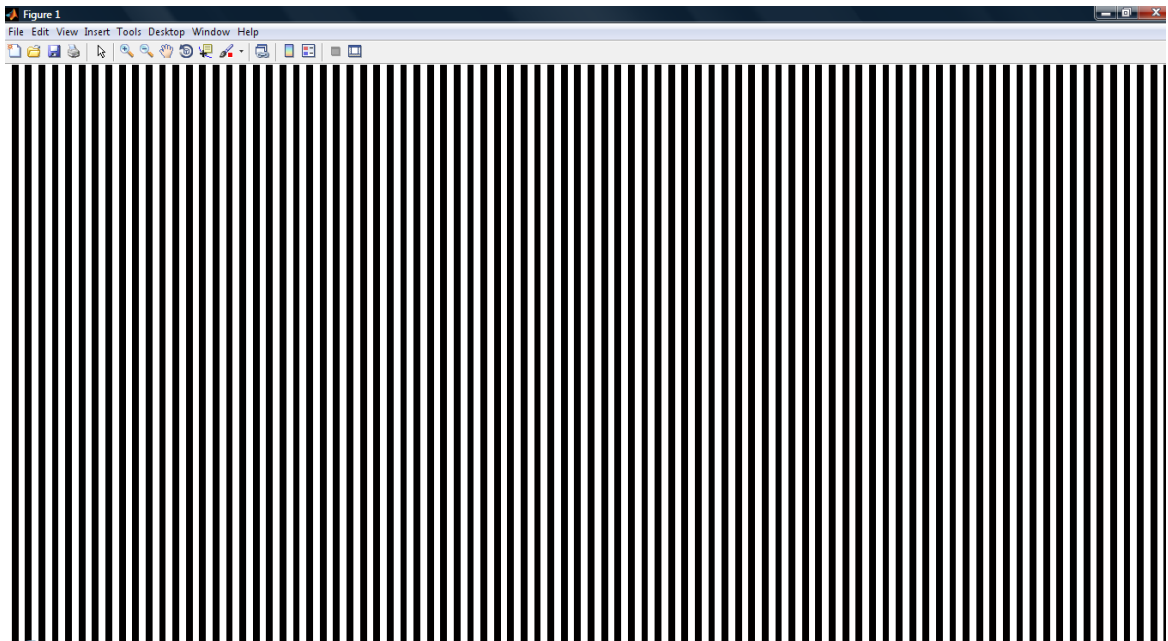
Figura 8. Extensión de la Pantalla



Fuente. Los autores

Una vez ajustados los dos monitores se procede a crear la figura de las franjas, las cuales aparecerán y se ubicaran en la extensión del monitor dos cuando se inicie el software de reconstrucción tridimensional dentro de la codificación de dicho software se ajusta el ancho de la imagen a proyectar así como su color en escala de grises y su intensidad luminosa entre 0 y 255, además cada franja será igual a la siguiente y estarán espaciadas de acuerdo a un paso definido.

Figura 9. Franjas Proyectadas



Fuente. Los autores

Finalmente conectando los dos Video Proyectoros por medio de un cable USB cualquier imagen ubicada dentro la región del monitor 2 (es este caso la figura de las franjas) será proyectada por los dos sistemas de proyección.

3.1.2 Reconocimiento de las cámaras en el entorno MATLAB.

Con la herramienta *ActiveX control* del GUIDE⁷, se puede reconocer el Software (uEye64_38200_WHQL) de las cámaras 1 y 2 desde MATLAB y fácilmente quedan introducidas dentro de la interfaz gráfica como si se tratara de cualquier tipo de imagen. Tal como se aprecia en la siguiente figura.

Figura 10. Visualización de las cámaras dentro de la Interfaz



Fuente. Los autores

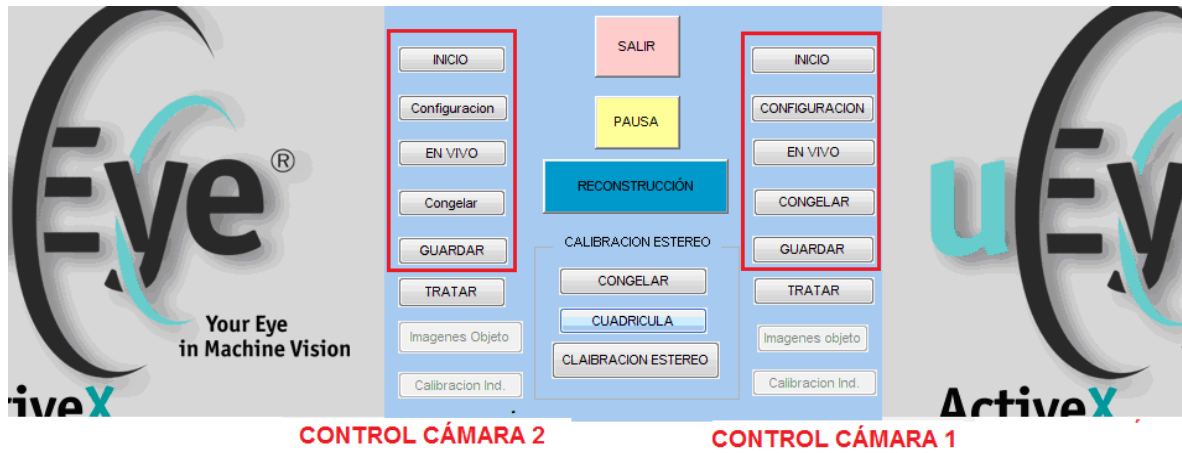
3.1.3 Control de los sistemas de Observación

Dentro del control de las dos cámaras CCD se utilizan cinco operaciones, las cuales corresponden a *5Pushbuttons*⁸ para cada cámara, distribuidos junto a la visualización de cada sistema de observación.

⁷ Ver anexo: Herramientas de Matlab

⁸ Ver anexo: Herramientas de Matlab

Figura 11. Controles de las Cámaras



Fuente. Los autores

Dichos controles son:

3.1.3.1 Inicio

Para la inicialización de las cámaras 1 y 2 se utilizan dos *Pushbutton* cuyo “Tag”⁹ es el de INICIO.

⁹ Ver anexo: Herramientas de Matlab

Figura 12. Control para la Inicialización de las cámaras



Fuente. Los autores

Cada uno de los botones de inicio están configurados para que una vez sean activados bloqueen y habiliten otros controles de las cámaras de acuerdo a la condición en la cual se encuentren, de acuerdo a esto si no se ha activado este control, los controles de en vivo, congelar y guardar deben estar bloqueados, de lo contrario deben quedar habilitados excepto el control del botón “EN VIVO” el cual como se explicará más adelante estará habilitado solamente después de haber congelado la imagen.

Adicionalmente previo a la inicialización de las cámaras, se pide que se seleccione el puerto al cual está conectado cada una de ellas, de esta manera se puede seleccionar cual cámara se desea ver en cualquiera de las dos regiones de visualización.

1.3.1.2 Configuración

Figura 13. Control para configuración de las cámaras



Fuente. Los autores

Este control corresponde a la configuración de las cámaras, como ya se ha explicado anteriormente, las cámaras deben estar configuradas en escala de grises, además no queda habilitado hasta que el control de inicio sea activado.

Una vez activado este control, se abre un cuadro de diálogo, proveniente del software de las cámaras, donde se pueden configurar las características de cada una de ellas.

3.1.3.3 Congelar

Figura 14. Control Congelar Imagen



Fuente. Los autores

En la Figura 3.9 se aprecia este control que permite detener y congelar la imagen que en el momento de ser oprimido esté mostrando la cámara correspondiente, además solamente cuando esté activado se habilita el botón de EN VIVO.

3.1.3.4 En vivo

Una vez se ha activado el control de congelar, para volver a reanudar la captura en vivo de las imágenes se debe oprimir este control.

Figura 15. Control para Reanudación captura de imágenes



Fuente. Los autores

3.1.3.5 Guardar

Con este control se pueden adquirir fotografías y guardarlas, es decir al activarlo, se congela la imagen y se abre el cuadro de diálogo para guardarla en la carpeta especificada y con el nombre que se le dese adjuntar.

Figura 16. Adquirir y Guardar Fotografías.



Fuente. Los autores

Estas imágenes serán guardadas en formato *BMP*¹⁰, el cual es el utilizado en general para el tratamiento de las imágenes dentro de esta investigación.

3.2 DISEÑO DE LOS DISPOSITIVOS DE SUJECIÓN

El diseño de los dispositivos de sujeción de los sistemas de proyección y observación se realizó basado en los requerimientos de espacio y rango de visión requeridos por dicho sistema dual.

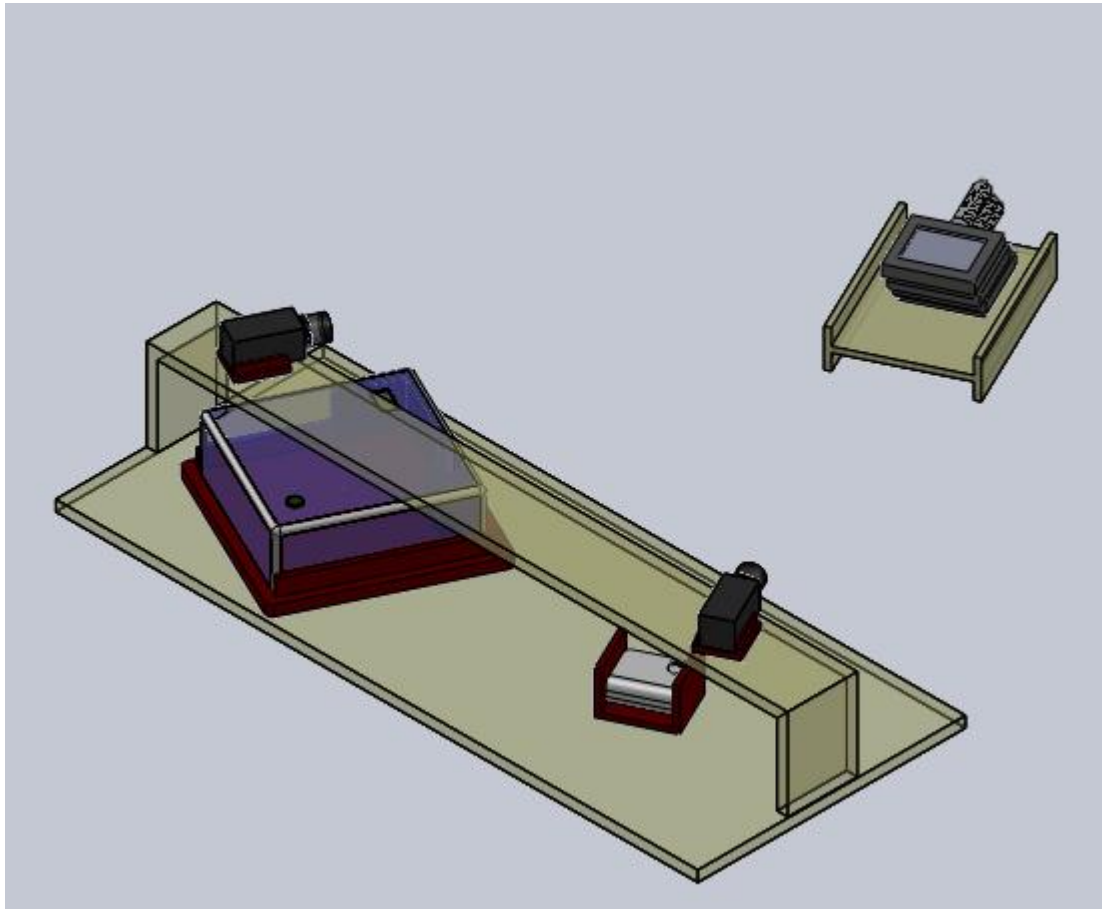
Como se mencionó en el plan de proyecto, el material seleccionado debía tener características especiales como poco peso, dureza, elegancia y economía; debido a esto y según lo previsto, se escogió la madera del futuro (MDF).

¹⁰ BMP: Formato de Mapa de Bits (BitMap)

Si bien el sistema de sujeción no es de gran complejidad, cumple a cabalidad con los requerimientos necesarios, permitiendo ajustar (rotando sobre su propio eje) los elementos del sistema de proyección – observación.

La disposición final del dispositivo de sujeción se puede observar en la siguiente imagen.

Figura 17. Dispositivos de sujeción



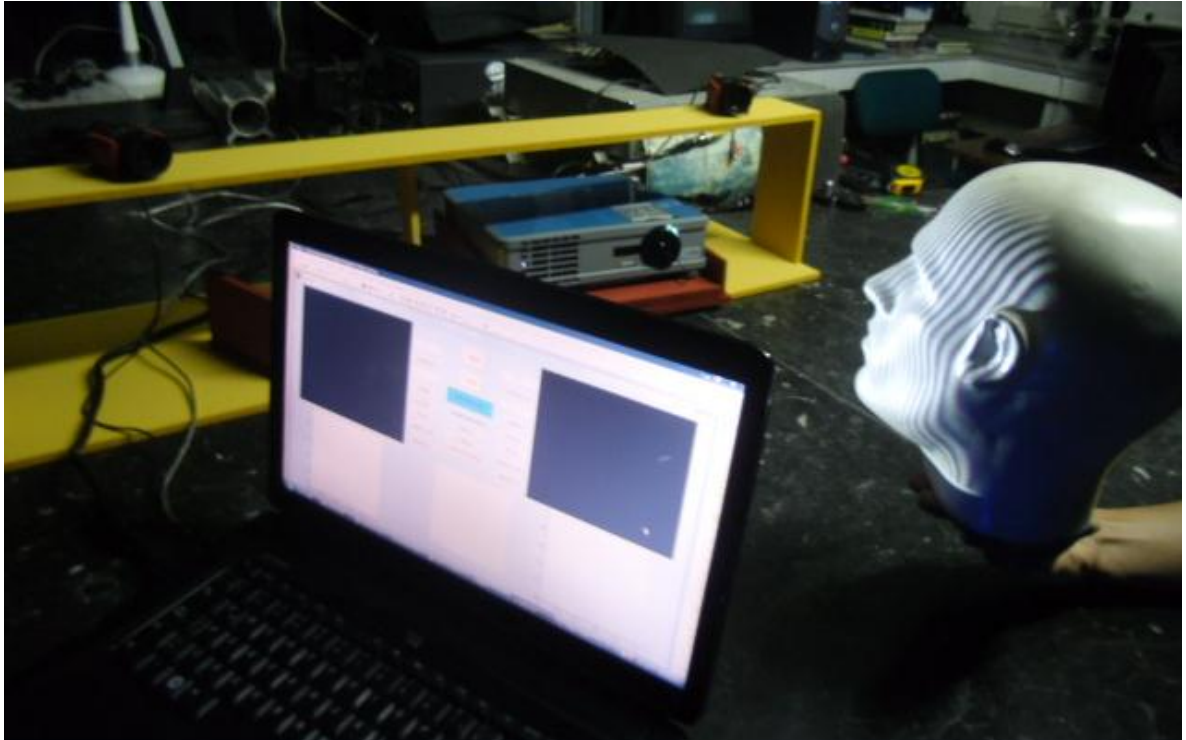
Fuente. Los autores

3.2.1 Posicionamiento preliminar

Una vez inicializado el programa, proyectadas las franjas, inicializadas y configuradas las cámaras, el siguiente paso es un posicionamiento preliminar, el

cual es llevado a cabo por simple inspección y consiste en que las franjas proyectadas, estén direccionadas hacia la zona donde se va a ubicar el objeto a reconstruir, y de igual manera que las cámaras estén apuntando en la dirección correcta y su campo de visión sea el mas óptimo para esta misma zona.

Figura 18. Posicionamiento preliminar del objeto a reconstruir



Fuente.Los autores

Adicionalmente es importante ajustar y verificar que entre los sistemas de proyección y observación exista cierto ángulo, es decir que no se encuentren alineados con el fin de que la deformación de las franjas se pueda apreciar adecuadamente.

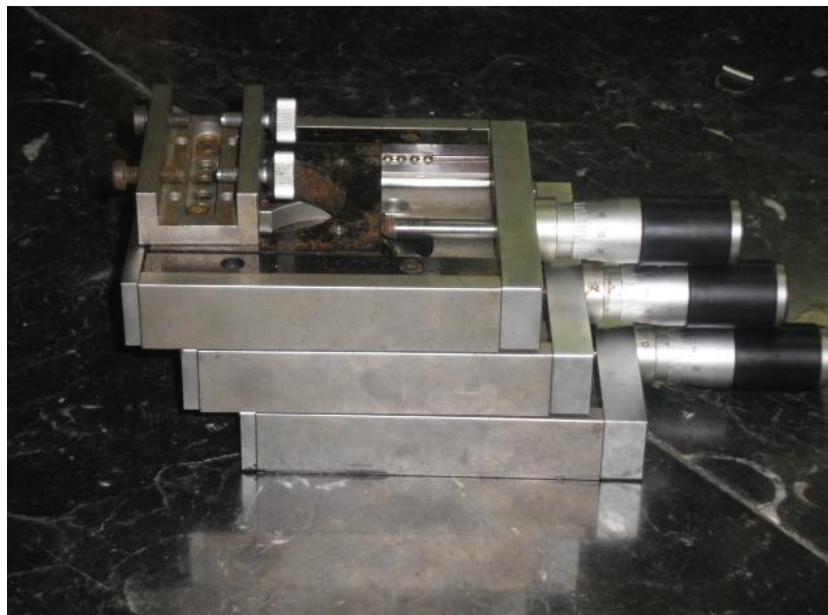
4. CALIBRACIÓN DE LOS SISTEMAS INDEPENDIENTES

La calibración de los sistemas se utiliza para además de verificar el posicionamiento de las cámaras y de los Video Proyectores, para crear referencias las cuales posteriormente serán introducidas dentro del tratamiento de las imágenes del objeto a reconstruir y así poder extraer la información correspondiente a la coordenadas de los ejes X, Y y Z.

Para la calibración independiente se utilizaron los siguientes objetos:

1. Las placas de calibración que son básicamente, tres placas con precisión de 10 micras las cuales nos permiten movernos en lo que definimos como el eje Z. (Figura 4.1)

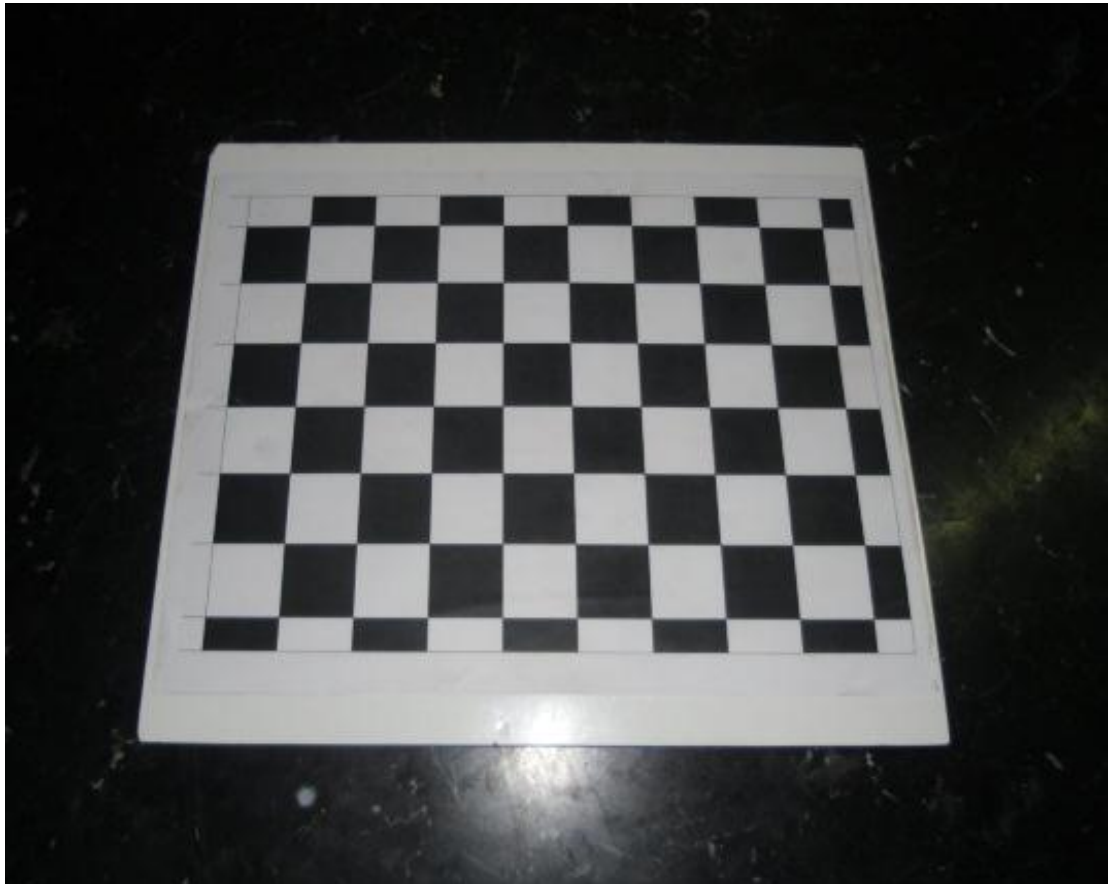
Figura 19. Placas de calibración



Fuente. Los autores

2. Una tabla lo más lisa posible sobre la cual se montó una cuadrícula ajedrezada cuyos cuadrados tienen de lado 18 mm y los cuales servirán de base para la calibración estéreo del sistema. (Figura 4.2)

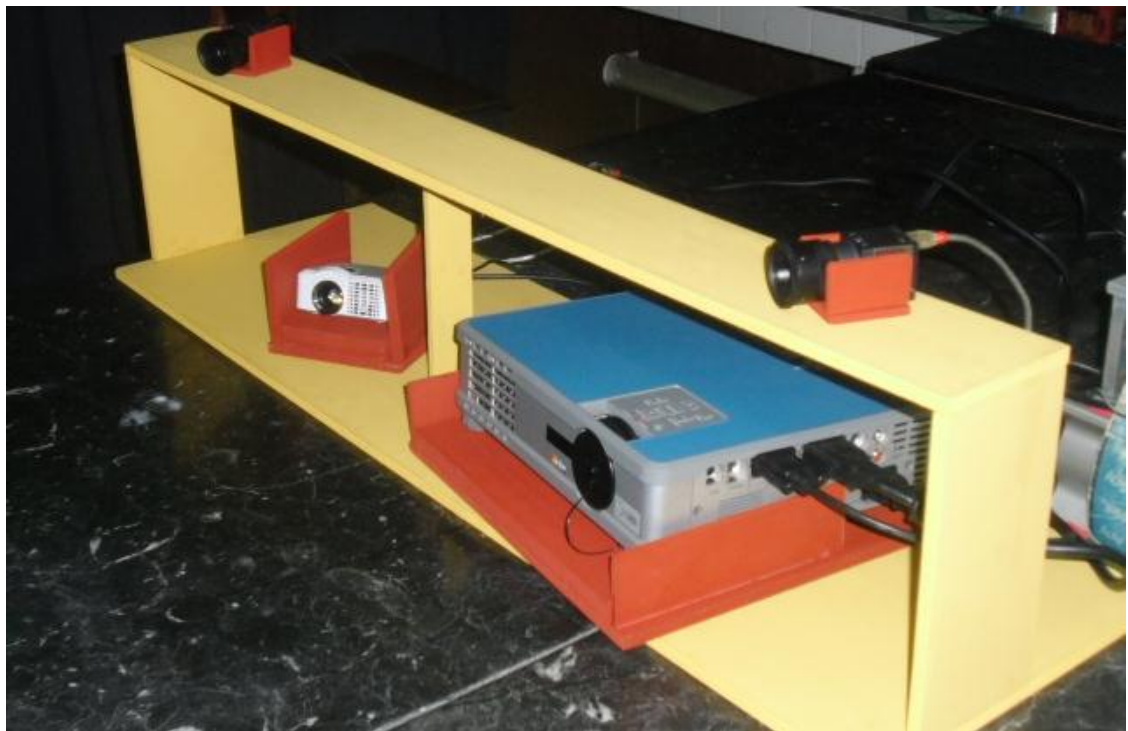
Figura 20. Tabla con cuadrícula ajedrezada



Fuente. Los autores

3. Los soportes de las cámaras y de los Video Proyectores los cuales garantizan que los sistemas de proyección y observación permanezcan estáticos durante todo el proceso de calibración y reconstrucción. (Figura 4.3)

Figura 21. Soportes de cámaras y proyectores



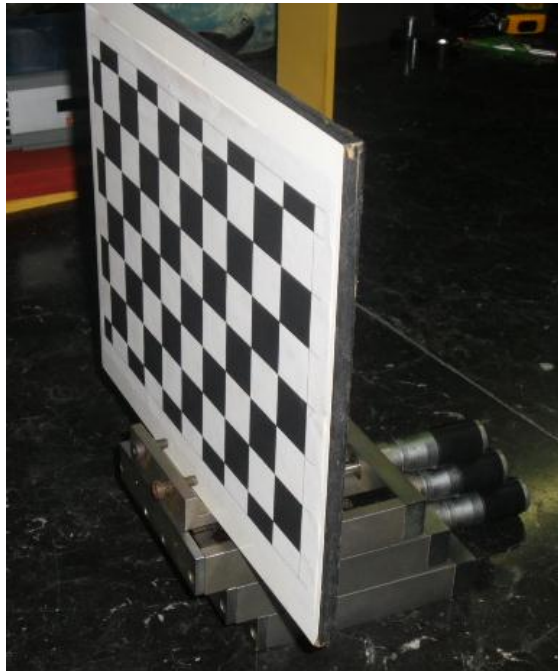
Fuente. Los autores

4.1 PASOS PARA LA CALIBRACIÓN INDEPENDIENTE

Los pasos para la calibración independiente son los siguientes:

1. Se ubica la tabla con la cuadrícula sobre las placas de calibración, de tal manera que quede perpendicular con el eje óptico del sistema de adquisición.

Figura 22. Ubicación de la cuadrícula en las placas de calibración



Fuente. Los autores

2. Dentro de la interfaz del Software, la calibración independiente se muestra como un *PushButton* (Figura 4.5) para cada sistema de observación el cual solamente se habilita si las cámaras han sido inicializadas.

Figura 23. Control para la calibración independiente



Fuente. Los autores

Al activar este botón, se inicia la calibración para el respectivo sistema de proyección-observación.

3. Se adquieren cinco imágenes de la cuadrícula para la posición 1, las cuatro primeras imágenes corresponderán a las franjas proyectadas sobre la tabla con la cuadrícula, cada imagen una a continuación de la otra con un corrimiento de las franjas φ_i el cual ya fue definido, la quinta imagen corresponde a una sola franja proyectada sobre la misma tabla, esta franja se encuentra en la mitad de la figura proyectada.

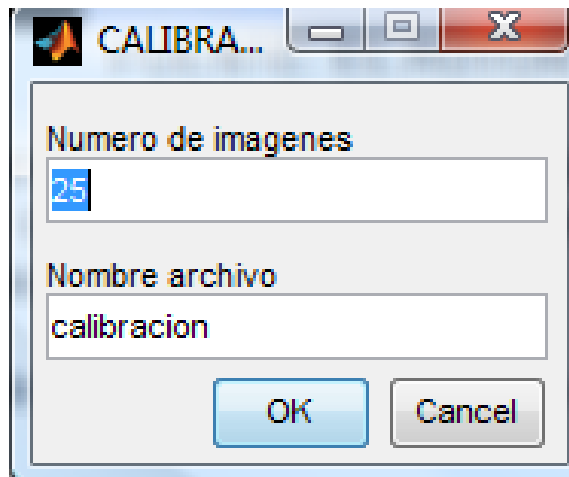
Figura 24. Franja proyectada capturada por las cámaras



Fuente. Los autores

El software automáticamente guarda todas las capturas realizadas, con el nombre que se le desee adjuntar cuando se inicia la calibración en el cuadro de dialogo que se muestra a continuación.

Figura 25. Cuadro de diálogo para nombrar la calibración independiente

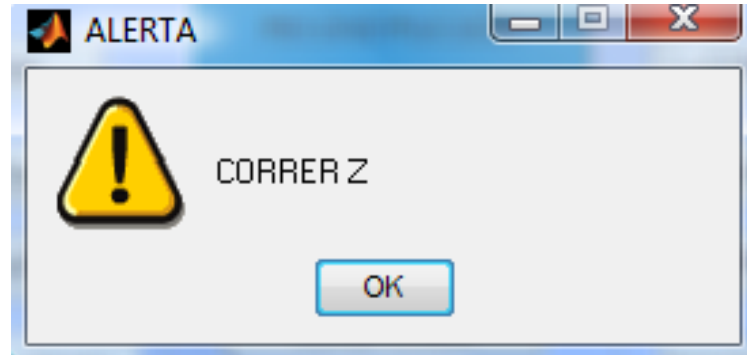


Fuente. Los autores

4. Terminada la captura de las primeras cinco imágenes es necesario mover la tabla con la cuadrícula en la dirección definida como Z (durante esta investigación cada 5 mm) la cual en cada sistema de proyección-observación corresponde a una línea perpendicular a la superficie de la cuadrícula y al lente de la cámara, como se observa en la figura 4.9y el paso anterior se repite una vez se presione el botón de "OK" en la advertencia que aparece (Figura 4.8.), la longitud total de desplazamiento en Z, se asumió de 12 cm lo cual crea un total de 25 posiciones. Es importante resaltar que en la posición 13 (es decir en la mitad de todo el

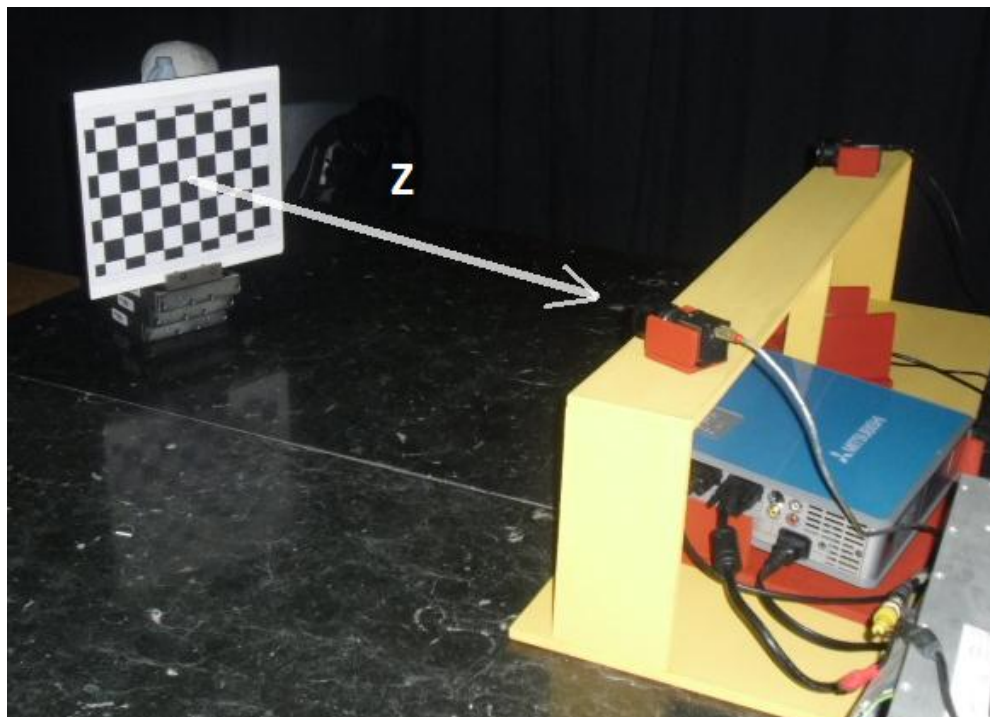
desplazamiento) se encuentra el Z_0 , el cual no es más que una referencia dentro de lo que se denomina tratamiento de las imágenes.

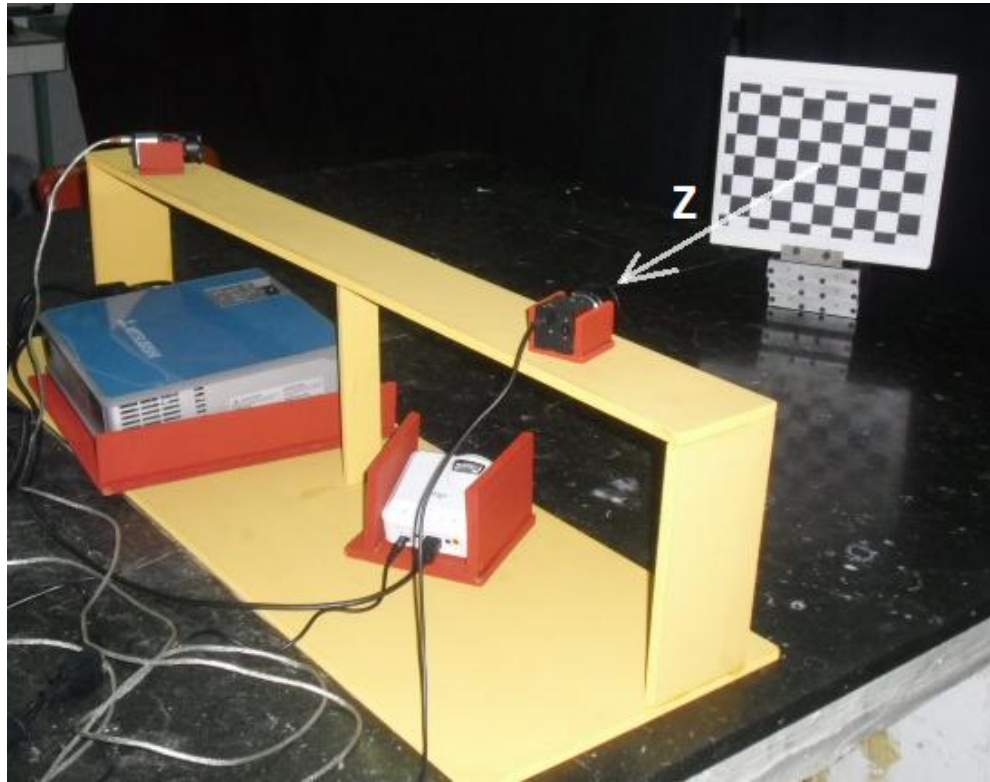
Figura 26. Advertencia para correr la cuadrícula en dirección Z



Fuente.Los autores

Figura 27. Dirección del eje Z





Fuente. Los autores

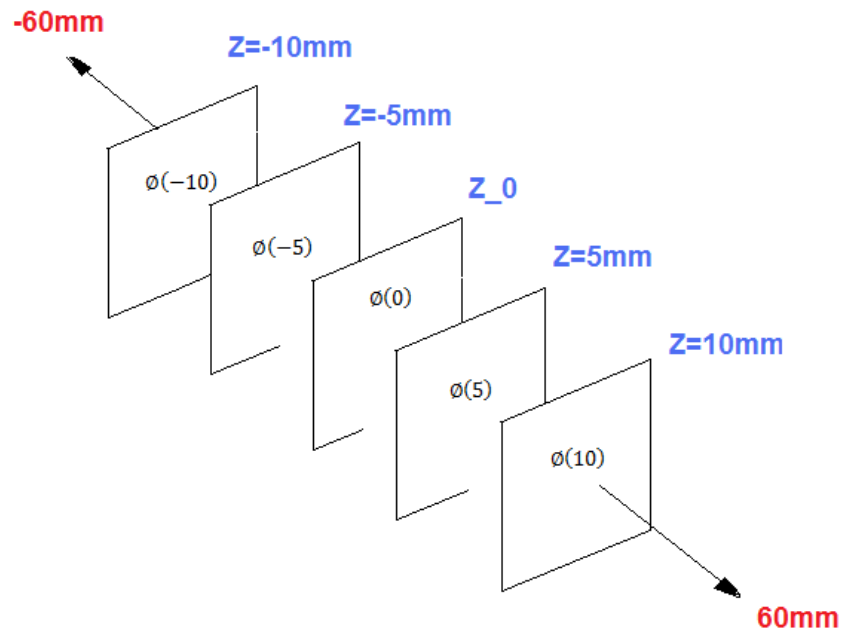
Finalizado el anterior proceso el software automáticamente guarda las *Matrices de calibración* con las cuales se realiza el cálculo de la función de fase.

4.1.1 Matrices de Calibración.

Las matrices de calibración son datos extraídos de cada una de las veinticinco posiciones en las cuales fueron capturadas cinco imágenes, con estos datos realiza el cálculo de la fase para cada una de dichas posiciones (ver figura 4.10) y son la base para el hallar la función de fase, la cual (como se indicará en la sección 3.4.3 *Cálculo de la fase*) es una función cuadrática que recoge todos estos puntos y permite a partir de ella obtener información de la altura de cualquier

punto ubicado en ya sea dentro de la región de calibración (12 centímetros) o por fuera de la misma.

Figura 28. Posiciones para el cálculo de la fase



Fuente. Los autores.

De esta manera se da por terminada la calibración independiente de un sistema de proyección-observación, para el otro sistema se repiten todos los pasos anteriores.

Adicionalmente se realiza una captura adicional de la cuadrícula puesta en el Z_0 de cada sistema independiente, esta imagen será utilizada durante el proceso de la *Calibración Estéreo del Sistema*.

5. CALIBRACIÓN ESTÉREO DEL SISTEMA

La calibración estéreo es el método por el cual se realiza una caracterización de un sistema dual de cámaras para diferentes procesos de extracción de información; para nuestro caso, poseemos dos sistemas de proyección y observación, en los cuales, se realiza la calibración de cada uno de los sistemas independientes y posteriormente, para obtener la información 3D con alta precisión, es necesario la realización del proceso que a continuación se describirá.

5.1 EXTRACCIÓN DE LAS ESQUINAS, CALIBRACIÓN Y HERRAMIENTAS ADICIONALES.

Como primera medida, debemos resaltar que la realización de la calibración estéreo del sistema de adquisición de información que se lleva a cabo en este proyecto, es software libre, denominado CAMERA CALIBRATION TOOLBOX el cual está disponible en la red.¹¹

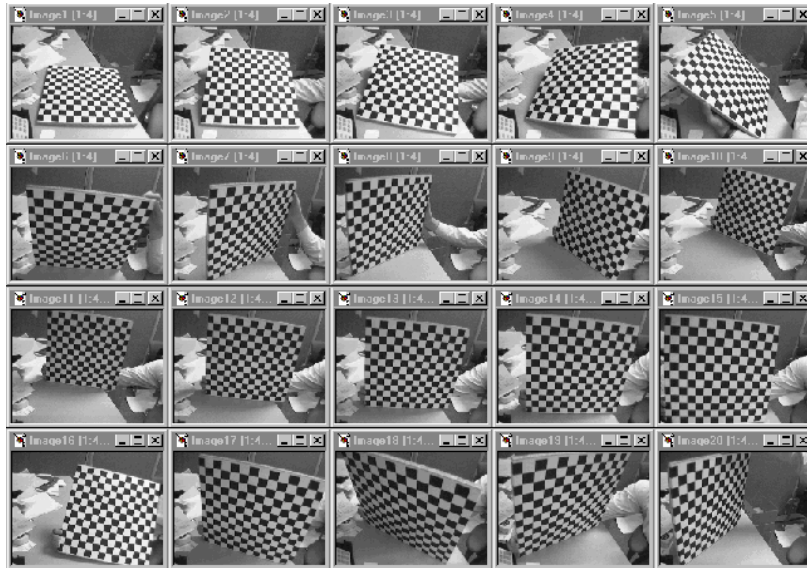
El objetivo principal de este procedimiento es el de adquirir información clave para realizar el empalme de las dos figuras de la reconstrucción tridimensional que se obtienen por medio de cada uno de los sistemas de proyección-observación, en esta primera sección de la calibración estéreo se busca obtener los parámetros intrínsecos y extrínsecos de las dos cámaras de la manera como describe a continuación.

El primer paso consiste en la extracción de imágenes de un tablero plano, en el cual se diseñó una cuadrícula de 18mm*18mm, el cual será nuestro punto de referencia para la toma de datos posterior.

¹¹ Tomado de:<http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html>

El siguiente paso consiste en realizar la captura de las imágenes, la cual será un total de quince imágenes por cámara, variando la posición de cada una de dichas imágenes, como se observa en la siguiente figura.

Figura 29. Adquisición de imágenes de la cuadrícula



Fuente. Disponible en [6]

La captura de las imágenes debe ser simultánea es decir corresponde a la misma posición de la tabla vista desde cada una de las dos cámaras, para esto en la interfaz del Software de esta investigación se utiliza el botón de “CONGELAR” que se encuentra dentro del título de Calibración estéreo como se muestra en la siguiente figura.

Figura 30. Control para capturar imágenes simultáneas



Fuente.Los autores

Una vez tomadas y guardadas todas las imágenes se puede invocar al software de CAMERA CALIBRATION TOOLBOX desde la interfaz del programa de nuestra investigación, simplemente al oprimir el botón de “CUADRICULA”.

Figura 31. Control para invocar el Software Camera Calibration Toolbox

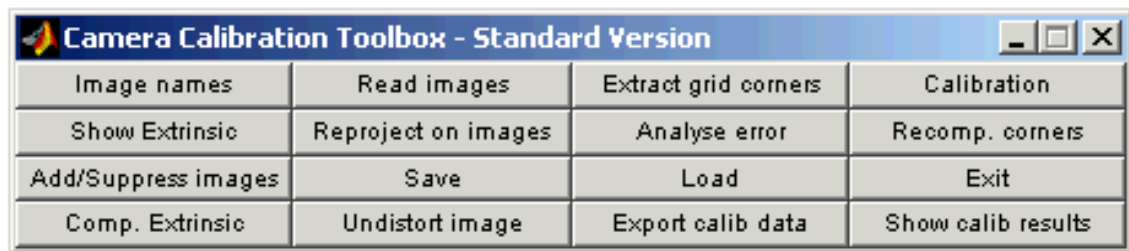


Fuente.Los autore

5.1.1 Lectura de las imágenes

De acuerdo al Software CAMERA CALIBRATION TOOLBOX y siguiendo su metodología de uso, se debe realizar la lectura de dichas imágenes por el software de calibración, llamando a dichas imágenes por medio del botón *Read Images* que podemos observar en la figura 5.4.

Figura 32. Interfaz del software *Camera Calibration Toolbox*

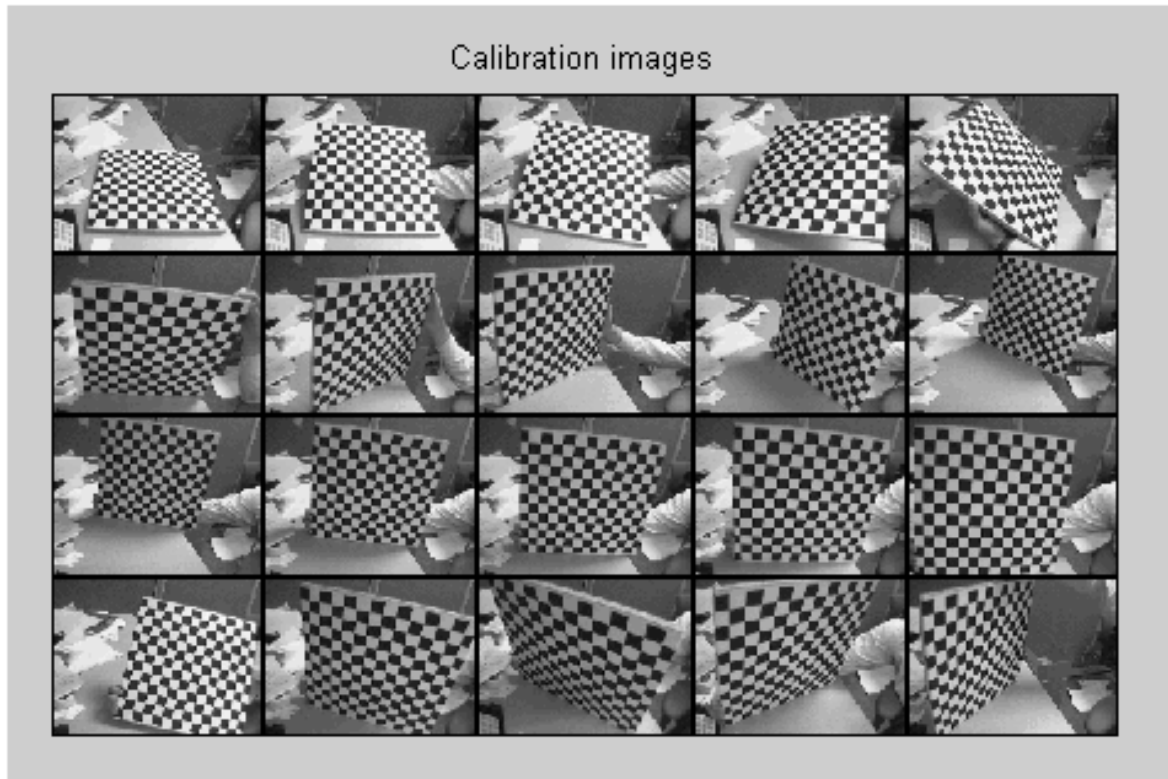


Fuente. Disponible en [6]

Las imágenes adquiridas serán almacenadas por el software y deberán ser guardadas con un formato de imagen (BMP).

En el momento en que la carga de las imágenes esté completa, se podrán observar en mosaico estas imágenes como se ve en la figura 5.5.

Figura 33. Imágenes para la calibración en el ambiente Matlab



Fuente.Disponible en [6]

5.1.2 Extracción de las esquinas de la cuadrícula

La extracción de las esquinas de la cuadrícula se realiza con el fin de tener un punto de referencia fijo en todas las imágenes tomadas, el cual, servirá para la posterior realización del proceso de calibración dual.

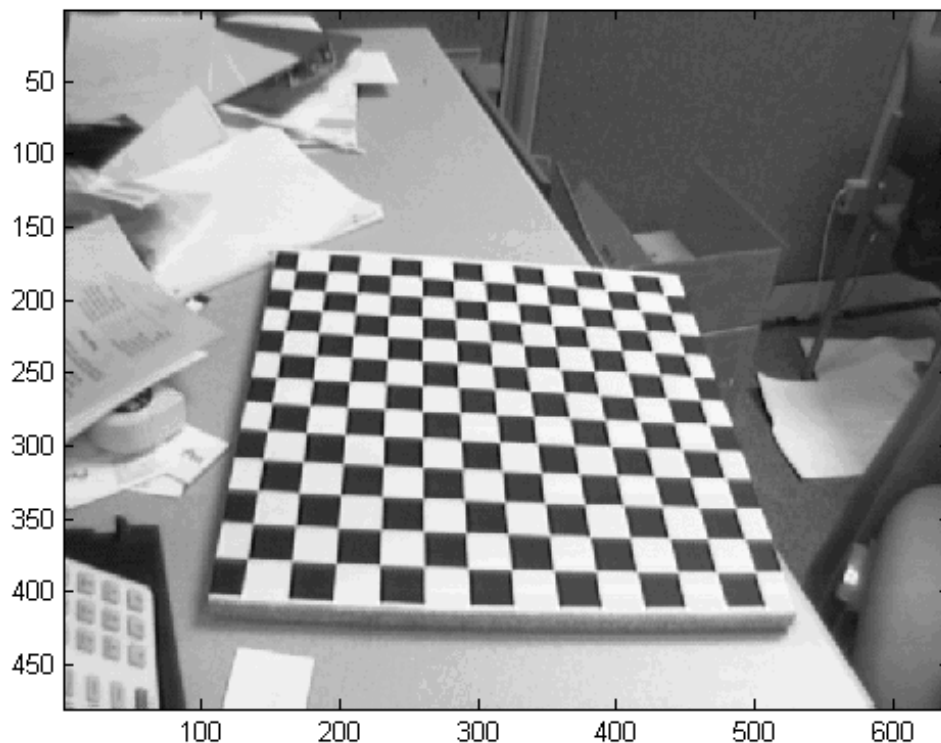
Para ello hay que dirigirse al CCT¹² y presionar el botón *ExtractGridCorners* en el cual se deberá realizar la selección de las imágenes, después de tener las imágenes, se deberá seleccionar el tamaño de la ventana del buscador de esquinas por defecto.

En la primera calibración de la imagen, se deberán seleccionar los puntos de referencia anteriormente mencionados.

¹²Camera Calibration Toolbox

Figura 34. Selección de puntos para el área común

Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



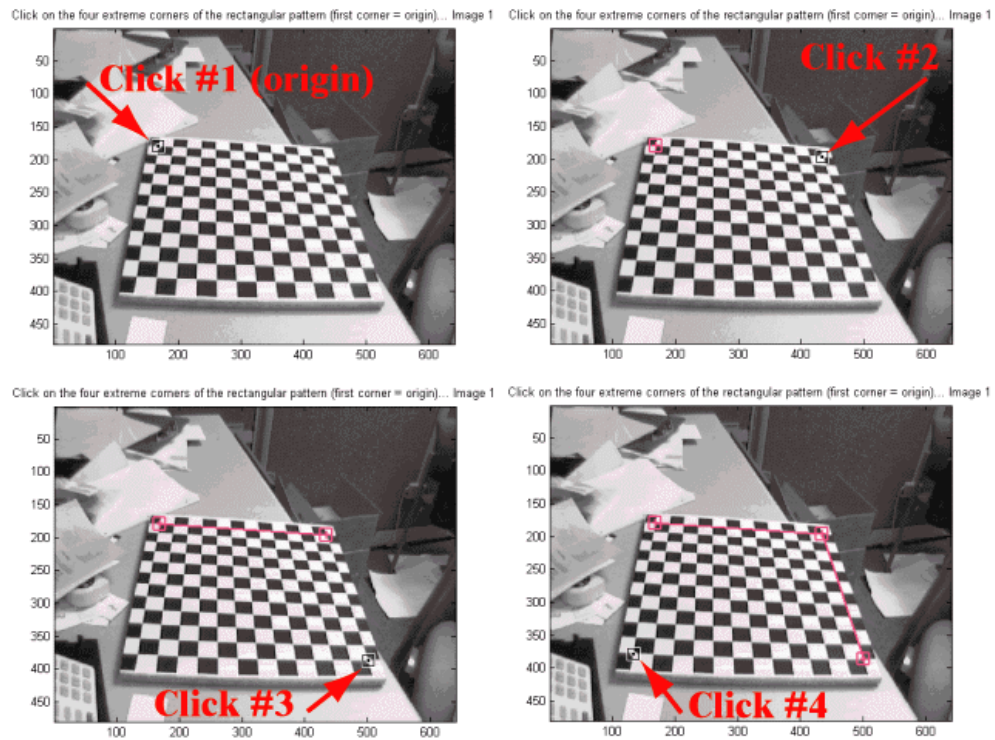
Fuente.Disponible en [6]

El software pide que se seleccionen las cuatro esquinas para la generación de un rectángulo de referencia (figura 5.6.). Para la selección de dichas esquinas, se requiere de gran exactitud, debido a que de no hacerse de la manera correcta, podría realizarse dicha calibración de manera defectuosa, llevando a la no lectura de los puntos seleccionados.

Se debe tener en cuenta que el primer punto seleccionado, será el punto de origen (ver figura 5.7.) del marco de referencia adjunto a la red, los demás puntos, podrán ser escogidos de manera aleatoria, esta norma se debe tener en cuenta, ya que necesitamos realizar la calibración de más de una cámara externa (las cuales poseen posiciones relativas una con respecto de la otra). Cuando se trabaja con varias cámaras, el mismo patrón de cuadrícula de marco de referencia debe ser siempre elegido para las imágenes de una cámara diferente (es decir, los puntos

de la red tienen que corresponderse a través de los puntos de vista de la otra cámara).

Figura 35. Secuencia de selección de puntos

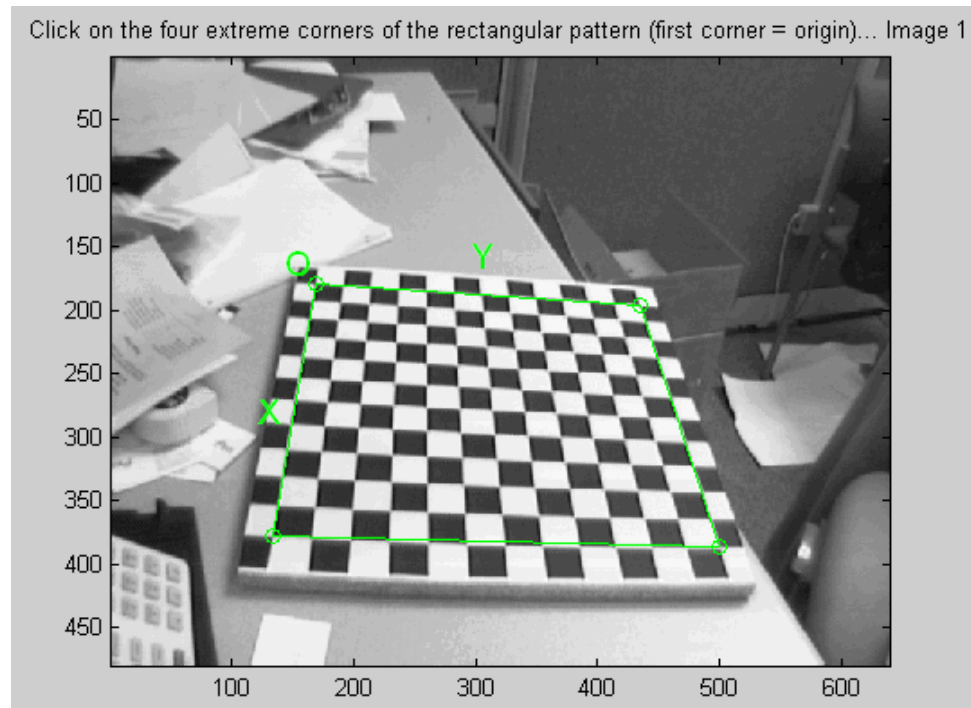


Fuente.Disponible en [6]

Después de la selección de los puntos en la cuadrícula, se puede visualizar un esquema del rectángulo resultante, como observamos en la figura 5.8.

Al momento de completar el rectángulo base, el programa pedirá la introducción de las medidas de los elementos diferenciales con los cuales el programa trabajará, estos elementos serán entonces cada uno de los recuadros de la cuadrícula, los cuales en este caso poseen una longitud de 18 mm.

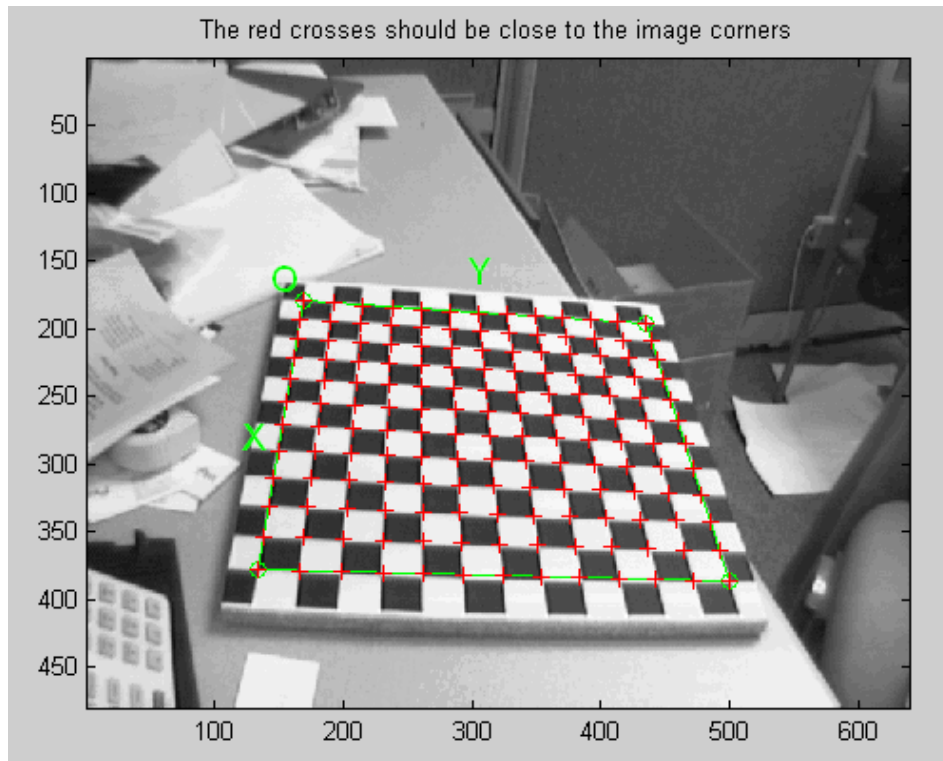
Figura 36. Cuadro común seleccionado



Fuente. Disponible en [6]

De igual manera, si se quisiera no ingresar la longitud de los elementos diferenciales, el programa tiene la capacidad de captar todas las esquinas de los elementos que han sido seleccionados, como lo muestra la siguiente figura, en la cual se puede observar la identificación de las esquinas de todos los elementos del área de trabajo, que de ahora en adelante será la referencia para las demás imágenes.

Figura 37. Selección de los elementos diferenciales del área de trabajo



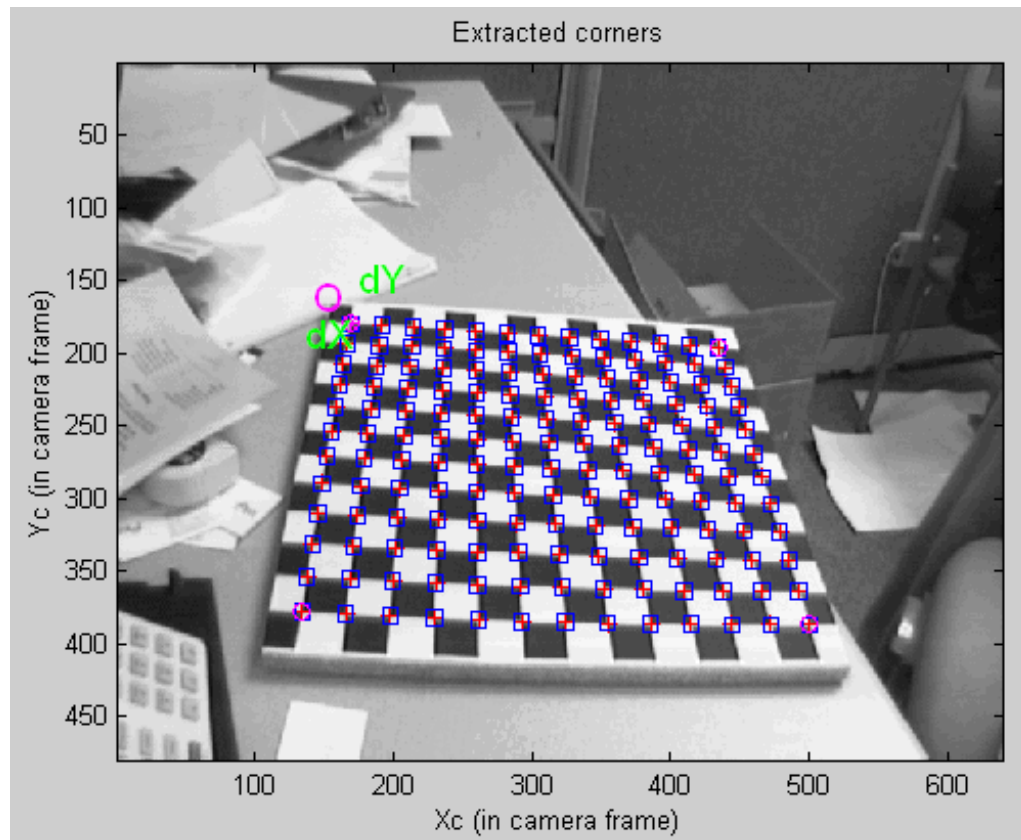
Fuente. Disponible en [6]

Al observar detenidamente la figura 5.9, las cruces rojas deberán estar sobre las esquinas de cada uno de los cuadros de nuestra área común, si éste es el caso, se puede proseguir con el proceso de calibración, de lo contrario se deberá realizar un proceso para ayudar a que la selección de estas esquinas sea la más aproximada, esto lo realiza el software por medio de una estimación para el coeficiente de distorsión radial, cabe resaltar que éste es un procedimiento interno del software para optimizar la obtención de los datos necesarios, por tal motivo, no se profundizará en dicho tema.

La figura que se muestra a continuación muestra los puntos que el software identificó como esquinas de los recuadros internos del área común, los pequeños recuadros azules nos indican los límites del buscador de esquinas que posee el software.

El software extrae la información de las esquinas (figura 5.10) con una precisión de aproximadamente 0,1 pixel.

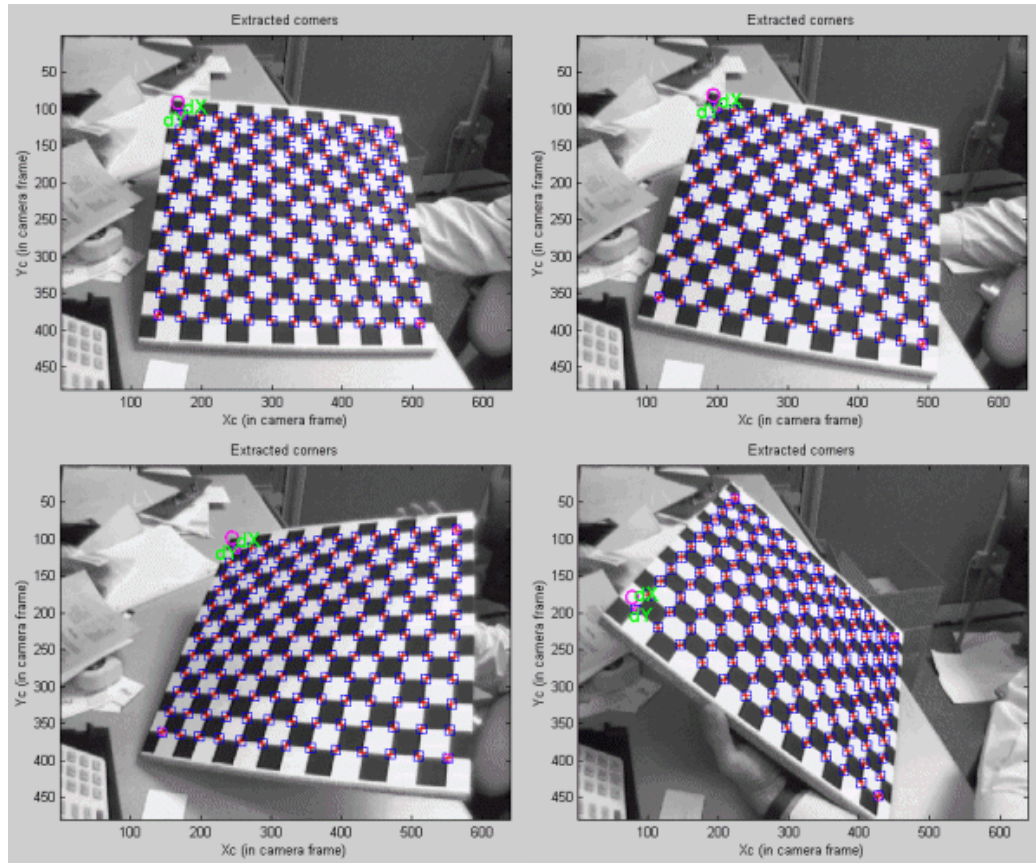
Figura 38. Extracción de las esquinas



Fuente. Disponible en [6]

Este procedimiento se deberá realizar para las quince imágenes tomadas de cada una de las cámaras del sistema dual.

Figura 39. Extracción de las esquinas para todas las imágenes

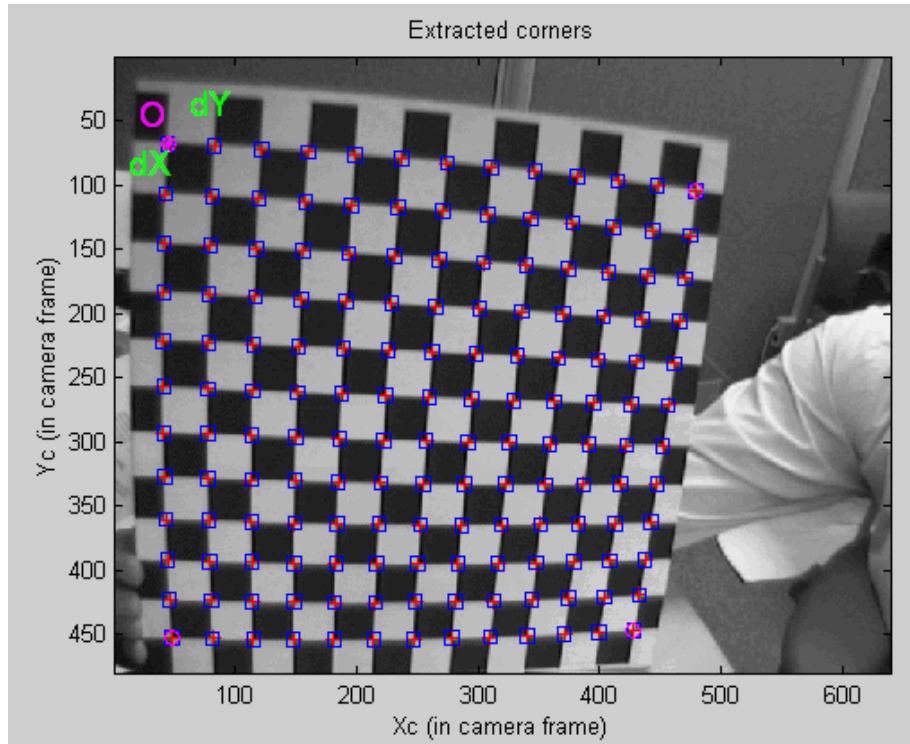


Fuente. Disponible en [6]

Si se observa detenidamente la figura 5.11, se aprecia que las longitudes de cada uno de los cuadros se ha mantenido igual a la longitud inicial, cabe resaltar que algunas de las imágenes no serán totalmente precisas, aun con la ayuda del software, como lo podremos observar en la figura 5.12, en este caso, la distorsión de la imagen es tal que el programa no puede determinar con precisión las esquinas de los cuadros internos, en este caso, la codificación del programa, tiene un parámetro denominado, coeficiente de distorsión (Kc), el cual, es posible modificar hasta obtener un perfil más preciso en la selección de las esquinas del área común. El proceso de modificación del Kc se puede realizar las veces que se desee, hasta llegar a un punto en el que la distorsión generada por la imagen sea mínima, esto quiere decir, que el proceso de debe realizar hasta que las esquinas

sean seleccionadas con la mayor precisión posible. Cabe resaltar que este proceso de modificación del K_c no afectará al programa de calibración principal.

Figura 40. Calculo de los elementos diferenciales y coeficiente de distorsión



Fuente. Disponible en [6]

Después de la recopilación de la información de las esquinas, el programa genera un archivo, en el cual se guarda toda la información adquirida (coordenadas de la imagen, coordenadas correspondientes de la cuadrícula, tamaño de la malla).

En determinado caso de que exista gran distorsión y no sea corregida, el programa puede que no sea capaz de realizar la lectura de la cantidad de elementos diferenciales X e Y , en este caso, el software da la posibilidad de realizar el ingreso manual de la cantidad de elementos diferenciales que se han tomado, para nuestro caso, no debería haber ningún problema, debido a que nuestros cuadros internos son lo suficientemente grandes como para poder

contarlos, este proceso sería tedioso, si existiera gran cantidad de cuadros en el área común.

5.2 PASO PRINCIPAL DE CALIBRACIÓN

Después del proceso de la extracción de las esquinas, es necesario dirigirse a la interfaz principal del software y presionar el botón de calibración, el cual comenzará el procedimiento, basado en dos etapas: la inicialización y la optimización no lineal.

En la inicialización, se calcula una solución de forma cerrada para la calibración de los parámetros, basada, en la no inclusión de distorsiones de la lente.

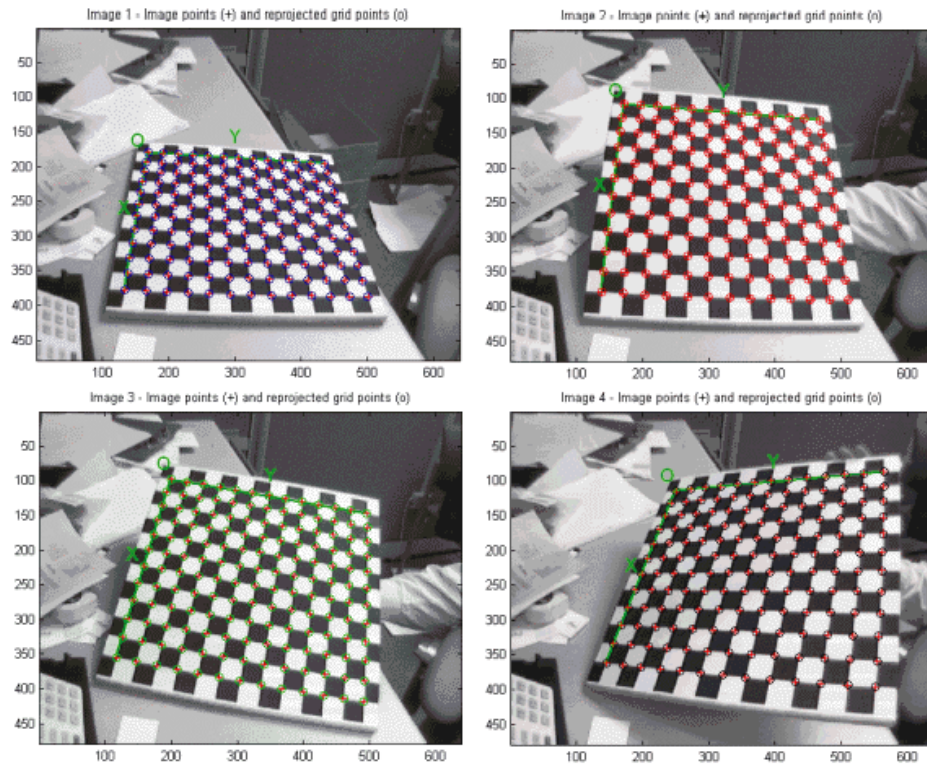
En la optimización no lineal, se minimiza el error de re-proyección total sobre todos los parámetros de calibración, esta optimización se lleva a cabo por el descenso del gradiente iterativo con una explícita forma de cálculo por medio de una matriz Jacobiana¹³.

Los parámetros de calibración se almacenan en una serie de variables.

En la siguiente figura se puede observar algunas de las imágenes que ya han sido tratadas, en las cuales, las esquinas detectadas son señaladas por medio de cruces rojas y las esquinas reproyectadas son señaladas por medio de círculos de diferentes colores para cada una de las imágenes.

¹³ Ver <<http://proton.ucting.udg.mx/robotica/r166/r87/r87.htm>>

Figura 41. Imágenes tratadas

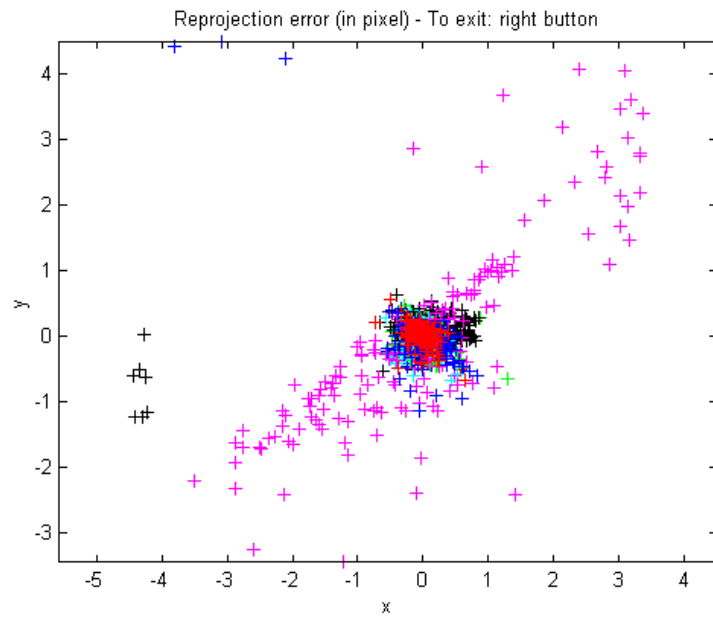


Fuente. Disponible en [6]

Existe, además, una forma de mostrar el error que se produjo en la re-proyección de las esquinas, y podemos mostrarlo en la forma de dispersión de puntos.

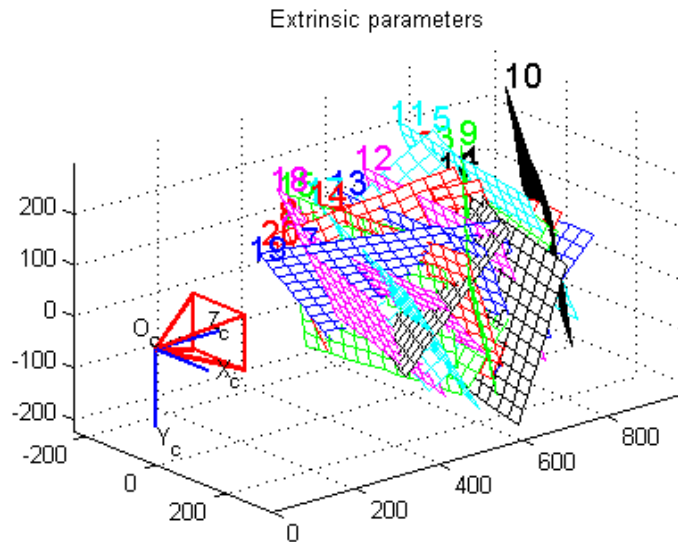
El software, además, da la posibilidad de mostrar los parámetros extrínsecos (posiciones relativas de la cuadrícula con respecto de la cámara). La figura 5.14 muestra todas las posiciones que se tomaron de la cuadrícula en una sola imagen, este tipo de imagen 3D se denomina, tipo parcela.

Figura 42. Error de reproyección



Fuente.Disponible en [6]

Figura 43. Parámetros extrínsecos

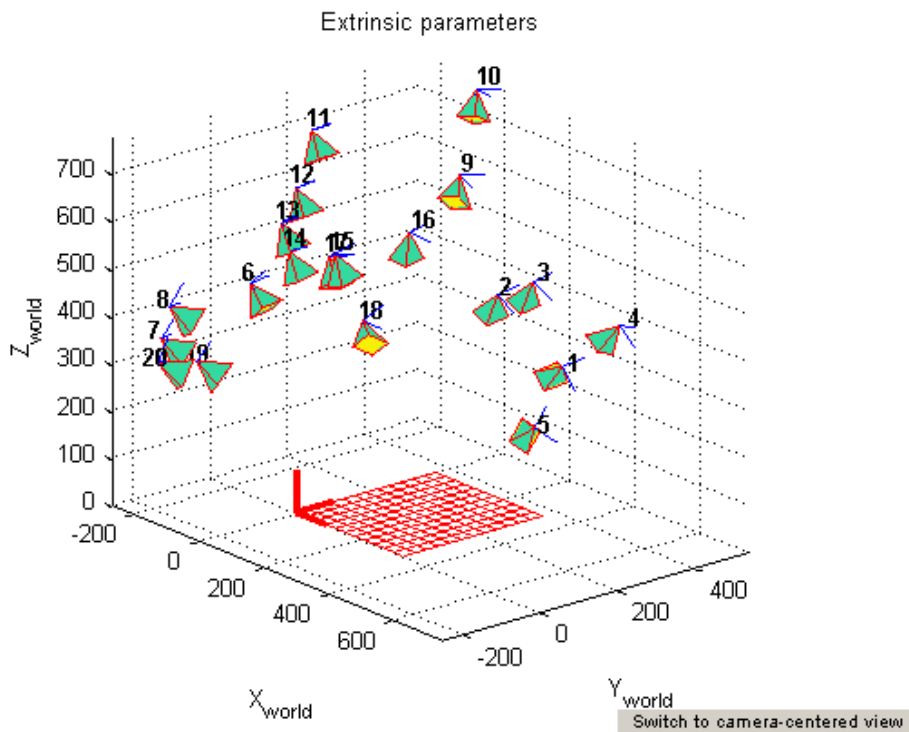


Fuente.Disponible en [6]

En la figura 5.15 de parámetros extrínsecos, el marco (O_c , X_c , Y_c , Z_c), indica el punto de referencia de la cámara y la pirámide que se forma en color rojo, es el campo de visión efectivo de nuestra cámara.

En la siguiente figura se puede observar cada una de las posiciones y orientaciones que la cámara tomó, representados por medio de pirámides verdes.

Figura 44. Orientación de la cámara

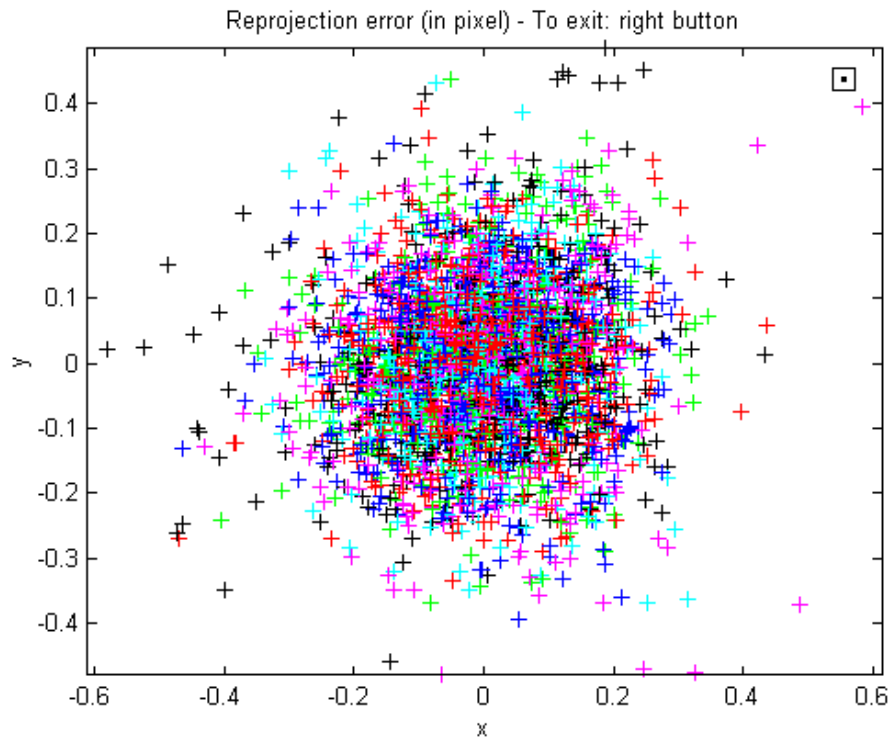


Fuente. Disponible en [6]

Observando la imagen generada del error de reproyección, de manera inmediata se hace notable que existe una gran dispersión en los puntos re-proyectados, eso indica que existe aún una gran distorsión en la extracción de las esquinas, por supuesto, esto generaría una mala calibración, a pesar de que ya se habló del uso del coeficiente de distorsión, probablemente éste no se usó para todas las imágenes, el software, dará la opción de corregir las distorsiones residuales, de una manera manual, para luego realizar el proceso de optimización no lineal y

obtener un resultado en la gráfica de error de re-proyección más satisfactorio y poder así seguir con el proceso de calibración.

Figura 45. Error de reproyección modificado

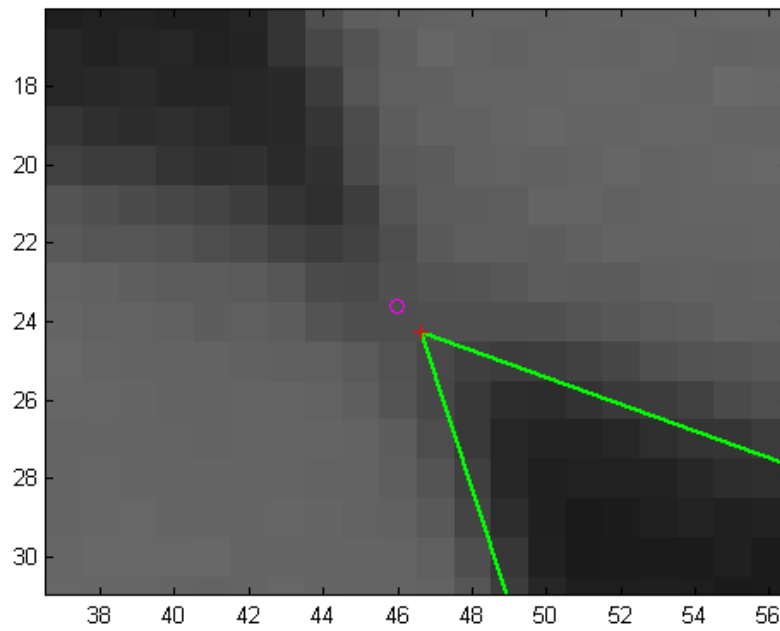


Fuente. Disponible en [6]

Existe una herramienta llamada, inspección de error (esta herramienta es llamada haciendo click en el recuadro pequeño ubicado en la parte superior derecha de la imagen), la cual es de alta importancia, debido a que existe la posibilidad de que se genere alguna mala extracción de cualquier esquina, con el uso de esta herramienta, se puede volver a calcular el tamaño de las esquinas realizando una ampliación o disminución del campo de observación y realizando la selección de dicho punto de manera manual, para de esa manera, obtener la información exacta.

Figura 46. Mala selección de una esquina

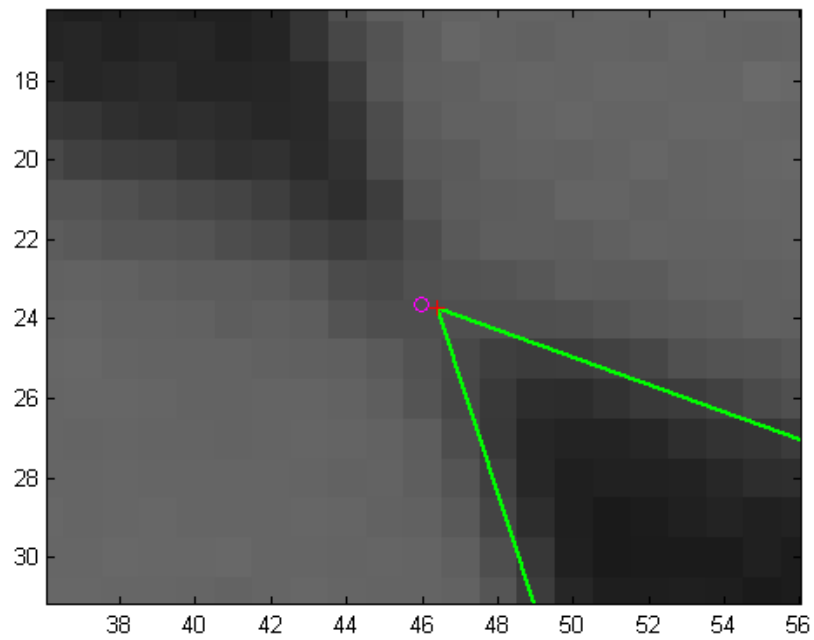
Image 18 - Image points (+) and reprojected grid points (o)



Fuente. Disponible en [6]

Figura 47. Modificación de la esquina seleccionada

Image 18 - Image points (+) and reprojected grid points (o)



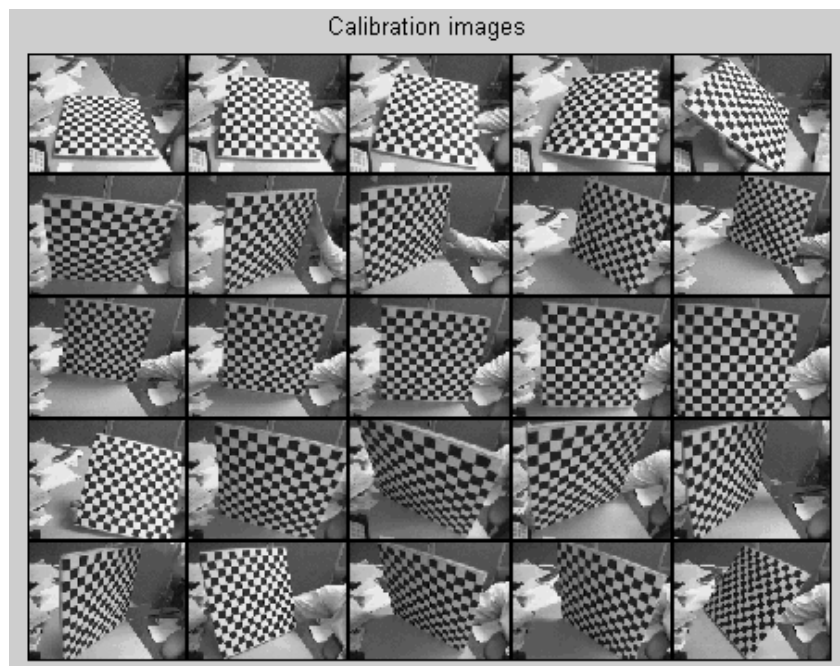
Fuente. Disponible en [6]

En las figuras anteriores se observa el uso de la herramienta de inspección del error, antes y después de su aplicación, y se aprecia que el punto reproyectado se acerca más al punto inicialmente seleccionado disminuyendo así el error generado.

El proceso de calibración (extracción de las esquinas del área común) deberá ser realizado para cada una de las 20 imágenes que se tomaron de cada uno de los sistemas de observación.

Después de la realización de todo este procedimiento, el programa guarda automáticamente estas imágenes y las muestra en mosaico como se ve en la figura 5.20.

Figura 48. Almacenaje de las imágenes de calibración

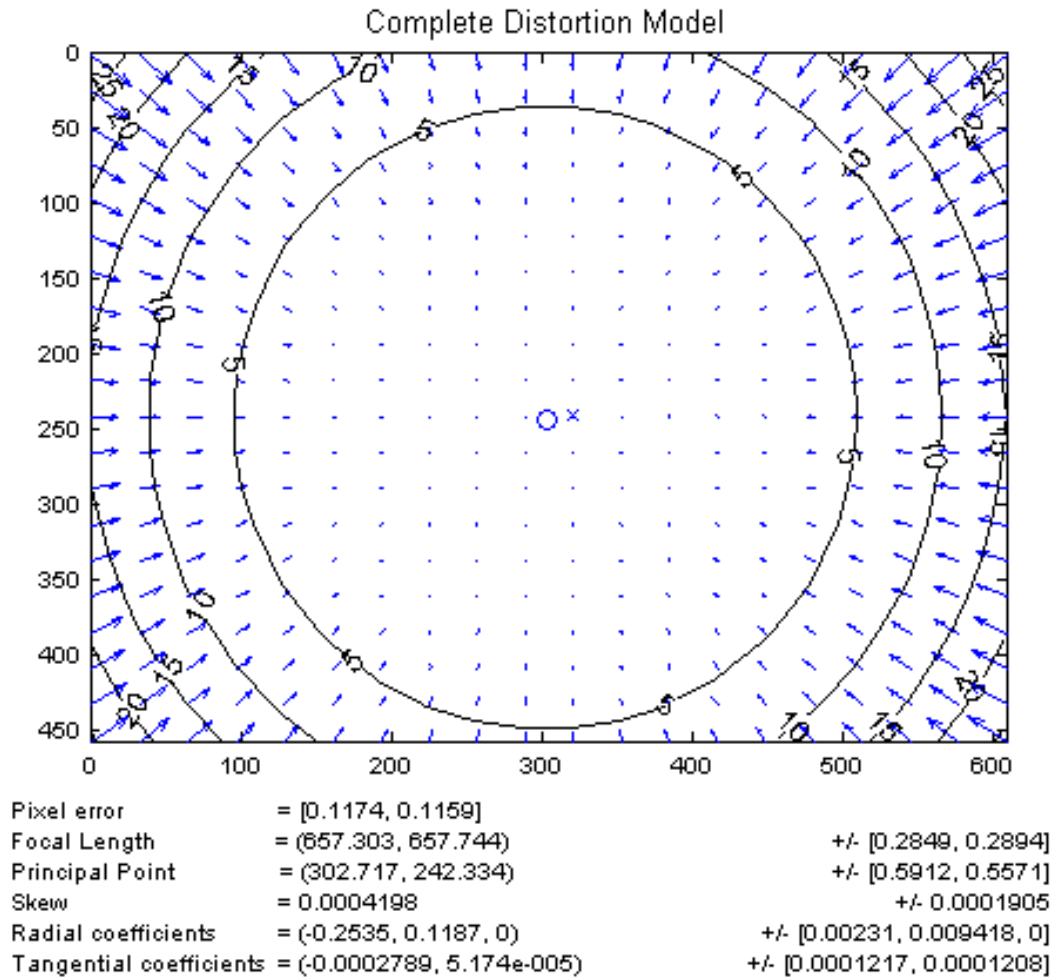


Fuente. Disponible en [6]

Después de todo el proceso anterior, mediante la re-calibración del sistema (por medio de las ayudas anteriormente comentadas), se obtienen las imágenes del modelo de distorsión. En la figura 5.21 del modelo completo de distorsión (radial y tangencial), observamos el impacto de dicho modelo sobre cada uno de los

pixeles de la imagen, cada una de las regiones indica el desplazamiento efectivo de un pixel introducido por la distorsión de la lente.

Figura 49. Modelo de distorsión completo

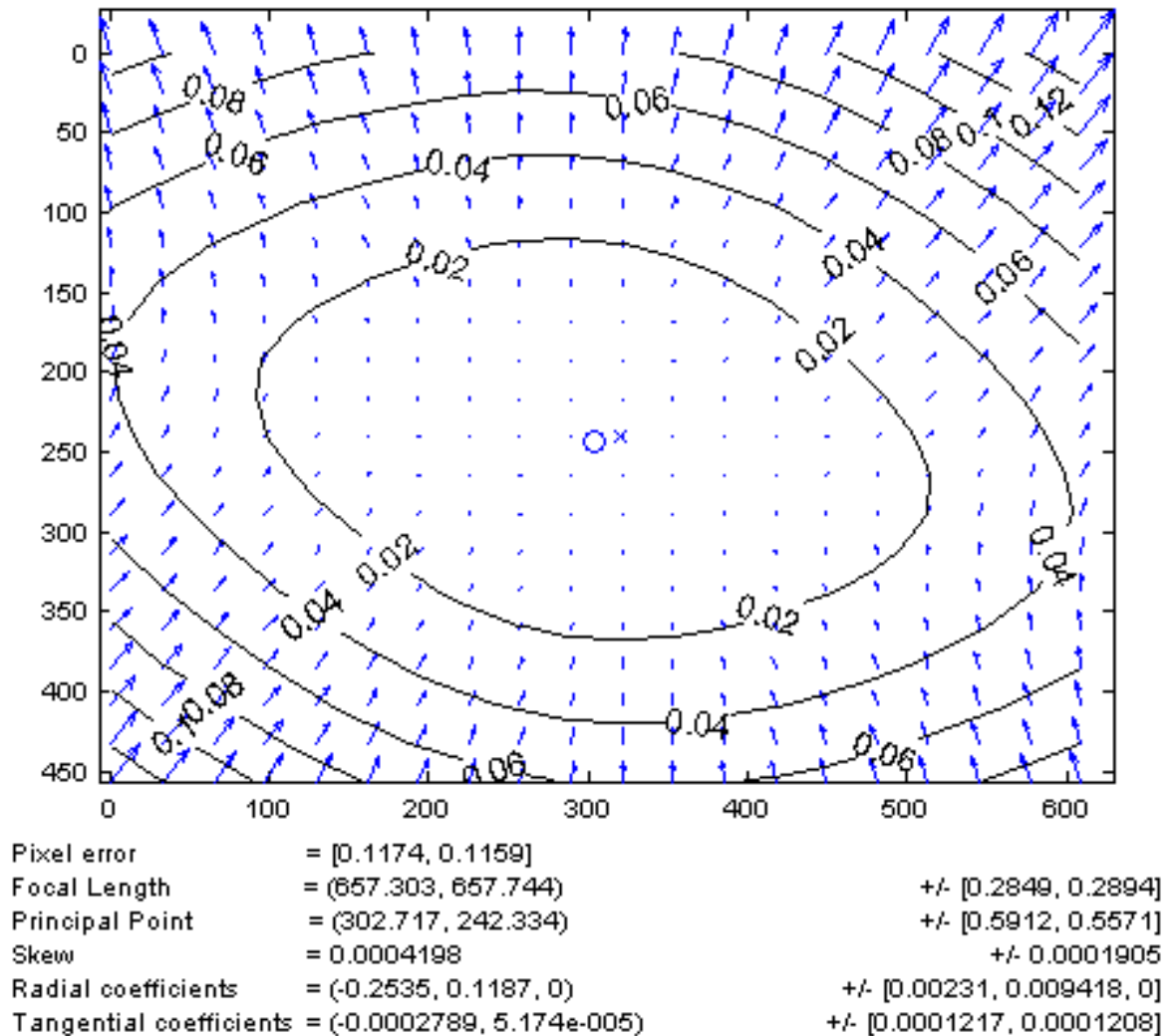


Fuente. Disponible en [6]

La siguiente figura muestra el impacto producido por la componente de distorsión tangencial. En esta imagen el máximo desplazamiento inducido es de 0,14 pixeles.

Figura 50. Modelo de distorsión tangencial

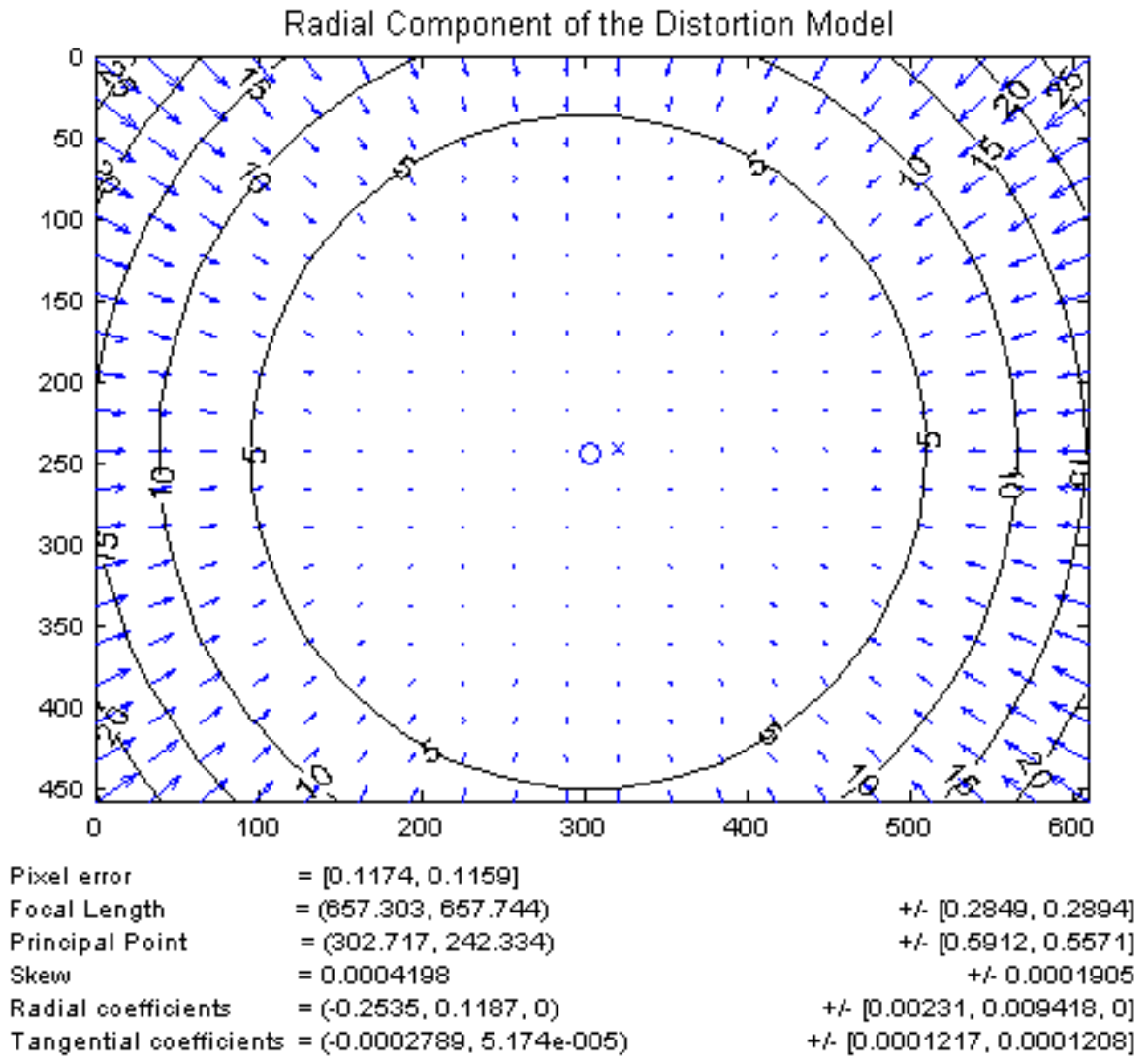
Tangential Component of the Distortion Model



Fuente. Disponible en [6]

Finalmente en la tercera figura 5.23, se observa el impacto que tiene la componente radial de distorsión. Esta imagen es muy similar a la imagen del modelo de distorsión completo o total, mostrando que el modelo de distorsión tangencial podría ser descartado del modelo total. En las tres imágenes, la cruz significa el centro de la imagen y el círculo la locación del punto principal.

Figura 51. Modelo de distorsión radial



Fuente. Disponible en [6]

Con esto se da fin a la primera parte de la calibración estéreo y los resultados que corresponden a los parámetros intrínsecos¹⁴ de las cámaras, se almacenan al presionar el botón de SAVE en la interfaz del software CCT.

¹⁴Ver anexo: Parámetros intrínsecos

Con el fin de obtener las posiciones de la cuadrícula ubicada en el Z_0 (parámetros extrínsecos) de cada sistema de proyección-observación, y habiendo almacenado los parámetros intrínsecos de las cámaras CCD, el proceso anterior se le realizó también a una sola fotografía de la cuadrícula en cada sistema ubicada en su respectivo Z_0 .

A continuación se darán los parámetros generales para realizar la calibración de un sistema estéreo partir de los resultados obtenidos con la realización de todo el proceso anterior.

5.3 CALIBRACIÓN DE UN SISTEMA ESTÉREO, RECTIFICACIÓN DE UNA IMAGEN ESTÉREO Y TRIANGULACIÓN 3D ESTÉREO

A continuación se mostrará cómo usar el CCT para la calibración de un sistema estéreo (dos sistemas de proyección – observación) y como usar los resultados de la calibración estéreo para trabajar la rectificación de las imágenes estéreo y la triangulación 3D estéreo.

El primer paso consiste en invocar la interfaz *stereo_gui* que se encuentra también dentro de la página del Software libre de CCT, esto se logra pulsando el botón de CALIBRACION ESTEREO dentro de la interfaz del software de nuestra investigación.

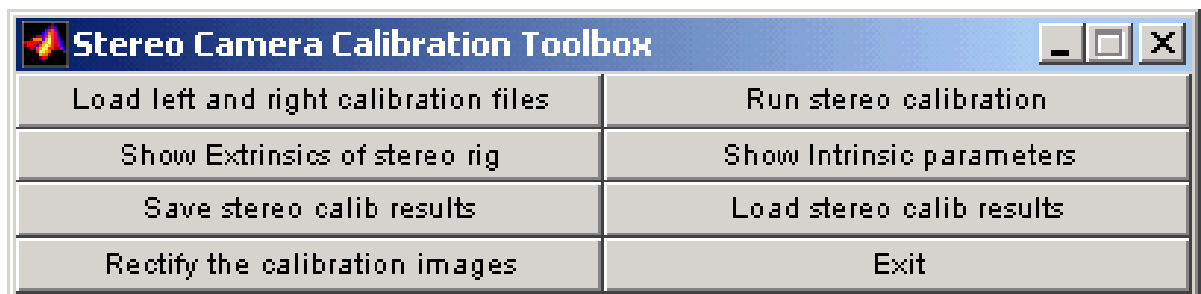
Figura 52. Control para invocar interfaz de *stereo_gui*



Fuente. Los autores

Ahora se deben obtener los veinte pares de imágenes correspondientes a cada uno de los sistemas de proyección, guardados debidamente para que sea de fácil identificación, además, se deben tener las calibraciones previas de cada uno de los sistemas independientes, paso que se realizó en el numeral anterior.

Figura 53. Interfaz de *stereo_gui*



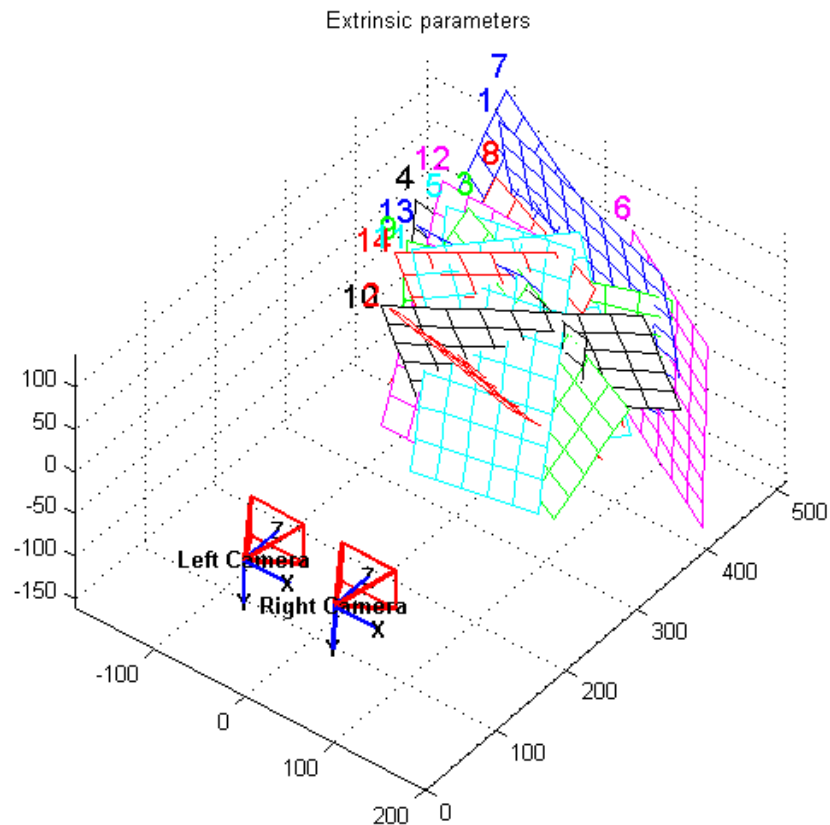
Fuente. Disponible en [6]

El primer paso, es la carga de los resultados obtenidos en la anterior sección, pulsando el botón *Load left and right calibration files* podemos buscar los archivos y cargarlos respectivamente.

Con la carga de los archivos, el programa recupera los valores intrínsecos y extrínsecos calculados (longitud focal, coeficiente de distorsión, punto principal, entre otros), lo cual, genera una ubicación de cada uno de los sistemas de observación con respecto del otro.

Para ejecutar el programa de optimización estéreo global, hacemos click en *Runstereocalibration*. Con esto, se re-calculan los parámetros intrínsecos y extrínsecos de los dos sistemas de observación, obteniendo así, los nuevos parámetros del sistema global, los cuales con este re-cálculo, han disminuido significativamente el error de reproyección (especialmente los valores focales), ya hecho este procedimiento, nos dirigimos a guardar los resultados de la optimización.

Figura 54. Parámetros extrínsecos sistema estéreo



Fuente. Disponible en [6]

La configuración espacial de las dos cámaras se pueden mostrar, como ha ocurrido anteriormente, haciendo uso de la gráfica de los parámetros extrínsecos. Finalmente, se debe realizar la rectificación de las imágenes estéreo usadas para la calibración, simplemente usando el *Rectifythecalibrationimages*, después de la rectificación, el programa genera un archivo de calibración únicamente para las imágenes rectificadas.

El proceso de calibración global se lleva a cabo como se describió con anterioridad, pasando por la extracción de las esquinas de cada una de las imágenes de los sistemas izquierdo y derecho, luego, optimizando y disminuyendo el error de re-proyección y por último extrayendo los parámetros extrínsecos e intrínsecos, como valor adicional, el software posee una función de triangulación estéreo la cual calcula la ubicación de cada uno de los puntos obtenidos en las imágenes izquierda y derecha.

Finalmente con los resultados obtenidos en esta segunda parte de la calibración estéreo se conoce la posición relativa de una cámara con respecto a la otra, así como la matriz de rotación los cuales son datos primordiales en tratamiento de las imágenes.

6. TRATAMIENTO DE LAS IMÁGENES

El tratamiento de las imágenes obtenidas del objeto a reconstruir es la metodología mediante la cual se convierten imágenes en dos dimensiones a imágenes tridimensionales y por lo tanto es el proceso que agrupa todo el marco teórico que es la base de esta investigación.

6.1 OBTENCIÓN DE LAS IMÁGENES DEL OBJETO A RECONSTRUIR

Teniendo calibrado correctamente el sistema (de manera independiente y estéreo), es necesario ubicar el objeto a reconstruir dentro de la zona de reconstrucción, es decir dentro del campo de visión de las dos cámaras CCD y en donde las franjas proyectadas desde los dos Video Proyectores incidan sobre la superficie de dicho objeto.

Una manera sencilla de determinar el lugar correcto del posicionamiento del objeto a reconstruir se logra mediante la proyección de los ejes ópticos de los dos sistemas de observación y el punto de intersección de estas dos líneas es la posición óptima para ubicar los modelos a reconstruir.

Una vez habiendo ubicado el modelo, ver Figura 6.1, es necesario extraer cinco imágenes de cada sistema de observación (de manera muy similar a como se hizo durante la calibración independiente), donde cuatro de estas imágenes una a continuación de la otra, no son más que la misma imagen del objeto con las franjas proyectadas sobre él aplicándoseles un corrimiento de un paso, y la imagen cinco corresponde al objeto con una sola franja proyectada sobre él.

Figura 55. Visualización del modelo a reconstruir



Fuente. Los autores

Para la obtención de las imágenes del objeto a reconstruir, dentro de la interfaz gráfica hay dos *PushButton* (uno para cada sistema de observación) llamados “imágenes objeto”, al activarlos se obtienen y guardan en la dirección especificada las cinco imágenes del modelo.

Figura 56. Control para la capturar y guardar imágenes del modelo



Fuente. Los autores

6.2 OBTENCIÓN DEL X, Y, Z DEL OBJETO A RECONSTRUIR.

Cuando se toman las imágenes del objeto, se puede iniciar el tratamiento, para esto se activa el botón de tratar de cualquiera de los dos sistemas de proyección-observación y se introduce el nombre del archivo el cual corresponde al nombre designado a las cinco imágenes del objeto.

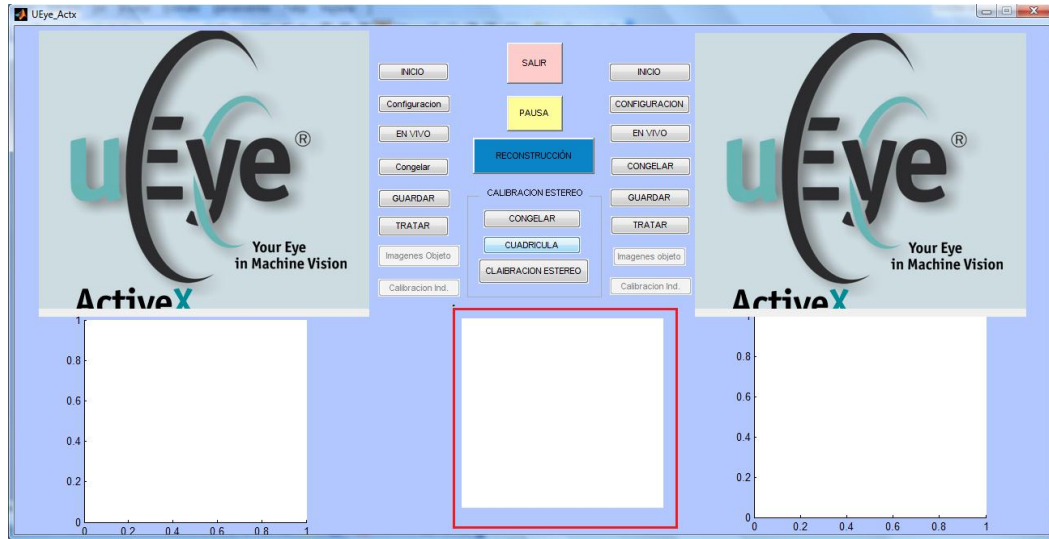
Figura 57. Control para Tratamiento de las imágenes



Fuente. Los autores

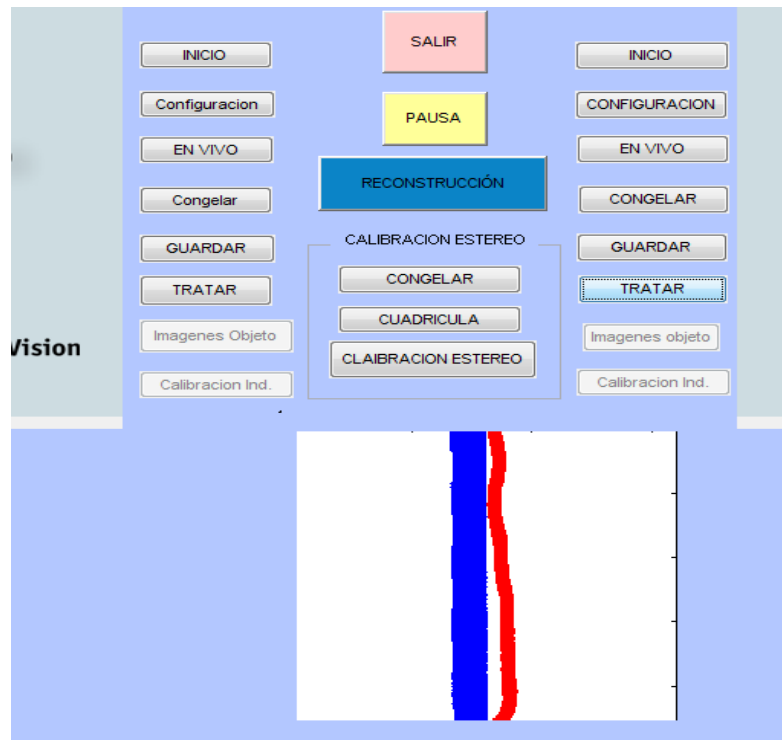
Posteriormente en la zona resaltada sobre la figura 6.4 de la interfaz del software el programa empieza a superponer los perfiles de la calibración sobre el perfil del objeto a reconstruir (Figura 6.5.) con el fin de recuperar información de un posicionamiento en X, Y, Z del objeto que sirva como referencia para el posterior cálculo apropiado de la fase.

Figura 58. Zona de superposición de los perfiles.



Fuente. Los autores

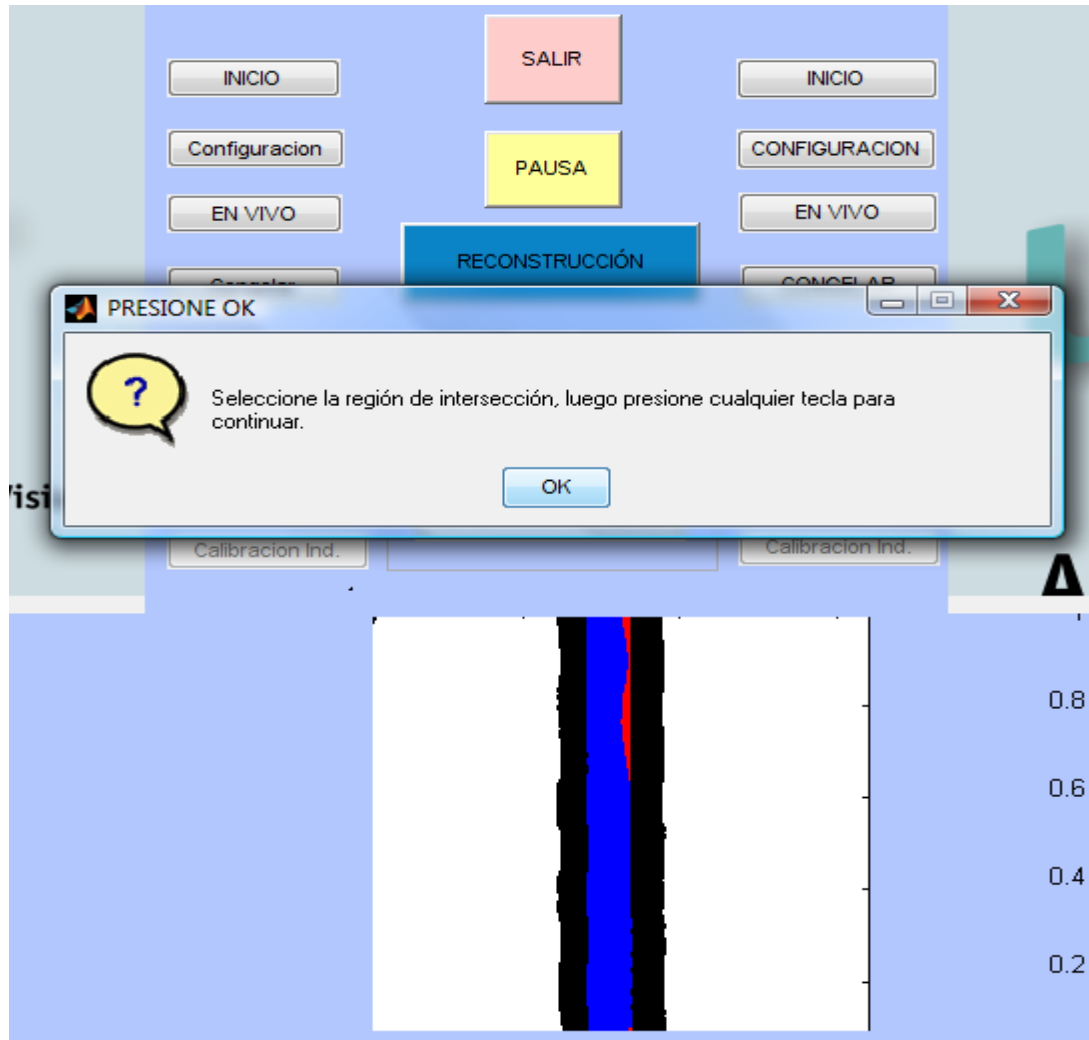
Figura 59. Superposición de los perfiles.



Fuente. Los autores

Terminada la superposición aparece un mensaje de advertencia como el que se ve en la figura 6.6 el cual sirve como pausa para seleccionar la región de intersección de algún punto de la línea del perfil del objeto (de color rojo) sobre alguna línea de los perfiles de calibración (de color azul).

Figura 60. Pausa iniciar proceso de selección del punto de intersección

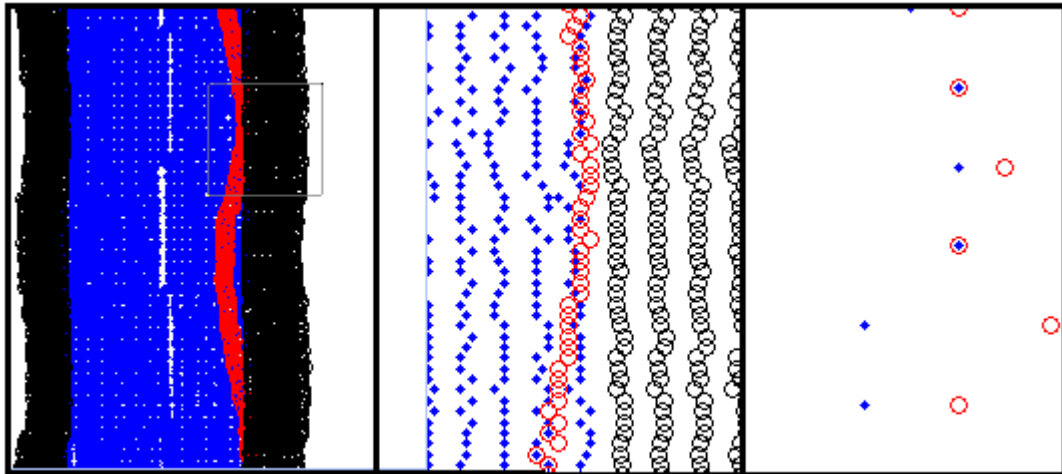


Fuente. Los autores

Al oprimir "OK" sobre la advertencia se habilita el zoom para aumentar el tamaño de la figura creada y así seleccionar de manera precisa el punto de intersección,

se aumenta el tamaño hasta que se visualice correctamente dicho punto así como se muestra en la secuencia de la siguiente figura.

Figura 61. Aumento de la visualización de los perfiles

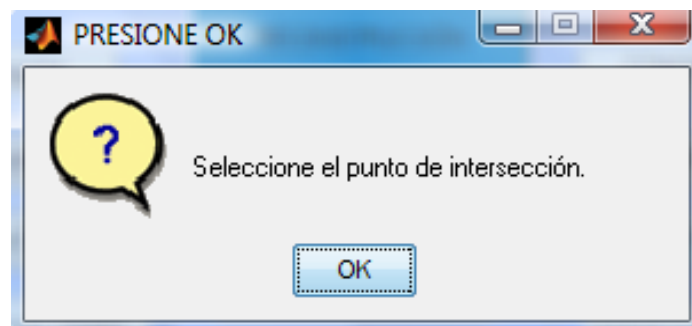


Fuente. Los autores

Las líneas negras corresponden a una extrapolación de los perfiles de calibración, como medida preventiva en el caso de que la línea del perfil del objeto no coincidiera con ninguna de las líneas azules.

Terminado esto, al oprimir cualquier tecla del ordenador aparece la advertencia que se muestra a continuación.

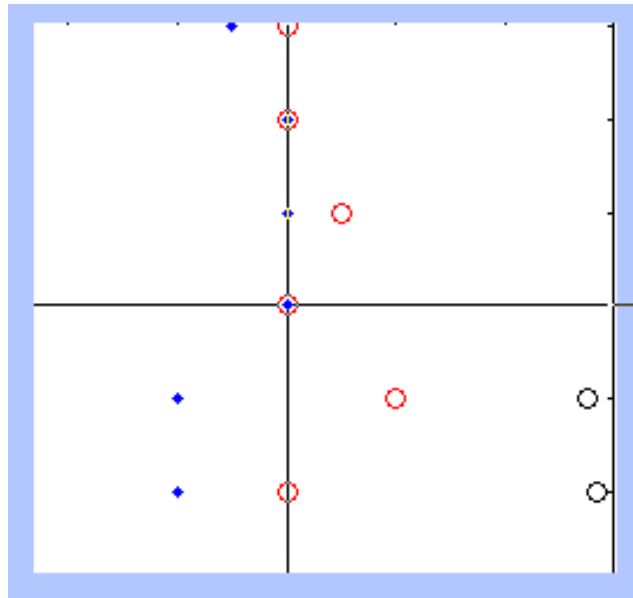
Figura 62. Advertencia para seleccionar el punto de intersección.



Fuente. Los autores

Oprimiendo “ok” se puede seleccionar el punto de intersección como se muestra en la siguiente imagen.

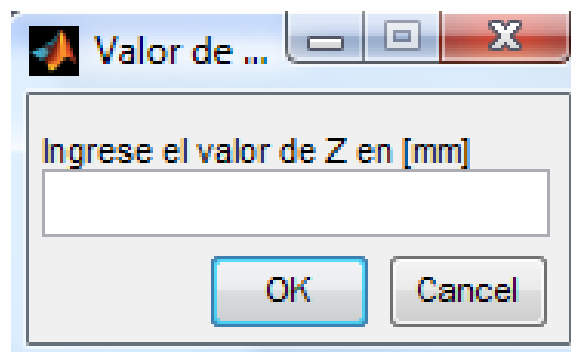
Figura 63. Selección del punto de intersección.



Fuente. Los autores

Al seleccionar este punto se pide información de la posición en Z para ese punto mediante la advertencia de la figura anexa.

Figura 64. Cuadro de dialogo para ingresar posición en Z

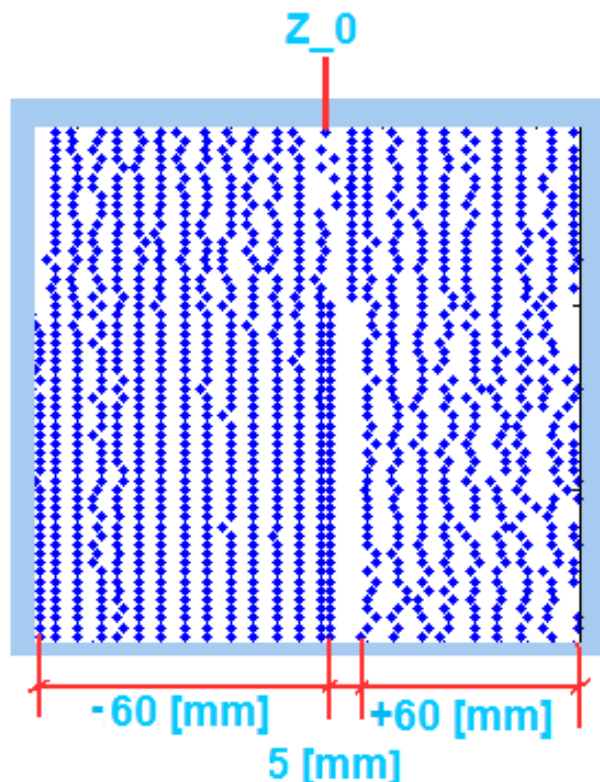


Fuente. Los autores

Este valor se determina basándonos en los perfiles de calibración, de la siguiente manera:

Como ya se anunció, en la posición 13 de la calibración independiente se encuentra la referencia del Z_0 y se sabe que a cada perfil de calibración está separado uno del otro por 5 [mm] y designando al eje Z hacia la derecha positivo y hacia la izquierda negativo como se muestra en la figura 6.11, resulta sencillo determinar el valor de Z donde el perfil del objeto coincide con el de la calibración.

Figura 65. Determinación de la posición de Z



Fuente. Los autores

Finalmente al seleccionar el punto exacto de intersección e introducir el valor de Z en milímetros, el programa simultáneamente extrae las coordenadas X e Y en píxeles las cuales posteriormente se convertirán a unidades métricas asegurando

que las coordenadas de la figura reconstruida tengan concordancia con las medidas del modelo de reconstrucción.

6.3 EXTRACCIÓN DE LA FASE

Dentro del mismo comando de TRATAR de la interfaz del programa de reconstrucción tridimensional, el software calcula la función corrimiento de fase $\emptyset(x, y)$ mediante un cálculo unitario de la fase para cada posición de las calibraciones independientes y su posterior agrupamiento dentro de una función cuadrática con la cual se podrá obtener información de la altura de los objetos a reconstruir.

Para calcular cada fase de cada uno de los posicionamientos en la dirección Z de las calibraciones independientes se parte de la ecuación 13 explicada en el ítem 1.3 de *Corrimiento de fase a Intervalos regulares*.

$$tg(\emptyset) = \frac{I_2 - I_4}{I_1 - I_3}. \quad (13)$$

Donde como ya se indicó I_1 equivale a un corrimiento de fase $\varphi_1 = 0$, I_2 corresponde a un corrimiento de fase $\varphi_2 = \frac{\pi}{2}$, I_3 corresponde a un corrimiento de fase $\varphi_3 = \pi$ y por último I_4 corresponde a un corrimiento de fase $\varphi_4 = \frac{3\pi}{2}$.

El contraste luminoso $A(x, y)$ es calculado (ecuación 14) con el fin principal de servir como base para la construcción de la máscara que permitirá seleccionar una *región blanca* dentro del contorno de una imagen del objeto a reconstruir para analizar y modificar el umbral.

$$2 * A(x, y) = \sqrt{(I_2 - I_4)^2 + (I_1 - I_3)^2}. \quad (14)$$

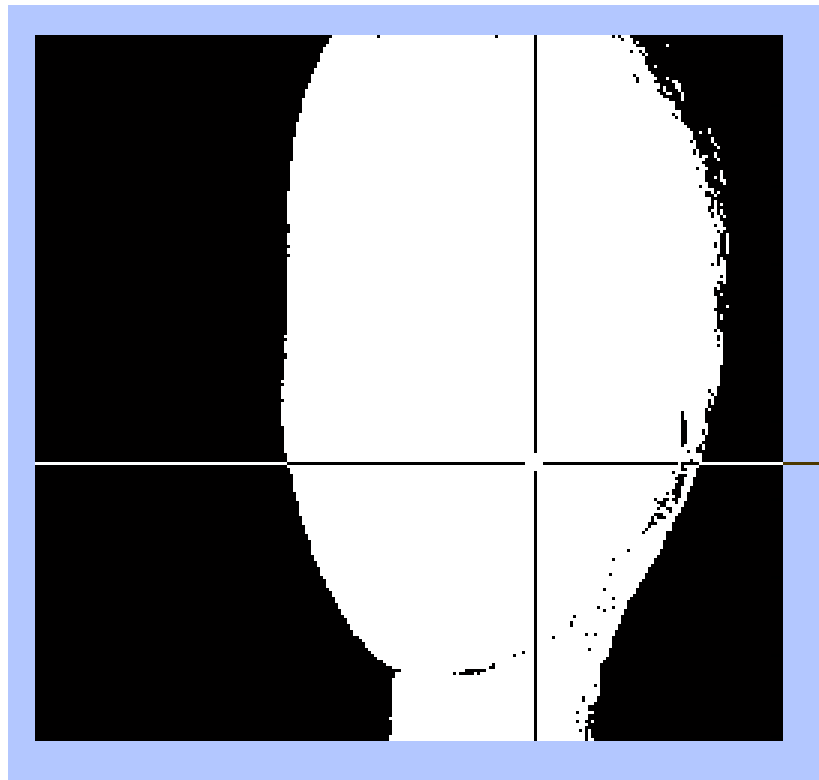
Esta máscara se construye en escala de grises y cada parte de la imagen resultante que sea de color blanca representa una imagen óptima para realizar la reconstrucción, dado el caso de no mostrarse completamente blanca la región que se desea reconstruir es necesario modificar el umbral.

6.4 UMBRAL INICIAL

Se denomina Umbral al valor que permite definir el filtro que se le interpone a las imágenes capturadas para ajustar la región que se quiere reconstruir, dentro del Software de esta investigación el Umbral inicial tiene un valor predeterminado de 0,1 entendiéndose que al aumentar la magnitud del umbral la región de la imagen a reconstruir es más amplia, haciendo que los detalles de dicha imagen sean más difíciles de apreciar, y si la magnitud del umbral disminuye es posible extraer información de aquellas zonas de difícil acceso.

Para realizar la reconstrucción es necesario que se seleccione un punto dentro la *región blanca*(figura 6.12) de esta manera se señala el punto de partida de acuerdo con el umbral predeterminado, si se desea cambiar el umbral, el programa arroja una advertencia la cual permite modificarlo como se ve en la figura 6.13 no sin antes mostrar una imagen previa a la reconstrucción que sirve para confirmar si el umbral asumido es correcto (Figura 6.14) y en donde se puede observar si la imagen no presenta discontinuidades que imposibiliten su reconstrucción.

Figura 66. Selección de la Región blanca dentro de la imagen a reconstruir



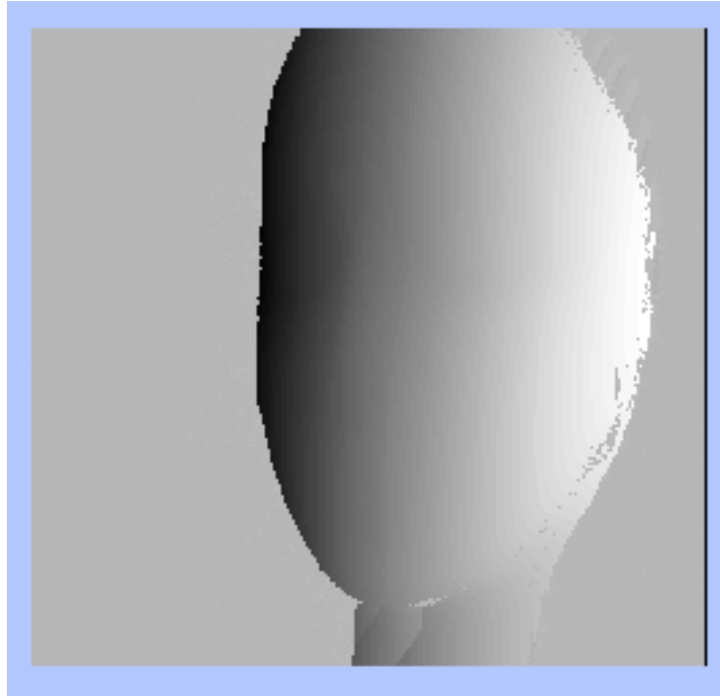
Fuente. Los autores

Figura 67. Advertencia para modificar el Umbral



Fuente. Los autores

Figura 68. Imagen de Confirmación del Umbral

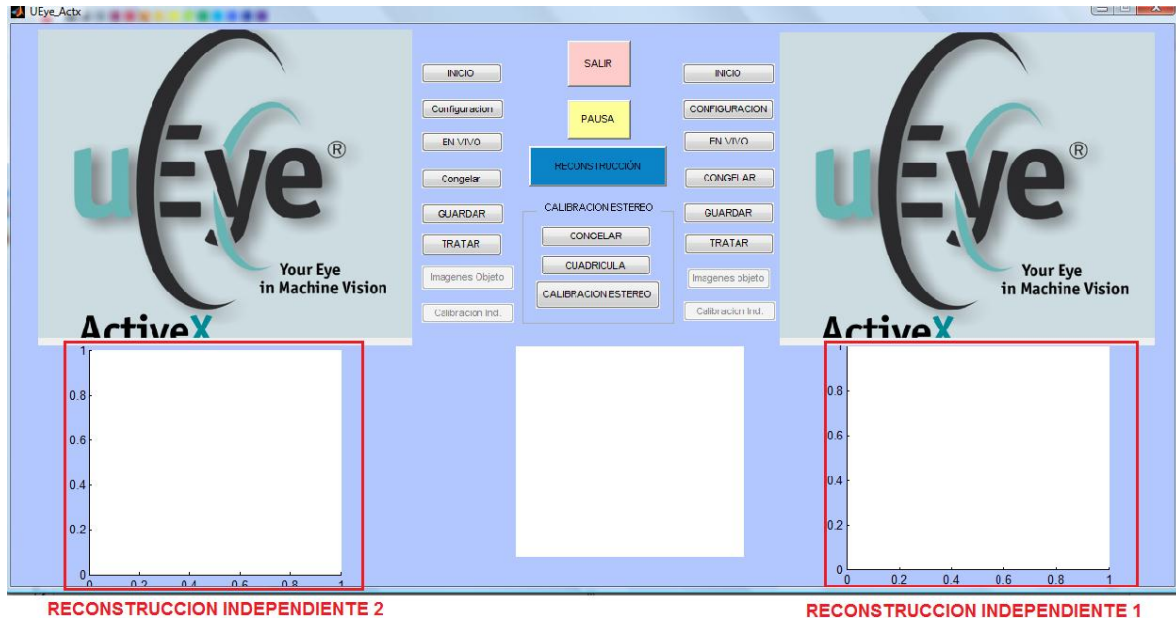


Fuente. Los autores

6.5 RECONSTRUCCIÓN DE LOS SISTEMAS INDEPENDIENTES

Al terminar el tratamiento independiente de las imágenes el software automáticamente reconstruye tridimensionalmente la sección del modelo de reconstrucción que fue visualizada por cada sistema de observación y son mostradas en la región señalada en la Figura 6.15 de la interfaz.

Figura 69. Zona de visualización de la reconstrucción independiente



Fuente. Los autores

Para realizar esta reconstrucción independiente el programa carga los parámetros de la calibración independiente extrayendo las matrices que se nombran a continuación, los cuales corresponden al cálculo de la fase en cada punto de cada posición de la calibración independiente.

Name	Size
MA	480x640
MB	480x640
MC	480x640
Phase_0	480x640

Estas matrices están relacionadas por una función cuadrática cuya variable corresponde al valor de Z de cualquier punto de cualquier objeto al cual se le haya iniciado el tratamiento. A continuación se muestran las ecuaciones para solucionar dicha función cuadrática.

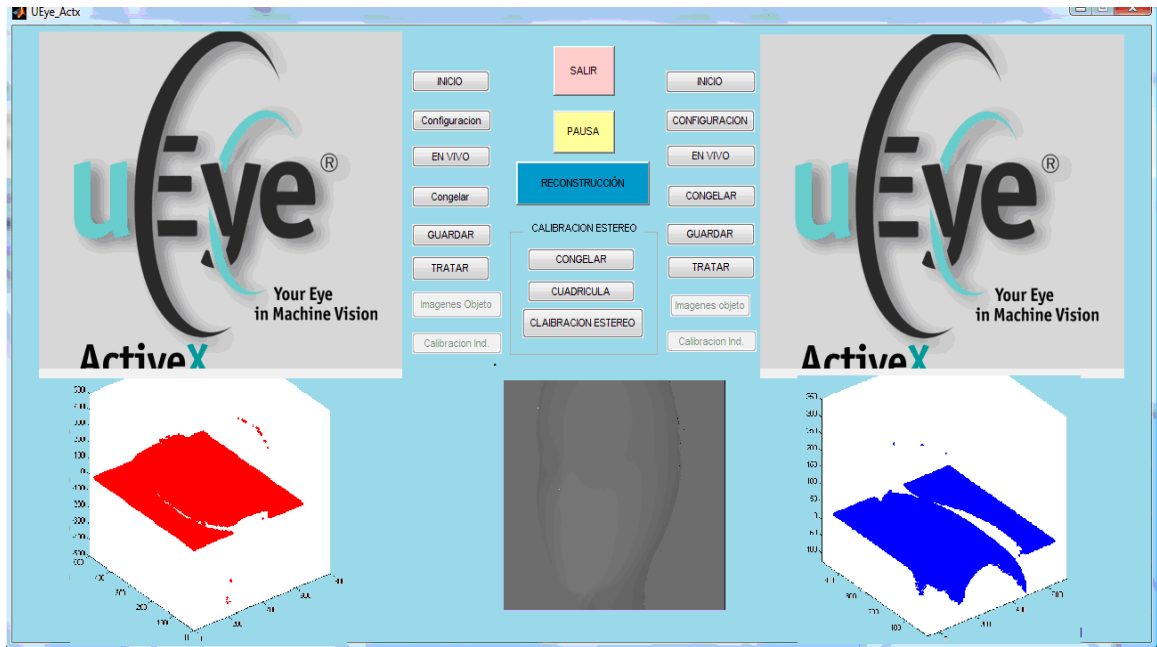
$$ZP = \frac{-MB + \sqrt{MB^2 - 4 \cdot MA \cdot (MC - \text{objeto})}}{2 \cdot MA} \quad (21)$$

$$ZN = \frac{-MB - \left[\sqrt{MB^2 - 4 \cdot MA \cdot (MC - \text{objeto})} \right]}{2 \cdot MA} \quad (22)$$

Donde la variable *objeto*, consiste en una diferencia de las fases obtenidas de la matriz Phase_0 y de la fase calculada una vez interpuesto la máscara y el umbral a la imagen del objeto a reconstruir.

De donde es lógico se esperan dos soluciones (ZP y ZN), de las cuales una corresponderá a la matriz de datos que se espera sirvan como solución y la otra puede ser despreciada. A continuación se muestran dos ejemplos de imágenes reconstruidas independientemente tal y como se muestran en la interfaz del software diseñado, sin la realización del tratamiento posterior de las figuras reconstruidas.

Figura 70. Reconstrucción independiente



Fuente. Los autores

6.6 RECONSTRUCCIÓN ESTÉREO (SISTEMA GLOBAL)

En la figura 7.8 y la figura 7.9, que corresponden a las imágenes de reconstrucción de los sistemas independientes, se aprecia que cada una corresponde a una sección del modelo a reconstruir y sin la realización del tratamiento final de las imágenes, el paso denominado reconstrucción estéreo consiste en la realización del empalme de estas dos figuras así como su posterior tratamiento para una óptima visualización, este empalme se basa en los datos obtenidos en la calibración estéreo.

Para iniciar el proceso de empalme y tratamiento de las imágenes de reconstrucción independiente es necesario pulsar el botón de RECONSTRUCCION en la interfaz del software de esta investigación.

Figura 71. Control para el empalme de las imágenes de reconstrucción



Fuente. Los autores

El software toma los resultados extrínsecos de la calibración estéreo, es decir las posiciones relativas de cada Z_0 con los ejes de cada sistema de observación correspondiente y adicionalmente la posición relativa de los dos sistemas de observación.

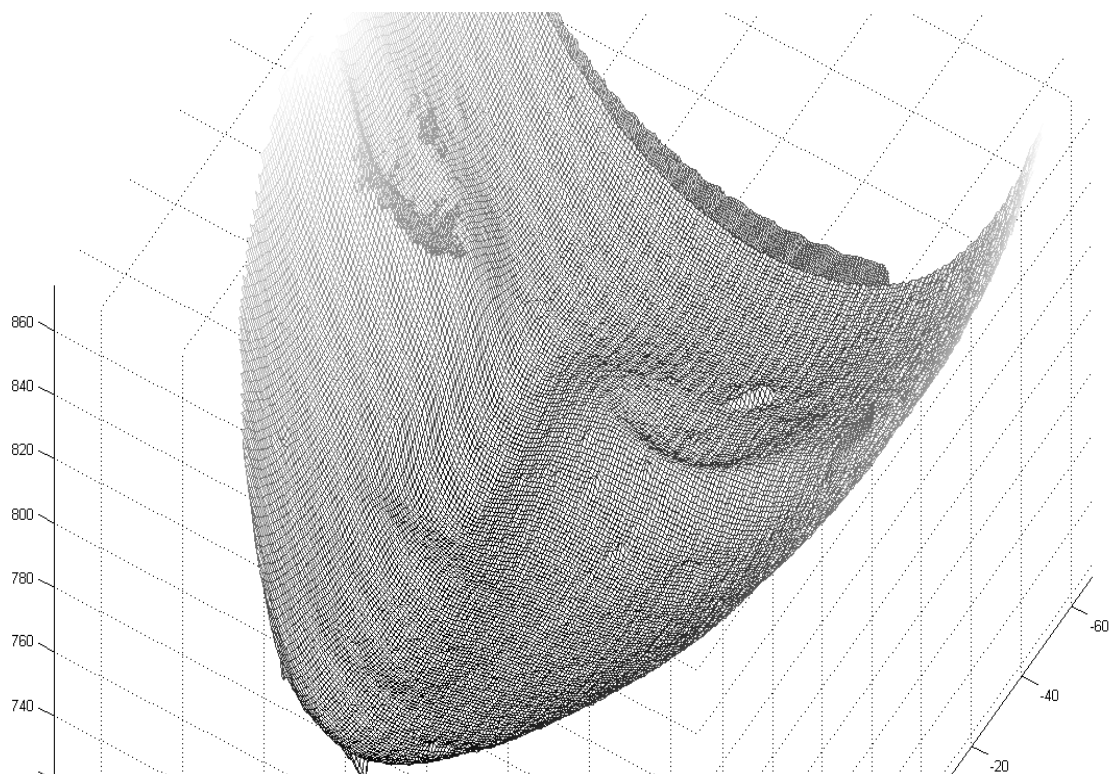
Con estos datos (principalmente los vectores rotacionales y traslacionales) y mediante la ecuación (23), se trasladan y rotan cada una de las figuras de reconstrucción independiente, logrando una correcta unión entre ellas como se muestra en la figura 6.18 la cual corresponde al objeto ya reconstruido.

$$X_R = R * X_L + T \quad (23)$$

En donde X_R corresponde a la posición de cada punto a donde se desea trasladar, X_L la posición de donde se quiere desplazar y R a la matriz de rotación y T al vector de traslación.

De esta manera se traslada cada punto de las figuras de reconstrucción independientes, primero hacia la posición de cada cámara y luego (definiendo como origen la posición de cualquiera de ellas) hacia un mismo origen.

Figura 72. Objeto reconstruido



Fuente. Los autores

7. ANÁLISIS DE RESULTADOS

7.1 ANÁLISIS DEL OBJETO RECONSTRUIDO

Los modelos de reconstrucción deben poseer ciertas características específicas para la optimización de la técnica implementada. Estas características, más que geométricas, son superficiales como: color, brillo y contraste.

Los objetos reconstruidos mediante la técnica de proyección de franjas, tienen en común que sus superficies son de colores claros (preferiblemente blancos) distribuidos uniformemente y no tienen recubrimientos que puedan reflejar la luz proyectada.

Sin embargo los modelos reconstruidos durante esta investigación y aquellos modelos que se quieran reconstruir por medio del software diseñado deben tener las dimensiones apropiadas (Medidas que puedan ser captadas en el rango de los sistemas de observación y además, estén dentro de los parámetros de la calibración), de tal manera que éstos queden ubicados correctamente al interior del espacio de trabajo; este último completamente definido por el campo de observación de los sistemas de adquisición.

7.1.1 Caracterización del objeto a reconstruir.

Como parámetro inicial, se seleccionó el objeto, teniendo en cuenta todos los aspectos anteriormente nombrados, en este caso, nuestra modelo, denominada “muñeca”, es un maniquí hecho de icopor, cuya superficie esta algo maltratada (como se observa en la figura 7.1), con una altura aproximada de 28 cm y un diámetro de la cabeza de aproximadamente 14 cm; debido a la naturaleza de su material, ésta, no refleja la luz incidida en ella y a pesar de que sus detalles son aparentemente tenues (nariz y boca), éste objeto no es sencillo de reconstruir.

Figura 73. Objeto a reconstruir

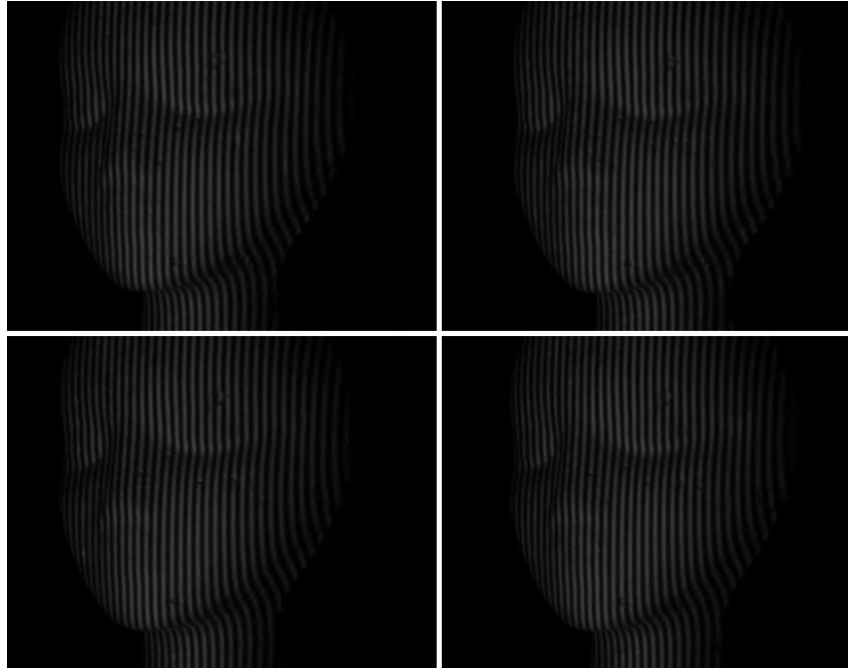


Fuente. Los autores

7.1.2 Recopilación de información del modelo

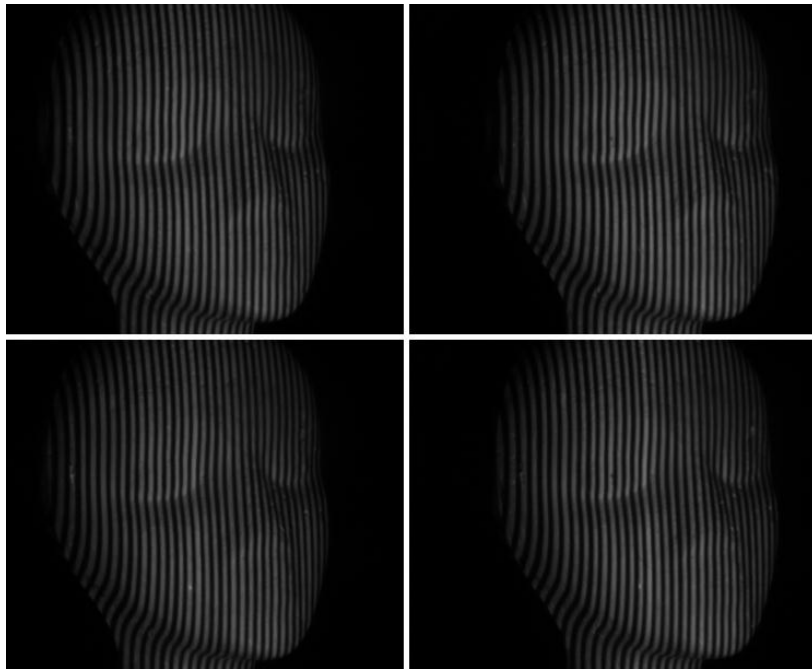
En esta parte del proceso de reconstrucción, habiendo realizado la calibración de los sistemas independientes y del sistema estéreo, se realizó la toma de imágenes con las franjas proyectadas tal como se indicó en los ítems 6.1 y 6.2; en las figuras 7.2 y 7.3 se aprecian las imágenes capturadas por cada uno de los sistemas de observación.

Figura 74. Imágenes tomadas por el sistema independiente derecho



Fuente. Los autores

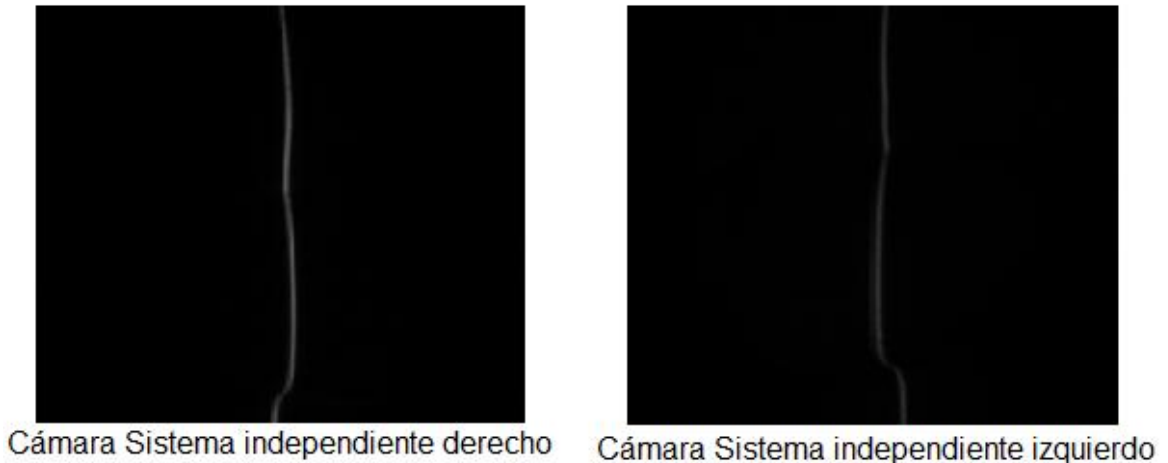
Figura 75. Imágenes tomadas por el sistema independiente izquierdo



Fuente. Los autores

Adicionalmente y siendo una imagen primordial para llevar a cabo el proceso de tratamiento de las imágenes de cada sistema de observación también es capturada una quinta imagen la cual corresponde al perfil del objeto a reconstruir como se observa en la figura 7.4.

Figura 76. Perfiles del Objeto a reconstruir



Fuente. Los Autores

7.1.3 Procesamiento de la información obtenida

Después de que el sistema ha analizado la información suministrada (objeto con franjas proyectadas), se procesan los datos de cada una de las imágenes, llegando a la obtención de las reconstrucciones de cada uno de los sistemas de manera independiente, como primera instancia, esto se hace con el fin de analizar cada una de las imágenes reconstruidas y observar que cantidad de puntos y áreas erróneas existen. Con la identificación de estos puntos o áreas (en la mayoría de los casos), se procede a realizar las modificaciones de los puntos con los cuales requerimos trabajar. A todo este proceso se le conoce como segmentación de las superficies para la reconstrucción tridimensional.

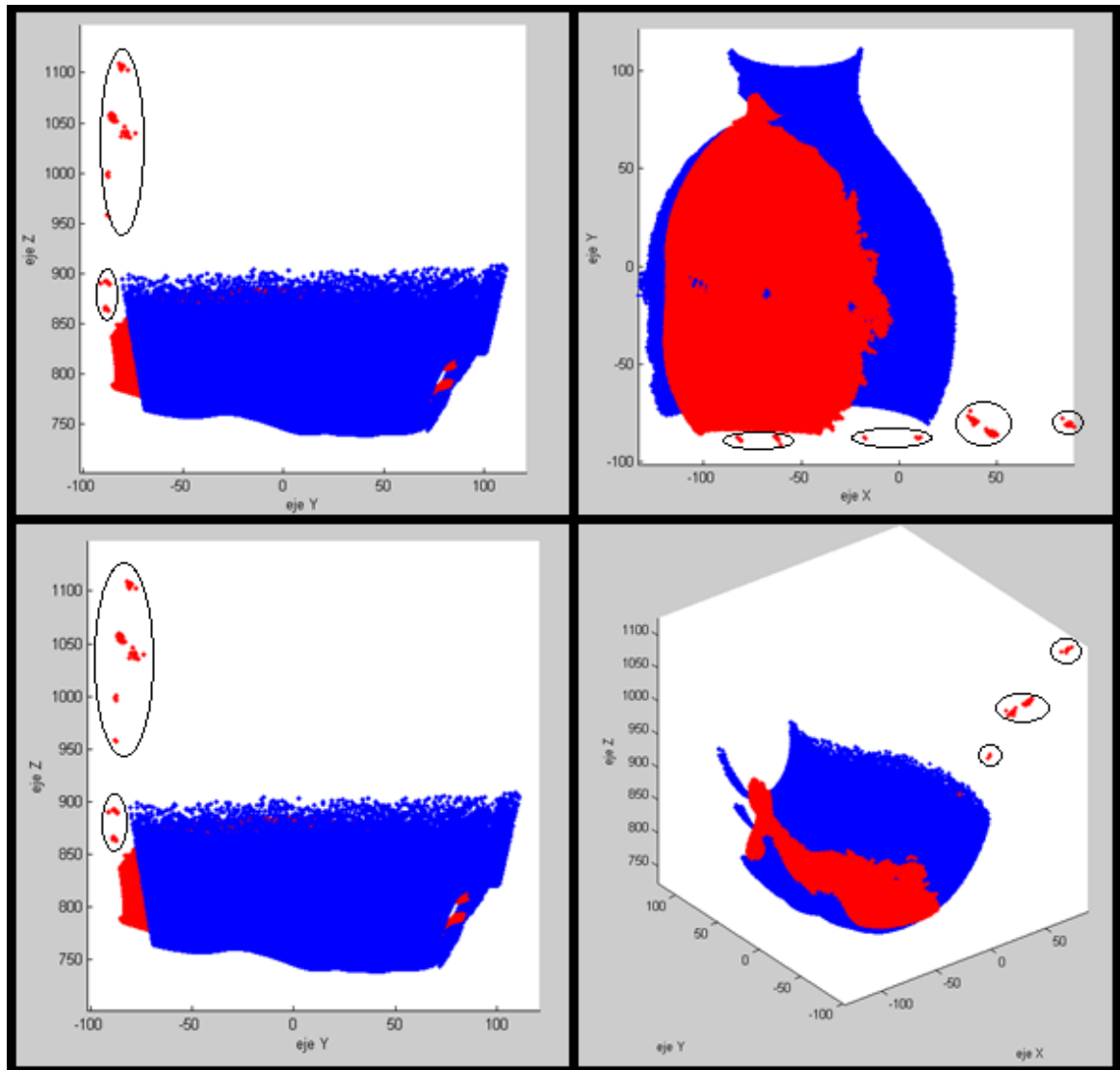
Figura 77. Reconstrucción del sistema independiente derecho, plano YZ



Fuente. Los autores

Como se puede observar en la Figura 7.5 el punto seleccionado corresponde a la coordenada (45.78, 52.42, 727.4); este punto así como cualquier otro que se encuentre fuera de la figura de reconstrucción, corresponden a distorsiones y por lo tanto deben ser eliminados para evitar resultados erróneos en el mallado final.

Figura 78. Aberraciones del objeto reconstruido



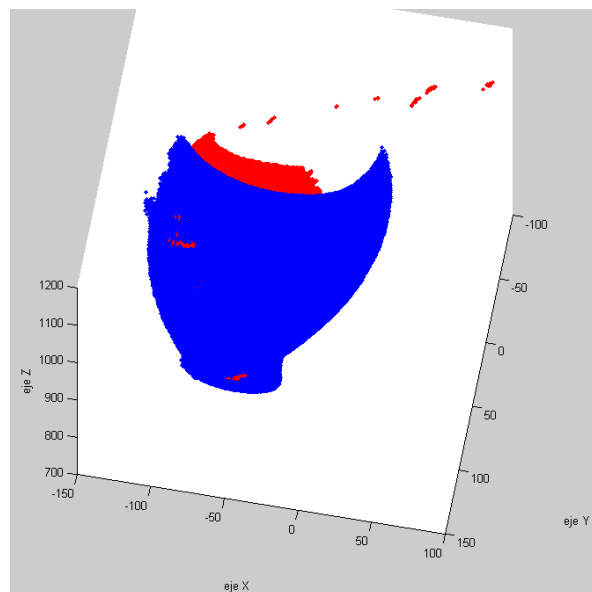
Fuente. Los autores

En la figura 7.6 se aprecian las distorsiones y aberraciones que se presentaron en el momento de realizar el tratamiento de las imágenes capturadas del modelo de reconstrucción. Estas distorsiones son generalmente producto de alguna fuente de luz dentro del lugar donde se lleva cabo la toma de imágenes.

7.1.4 Integración y tratamiento de las imágenes

El proceso de integración de las imágenes es la parte más robusta de nuestra investigación, en este paso, deben interactuar los vectores de traslación y las matrices rotacionales, los cuales se mencionan en el capítulo 5 y son calculados con los parámetros intrínsecos y extrínsecos de las cámaras CCD. La figura 7.6 muestra la integración de las imágenes reconstruidas por cada uno de los sistemas independientes, en donde la imagen de color rojo corresponde a la reconstrucción del sistema derecho y la azul del sistema izquierdo.

Figura 79. Integración de las reconstrucciones independientes (ejes coordenados en [mm])



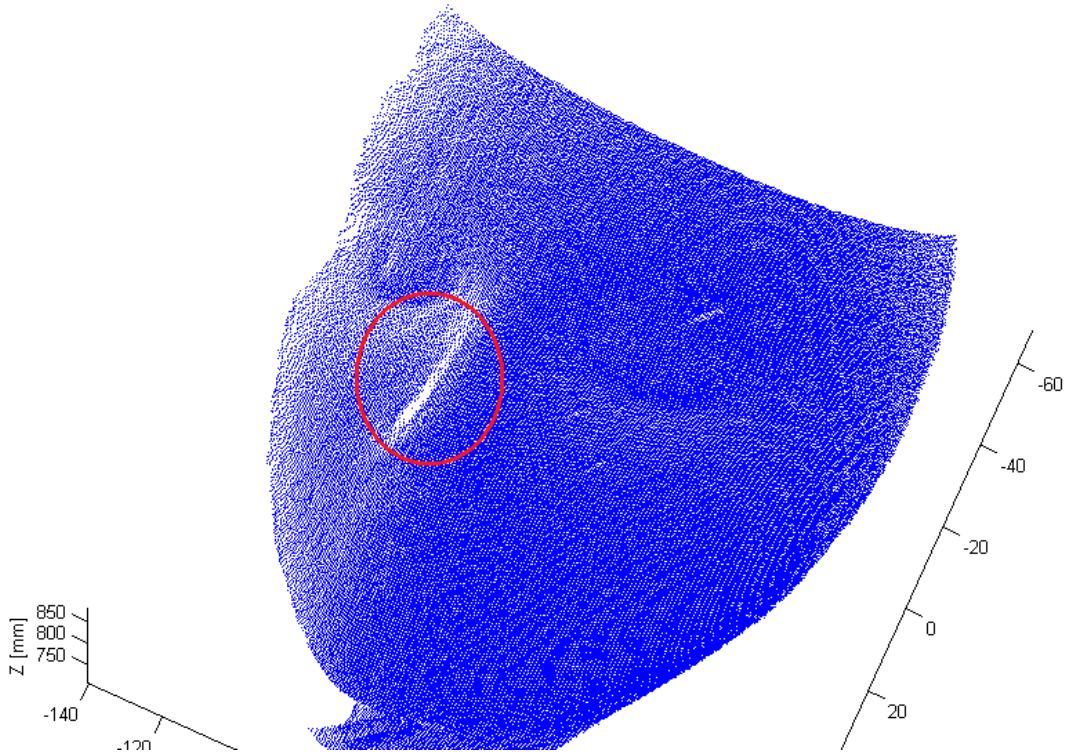
Fuente. Los autores

Si bien en la Figura 7.7 no es posible observar una forma específica del modelo, se quiere dar a conocer la integración de las reconstrucciones y más importante aún es el complemento que se observa en dicha figura, ya que como se habló a lo largo de esta investigación, la existencia de zonas ocultas (figuras 7.8 y 7.9) de

una figura obtenida de un sistema de proyección-observación serán completadas por la figura que se obtiene del otro sistema (ver figura 7.10).

Figura 80. Zona oculta para el sistema independiente derecho

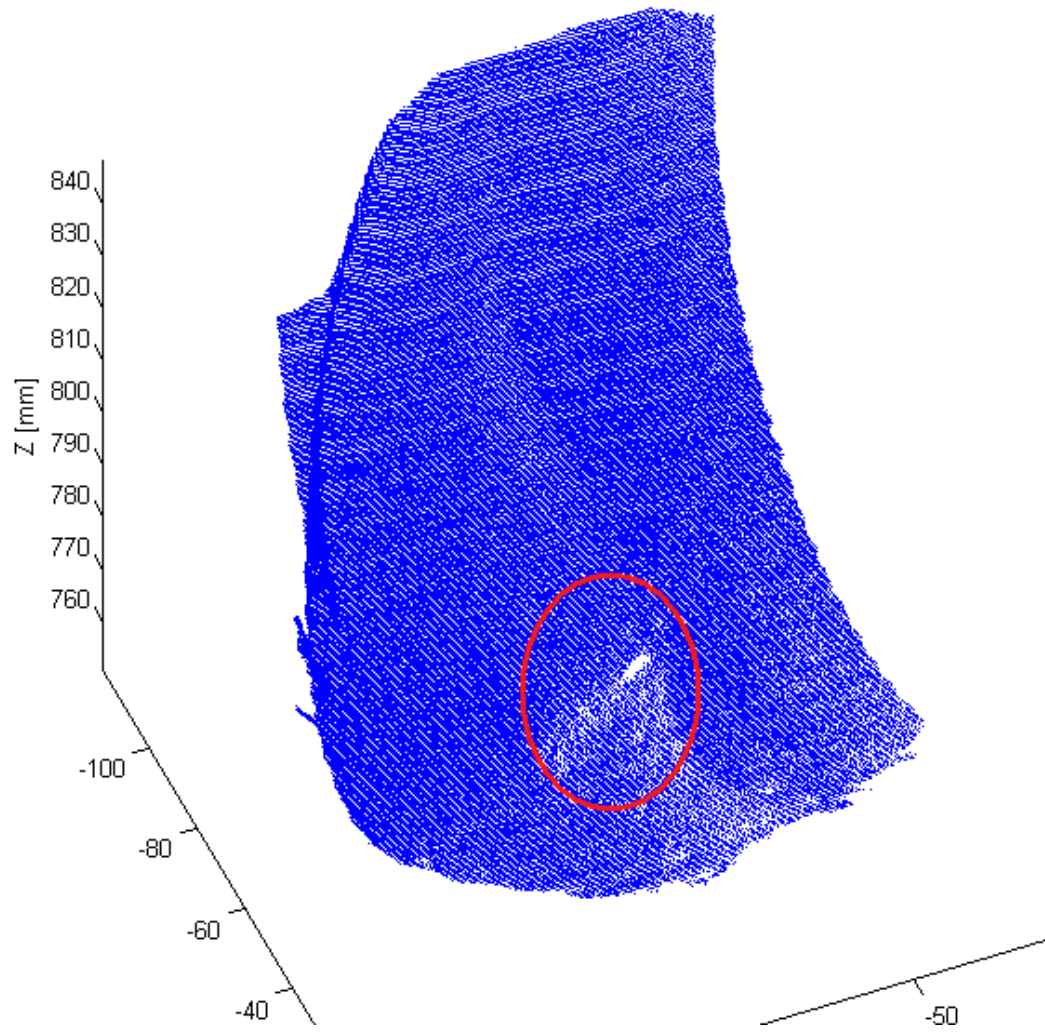
RECONSTRUCCIÓN 3D SISTEMA INDEPENDIENTE DERECHO



Fuente. Los autores

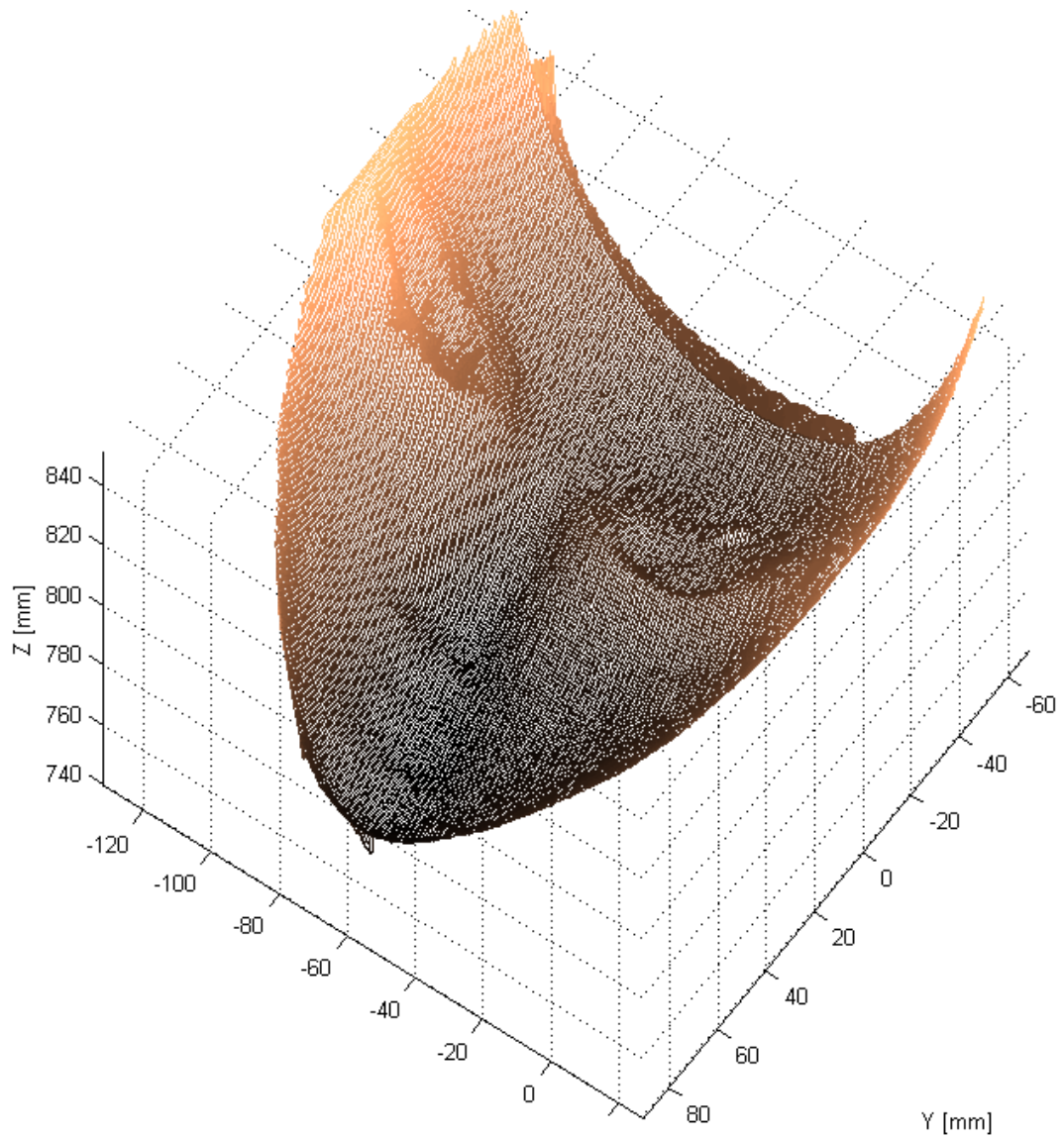
Figura 81. Zona oculta para el sistema independiente izquierdo

RECONSTRUCCION 3D SISTEMA INDEPENDIENTEIZQUIERDO



Fuente. Los autores

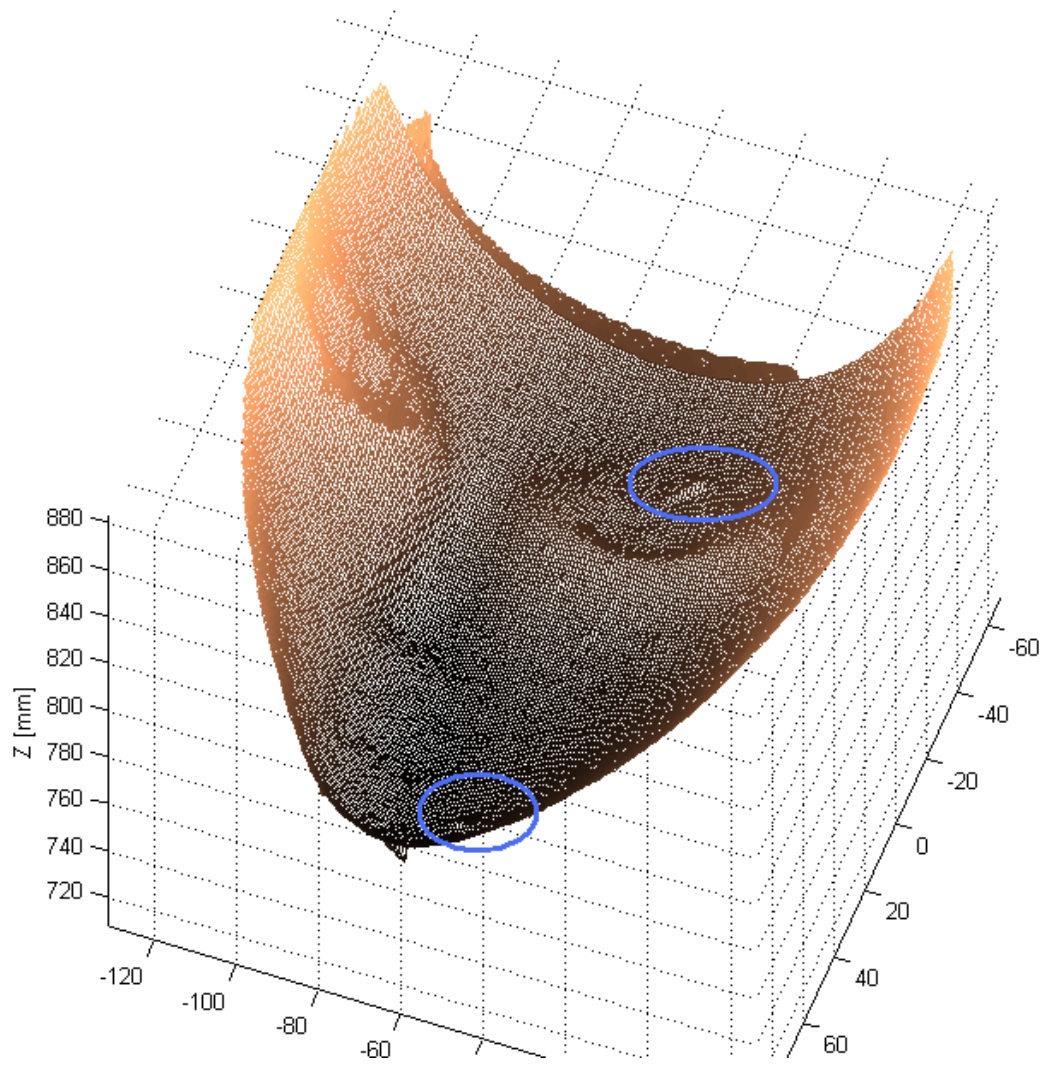
Figura 82. Imagen de la reconstrucción tratada por medio de un mallado



Fuente. Los autores

En la figura 7.11 y 7.12 se pueden apreciar la figura virtual y la figura real respectivamente, en las cuales se observan pequeños detalles que la reconstrucción es capaz de resolver, pero debido a que resulta muy complejo medir físicamente con alta precisión estos detalles se hace necesario diseñar una alternativa que cuantifique la precisión y resolución del sistema de reconstrucción 3D desarrollado.

Figura 83. Figura virtual



Fuente. Los autores

Figura 84. Figura real



Fuente. Los autores

7.2 RESOLUCION Y PRECISIÓN DEL SISTEMA

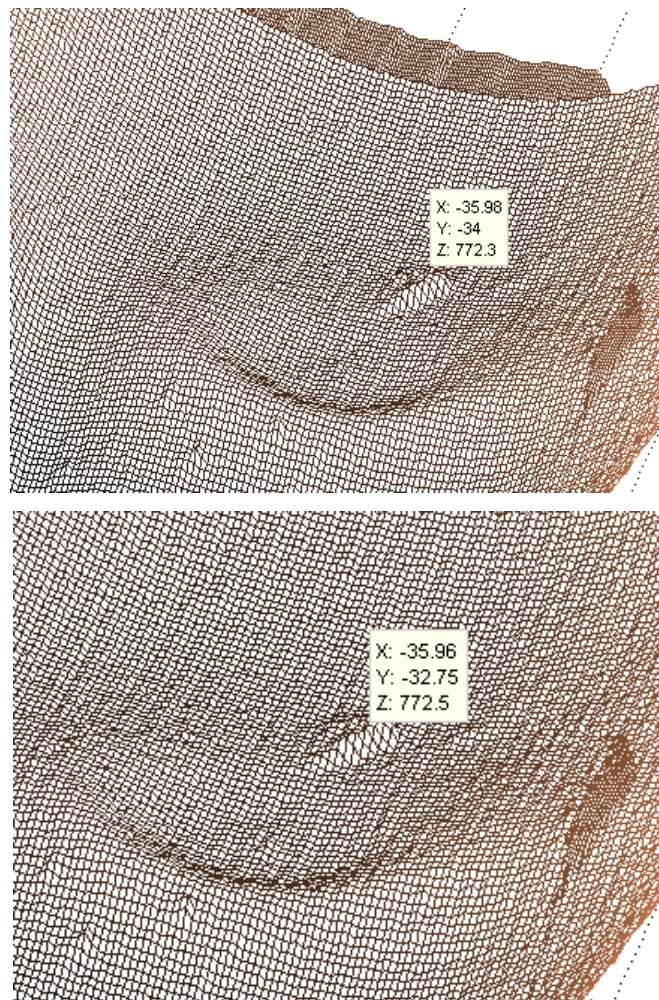
7.2.1 Resolución del sistema

La resolución del sistema de reconstrucción tridimensional se refiere a la capacidad que éste tiene de reflejar los detalles de un modelo. Para la determinación de la resolución global del sistema implementado en ésta investigación, es imprescindible hacer énfasis en los cambios de altura que han sido capturados y procesados por el software.

7.2.1.1 Resolución cualitativa

Una forma sencilla y experimental de determinar estos cambios de altura, es realizando la medición de ranuras o protuberancias existentes en el objeto de reconstrucción para luego compararlas con dicha ranura o protuberancia en la figura reconstruida, de esta manera se determina la resolución cualitativa del sistema. En la figura 7.13 se muestra la toma de datos en una variación topográfica visible en el objeto reconstruido.

Figura 85. Extracción de la profundidad de la ranura.



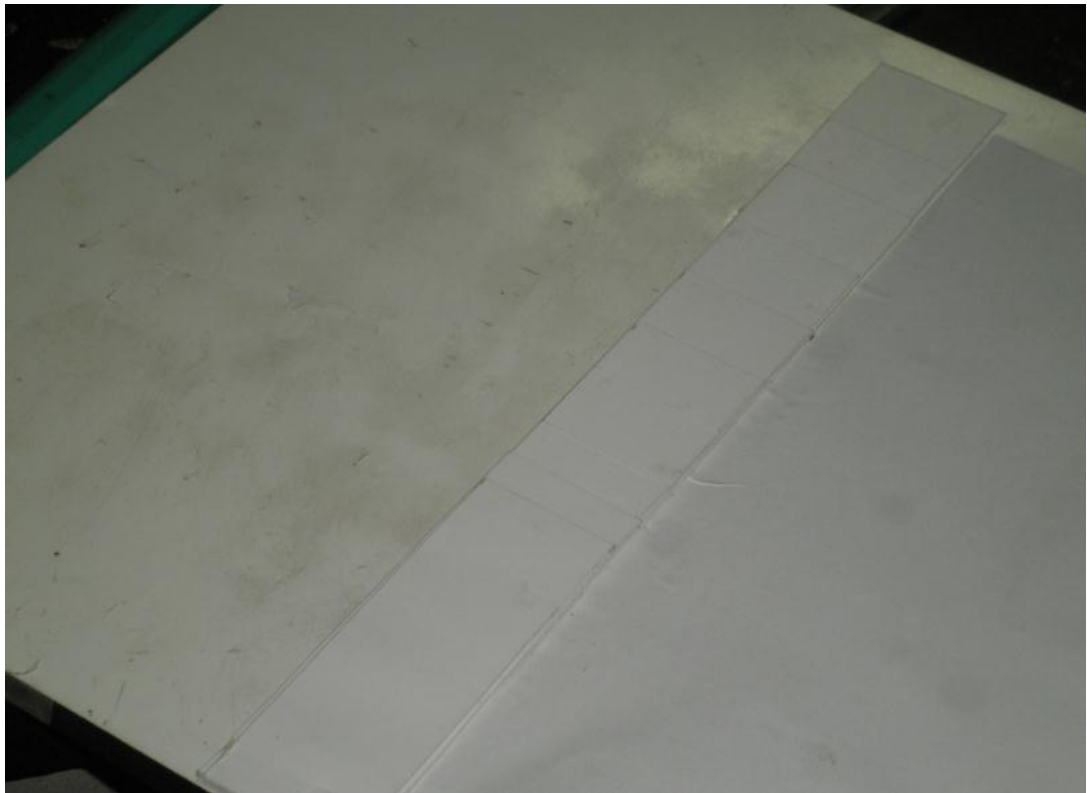
Fuente. Los autores

7.2.1.2 Resolución cuantitativa de los sistemas independientes

Mediante el cálculo de la resolución cuantitativa de los sistemas de proyección-observación, se puede determinar un valor más aproximado para la resolución de cada uno de dichos sistemas.

Para cuantificar la resolución de los sistemas independientes, es necesario, colocar tiras de cartulina una sobre otra en una tabla lisa como se muestra en la figura 7.14, y realizar la respectiva captura de las imágenes con sus franjas proyectadas, ubicando esta tabla en cada Z_0 de los sistemas independientes.

Figura 86. Tabla con tiras de cartulina superpuestas



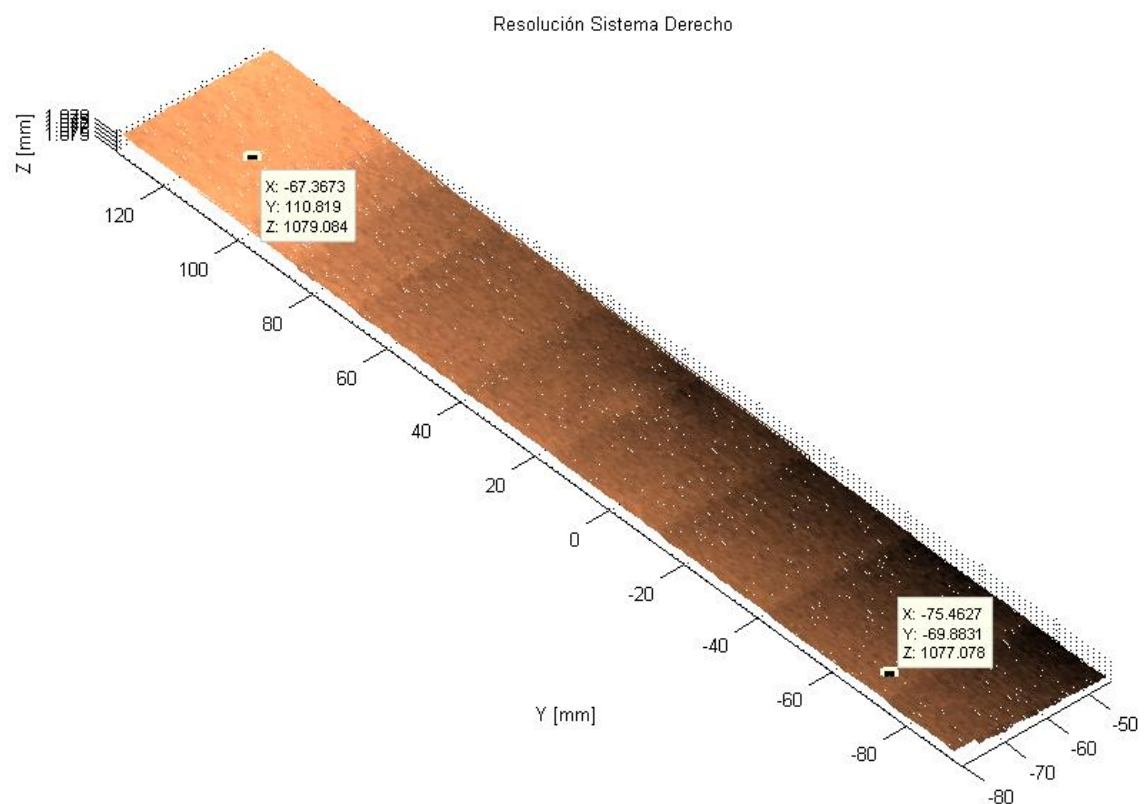
Fuente. Los autores

Una vez capturadas las imágenes el siguiente paso es realizar su tratamiento, de la misma manera como si se tratara de reconstruir cualquier objeto, y finalmente

se analizan cada una de las figuras reconstruidas, observando cuantas tiras de cartulina se pueden apreciar.

En la figura 7.15 se observan los resultados obtenidos para la determinación de la resolución del sistema derecho y se muestran las coordenadas (haciendo énfasis en el eje Z) desde la primera tira hasta la última para compararlas con la medida real de las tiras de cartulina.

Figura 87. Resolución sistema derecho



Fuente. Los autores

En la anterior figura se resalta la diferencia entre las coordenadas Z de cada imagen que es de 2.06 mm, ésta medida deberá corresponder al cambio de altura debido al grosor de las nueve tiras de cartulina superpuestas.

Para calcular el grosor de cada tira de cartulina, se realizaron veinte medidas con un micrómetro con precisión de $\pm 10 \mu m$.

Realizando el muestreo de datos, hallando la desviación estándar y la media de las medidas, se encontró que cada tira de cartulina tiene un grosor de $213 \pm 4 \mu m$, de tal manera que las nueve tiras superpuestas deben tener un grosor total de $1917 \mu m$ que corresponden a $1,917 mm$.

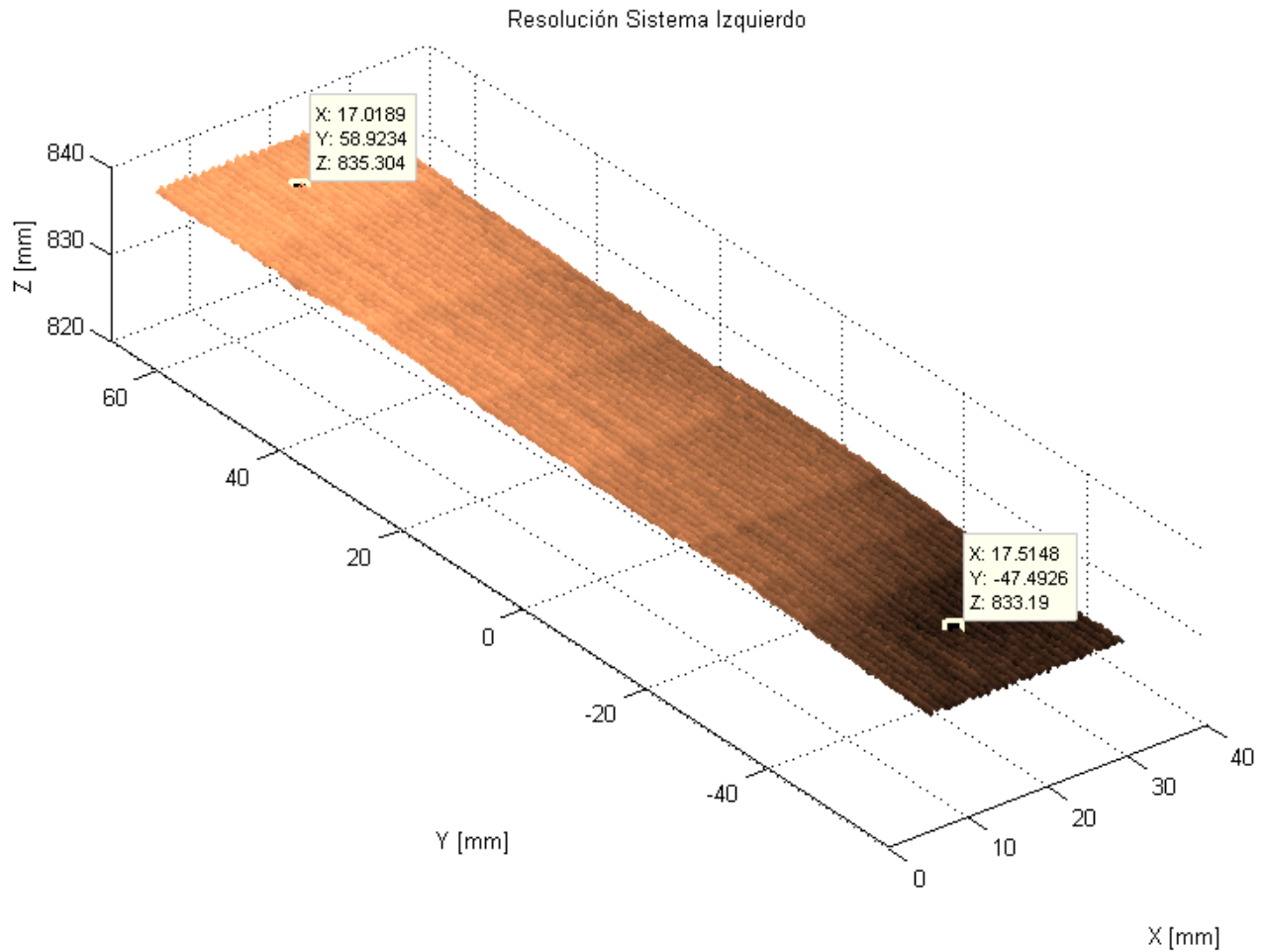
Comparando este resultado con el cambio de la coordenada Z de la figura 7.16 que es de $2.06 mm$, se determinó que el error es del 4.64% . Por lo tanto la resolución del sistema derecho se aproxima al valor de $213 \mu m$ lo que cumple y mejora el objetivo planteado de que el sistema contara con una resolución de $500 \mu m$.

A continuación se muestran los resultados de la determinación de la resolución del sistema independiente izquierdo (figura 7.16) para lo cual se realizó el mismo proceso que se describió anteriormente.

En dicha figura se resalta la diferencia entre las coordenadas Z de cada punto seleccionado que es de $2.114 mm$, ésta medida deberá corresponder igualmente al cambio de altura debido al grosor de las nueve tiras de cartulina superpuestas.

Como ya se indicó y por tratarse de las mismas tiras de cartulina, el grosor total de estas es de $1917 \mu m$ que corresponden a $1.917 mm$, con esto se determinó que el error es del 9.31% . Por lo tanto la resolución del sistema derecho se aproxima al valor de $213 \mu m$ cumpliendo y mejorando el objetivo planteado de que el sistema contara con una resolución de $500 \mu m$.

Figura 88. Resolución sistema izquierdo



Fuente. Los autores

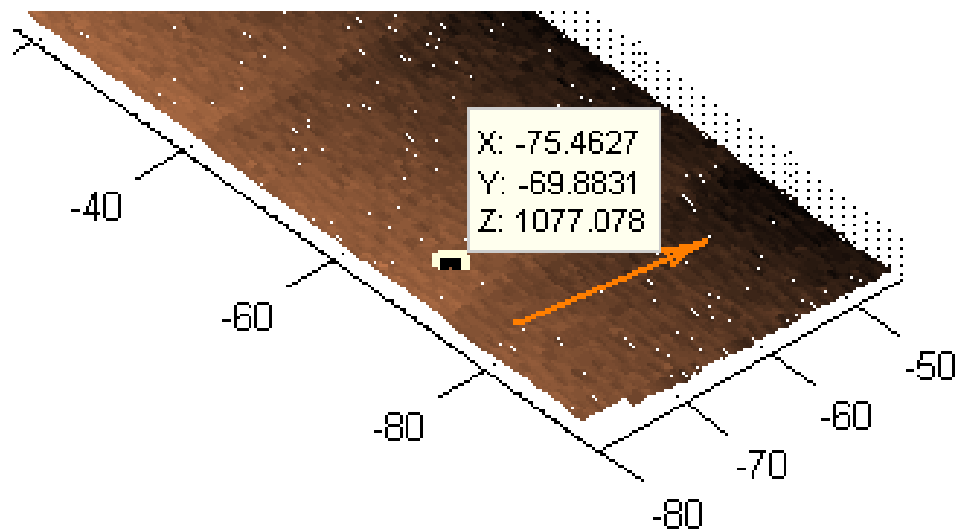
7.2.2 Precisión del sistema

En campos como la ingeniería, ciencia, industria y estadística; se puede definir como precisión a la capacidad de un instrumento de dar el mismo resultado en diferentes mediciones realizadas bajo las mismas condiciones.

Para una evaluación de la precisión del sistema implementado, se analizó la figura reconstruida de las tiras de cartulina superpuestas, y se realizó un muestreo de los diferentes valores de Z para los mismos valores de Y y de X (figura 7.16), determinando la variación en las coordenadas de Z. Para esto se seleccionaron aleatoriamente 20 puntos y determinando la desviación estándar se encontró que el sistema cuenta con una precisión de aproximadamente $76 \mu m$.

Mientras que realizando el mismo procedimiento en la figura obtenida como resolución del sistema independiente izquierdo, se encontró que la precisión de este sistema es de aproximadamente $83 \mu m$

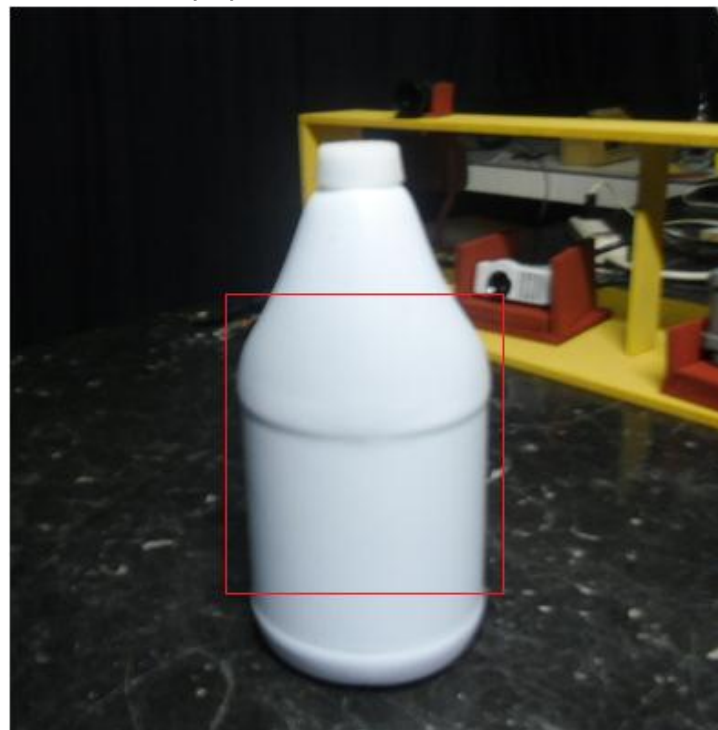
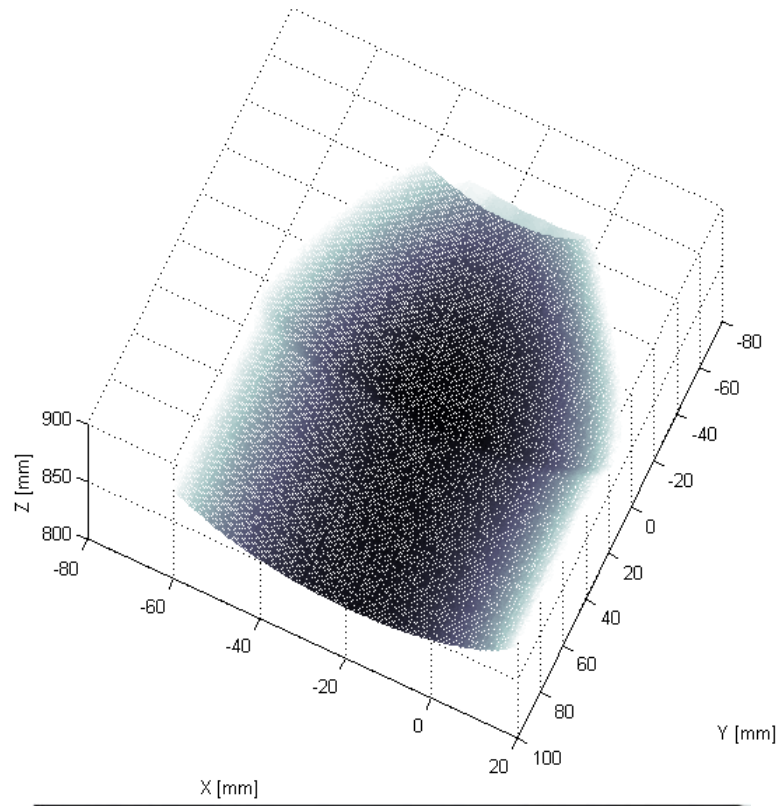
Figura 89. Muestreo de datos para la determinación de la precisión del sistema



Fuente. Los Autores

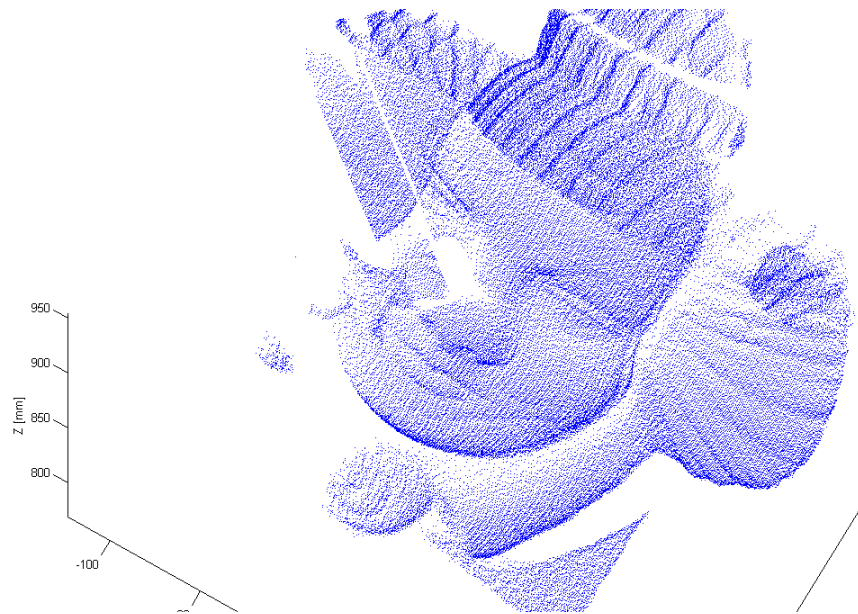
En las siguientes figuras se muestran otros objetos reconstruidos, con su respectiva imagen real.

Figura 90. Reconstrucción 3D sección cilíndrica y su imagen real



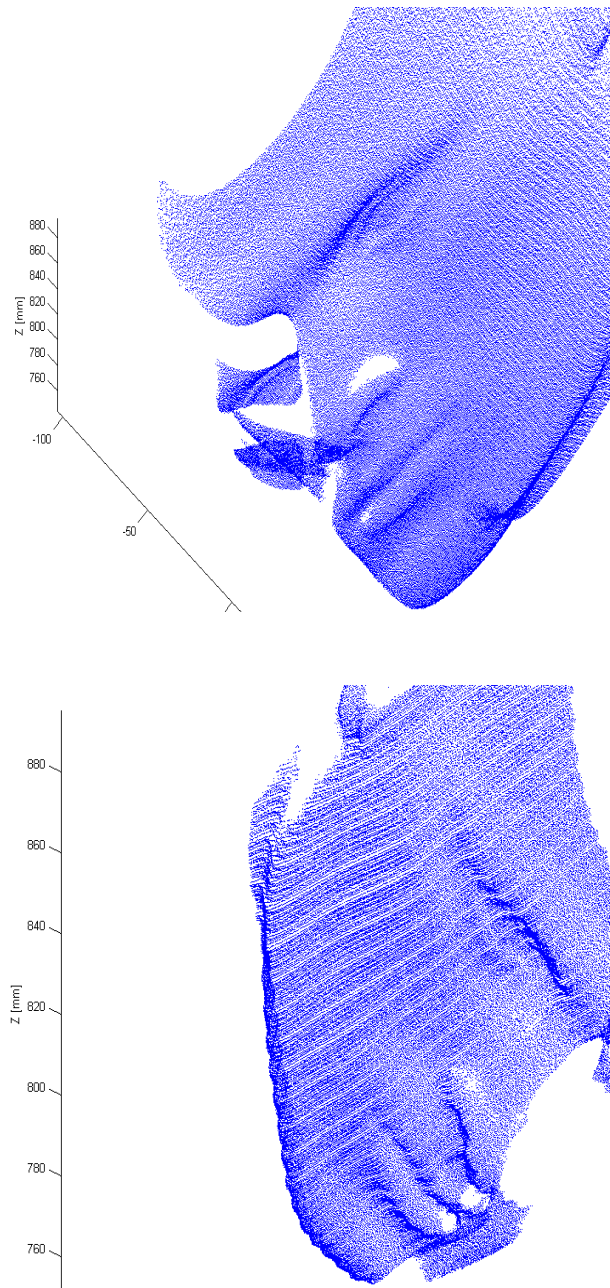
Fuente. Los Autores

Figura 91. Reconstrucción 3D en nubes de puntos y su imagen real



Fuente. Los Autores

Figura 92. Reconstrucción 3D por secciones en nubes de puntos y su imagen real





Fuente. Los Autores

8. CONCLUSIONES

- ✓ Se encontró que la técnica de proyección de franjas es una herramienta cuyas aplicaciones tienen cabida en distintos campos de la industria y de la medicina estética.
- ✓ El software diseñado e implementado durante esta investigación sentará bases para el proyecto de investigación 8186 titulado DISEÑO Y DESARROLLO DE DISPOSITIVOS DE RECONSTRUCCIÓN TRIDIMENSIONAL APLICADOS A LA EXPLORACIÓN TOPOGRÁFICA CORPORAL Y FACIAL en el cual se desea elaborar un sistema de reconstrucción tridimensional portable.
- ✓ Se diseñaron y construyeron soportes para los sistemas de observación y proyección con capacidad de rotar sobre su propio eje y capaces de minimizar las distorsiones que se presenten durante los procesos de calibración y tratamiento de las imágenes.
- ✓ Se diseñó, desarrolló e implementó un software cuya interfaz en ambiente Matlab permite agilizar y facilitar la adquisición de los datos (calibración independiente y estéreo), el registro, la integración, la segmentación y el ajuste de superficies para la reconstrucción tridimensional de objetos con características específicas.
- ✓ Se encontró que la resolución del sistema de reconstrucción tridimensional es de aproximadamente $213 \mu m$ cumpliendo con el objetivo de ser menor de $500 \mu m$.
- ✓ Se llevó a cabo satisfactoriamente el cálculo matemático y el procesamiento de las imágenes logrando el empalme de las dos imágenes reconstruidas independientemente.

- ✓ Mediante la implementación del sistema multi-captor fue posible eliminar las denominadas zonas ocultas de los objetos de reconstrucción, aumentando de esta manera la robustez del sistema de reconstrucción.

9. RECOMENDACIONES

- ✓ Seguir analizando la técnica de proyección de franjas para optimizar y encontrar una cantidad mayor de aplicaciones en la ingeniería mecánica.
- ✓ El proceso de calibración independiente preferiblemente debe realizarse en un ambiente oscuro, de manera rápida y cuidadosa, esto debido a que cualquier movimiento que sufran los sistemas de proyección u observación, afectan drásticamente los resultados obtenidos.
- ✓ La ubicación de los modelos a reconstruir preferiblemente debe coincidir con el cruce de los ejes ópticos de los sistemas de observación.
- ✓ Continuar con el proceso interdisciplinario de promover la fusión del trabajo entre distintas ramas que compacten ciencia e ingeniería.

BIBLIOGRAFÍA

- [1]. **BAEZ, David.** MATLAB con aplicaciones a la Ingeniería, Física y Finanzas. Alfaomega. Mexico. 2003. P 165-213.
- [2]. **CONTRERAS PICO, Carlos Ricardo.** Dispositivo óptico de medida 3-D con simetría cilíndrica: aplicaciones en balística. Trabajo de grado Físico. Universidad industrial de Santander, 2006. p.7
- [3]. **LILLEY F.; LALOR M. J. y BURTON D. R.** Robust fringe analysis system for human body shape measurement, Opt. Eng. 39 (1) (2000) 180–195.
- [4]. **MENESES, Jaime.** System optique pour la mesure des deformations de la voute plantaire sous charge statique. Tesis doctoral. Université de Franche-Compté, 1998.
- [5]. **PEREZ, Zayra y; ROMERO, Lenny.** Sistema óptico de reconstrucción 3D para la detección de ampollas en recubrimientos. Trabajo de grado Físico. Universidad industrial de Santander. 2004. p20-21
- [6]. **BOUGUET, Jean-Yves.** Computer Vision Research Group, California Institute of Technology, DISPONIBLE EN: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html
- [7]. **HECHT,J; LAMPRECHT,K; MERKLEIN, M; GALANULIS, K y STEINBECK, J.** Triangulation based digitizing of tooling and sheet metal part surfaces Measuring technique, analysis of deviation to CAD and remarks on us of 3D-coordinate fields for the finite element analysis, Key EngineerinMaterials 344 (2007) 847–853.

[8]. YILMAZ,S; OZUGUREL,U; BULUT,K y INCI,M. Vibration amplitude analysis with a single frame using a structured light pattern of a four-core optical fibre, *Opt. Commun*, (2005) 515–522.

[9]. DE ANGELIS, M; DE NICOLA, S; FERRARO, P; FINIZIO, A y PIERATTINI, G. Liquid refractometer based on interferometric fringe projection, *Opt. Commun*, (2000) 315–321.

[10]. ROGER ERNST y ALBERT WECKENMANN. Local wall thickness measurement of formed sheet metal using fringe projection technique, in: *Proceedings, XVII IMEKO World Congress, 1802–1805, 2003.*

[11]. HUANG,F; JIN,S y CHIANG,F. Quantitative evaluation of corrosion by a digital fringe projection technique, *Opt. Laser Eng.* 31 (5) (1999) 371–380

[12]. BURKE, J; BOTHE, T; OSTEN, W y HESS, C. Reverse engineering by fringe projection, *Proc. SPIE* 4778 (2002) 312–324.

[13]. DAVIS, J.; RAMAMOORTHY, R. y RUSINKIEWICZ, S. Spacetime Stereo: A Unifying Framework for Depth from Triangulation," *IEEE Trans. on Pattern Analysis and Machine Intelligence* , (2005).

[14]. ZHANG, Song. High Resolution Real Time 3D Shape. Tesis doctoral de filosofía en Ingeniería Mecánica. StonyBrookUniversity. 2005.

ANEXOS

Anexo A. Aplicaciones Médicas

Como se ilustró en un principio de éste trabajo, la técnica de proyección de franjas ha encontrado diversas aplicaciones en diversos campos; en éste caso, las aplicaciones biomédicas como: Mediciones 3D intra-orales¹⁵, Imágenes 3D no invasivas y control de deformaciones en las paredes vasculares¹⁶, Medición de la forma del cuerpo humano para el tratamiento de radioterapia guiada¹⁷, Medición de deformaciones de la espalda baja¹⁸, Detección y monitoreo de la escoliosis¹⁹, Inspección de cicatrices²⁰, Medición de la topografía de la piel para su uso en la cosmetología²¹, entre otras.

Como anexo a nuestro trabajo, mostraremos un trabajo de reconstrucción 3D de la superficie de la piel, denominado: *In vivo determination of skinsurfacetopographyusinganoptical 3D device*²², para su uso en la medicina estética, pudiendo realizar comparaciones de un antes y un después del uso de medicamentos o tratamientos.

¹⁵ L. Chen, C. Huang, Miniaturized 3D surface profilometer using digital fringe projection, *Meas. Sci. Techn.* 16 (5) (2005) 1061–1068.

¹⁶ K. Genovese, C. Pappalettere, Whole 3D shape reconstruction of vascular segments under pressure via fringe projection techniques, *Opt. Laser Eng.* 44 (12) (2006) 1311–1323.

¹⁷ F. Lilley, M. J. Lalor, D. R. Burton, Robust fringe analysis system for human body shape measurement, *Opt. Eng.* 39 (1) (2000) 187–195.

¹⁸ A. Hanafi, T. Gharbi, J. Cornu, In vivo measurement of lower back deformations with Fourier-transform profilometry, *Appl. Opt.* 44 (12) (2005) 2266–2273.

¹⁹ F. Berryman, P. Pynsent, J. Fairbank, S. Disney, A new system for measuring three-dimensional back shape in scoliosis, *European Spine Journal* 17 (5) (2008) 663–672.

²⁰ T. Hain, R. Eckhardt, K. Kunzi-Rapp, B. Schmitz, Indications for optical shape measurements in orthopaedics and dermatology, *Medical Laser Application* 17 (1) (2002) 55–58.

²¹ S. Jaspers, H. Hopermann, G. Sauermann, U. Hoppe, R. Lunderstadt, J. Ennen, Rapid in vivo measurement of the topography of human skin by active image triangulation using a digital micromirror device, *Skin Research and Technology* 5 (3) (1999) 195–207.

²² Tomado de <

http://www.simatecno.it/gfm/skin_scanner_3d/pdf/publicazioni/determinationskinsurfacetopography.pdf>

In vivo determination of skin surface topography using an optical 3D device

Ute Jacobi¹, Mai Chen², Gottfried Frankowski², Ronald Sinkgraven³, Martina Hund³, Berthold Rzany³, Wolfram Sterry¹ and Jürgen Lademann¹

¹Center of Experimental and Applied Cutaneous Physiology, Department of Dermatology, Charité – Medical Faculty, Berlin, Germany, ²CFMasstechnik GmbH, Teltow, Germany, ³Division of Evidence Based Medicine, Department of Dermatology, Charité – Medical Faculty, Berlin, Germany

Background/purpose: Topography of the skin surface is a mirror of the functional skin status. Therefore, its evaluation is of great interest for dermatological research. In the present study, a new optical measuring system was applied to evaluate human skin surface topography *in vivo* and non-invasively, in order to determine elastic and mimic wrinkles dependent on age, body site and therapy.

Methods: The measurements were performed on the back ($n = 5$), the flexor forearm ($n = 14$) of volunteers in different age groups and on the face of women ($n = 4$) participating in an anti-wrinkle study. The evaluation of the three-dimensional-profiles was performed determining skin surface parameters (mean roughness (R_a), mean depth of roughness (R_z), maximum roughness (R_m) and waviness (W_l)) and applying Fourier analysis.

Results: Significant increases in all surface parameters were obtained on the forearms with increasing age.

Differences between the surface topographies of forearm and back were only expressed by the waviness and the Fourier analysis. The treatment with botulinum Toxin A led to a diminution of mimic wrinkles as reflected by the furrow profiles.

Conclusions: The results show a significant dependence of the skin surface topography on the age of the volunteer and the body site measured. Particularly the waviness, the Fourier analysis and the furrow profiles reflect such dependencies. The system used also fulfills the requirements for the testing of therapies, as was demonstrated by the reduction of mimic wrinkles after treatment with botulinum Toxin A.

Key words: aging – body site – botulinum toxin A – optical measurement – skin surface topography

© Blackwell Munksgaard, 2004
Accepted for publication 2 April 2004

THE *IN VIVO* evaluation of human skin surface topography is of great interest for dermatological research because this type of topography represents the three-dimensional (3D) organization of the dermis and the subcutaneous tissue. The surface topography can be considered to be a mirror of the functional status of the skin (1). It is an expression for the possibility of the skin to respond to mechanical stimuli and threats (2).

The network of the furrows is influenced by muscle and joint movements, the environment and changes associated with aging (2–6). Various treatments exist to reverse or at least decrease these changes. In order to assess the success of a therapy or of the application of an anti-aging product, the depth, variation and distribution of the furrows and wrinkles need to be measured prior and after application of the pharmaceutical or cosmetic product.

The characterization of the skin surface topography is expressed in two-dimensional parameters used to evaluate metal surfaces (7). The maximum roughness (R_m), the mean roughness (R_a) and the mean depth of roughness (R_z) have often been used (3, 7–10). Other parameters such as waviness (W_l) (5, 11) and the depth of roughness (R_t) (3, 10) were also determined. These parameters indicate different properties of the network of the furrows (3, 11).

Usually, non-invasive methods are used to evaluate the skin surface topography. Most of these methods are based on the preparation of skin replica or reprints. The replica are measured using mechanical or optical profilometry techniques and surface image analysis (1, 7). In this way, the skin topography of skin diseases (8, 9, 11, 12) and the effect of different therapies (3, 13–15) can be determined.

A disadvantage of the replica technique is that the quality of the skin replica depends on many parameters (3, 7, 16). Besides the formation of air bubbles, the direct contact to the skin surface can change the subject of measurement. Therefore, *in vivo* methods without direct skin contact should be favored.

Recently, optical devices normally used in the technical industry (5, 10, 16) have been utilized to evaluate skin surfaces. The principle of this technique is based on the stripe projection onto the skin surface, and measuring its reflection by means of skin topography (10, 17). Until now, by the use of micro-mirrors, only one system has achieved a short measuring time of 70 ms connected with a high resolution of $1\ \mu\text{m}$ (17). Herein, we report on the application of this new system to evaluate human skin surface topography. This evaluation includes the influence of the body site and the age on the surface topography using surface parameters and, for the first time, Fourier analysis. The measurements were performed on the back and the flexor forearm of volunteers in different age groups. In addition, the effect of the treatment with botulinum Toxin A on mimic wrinkles was measured.

Materials and Methods

Volunteers

The experiments were carried out on 23 healthy volunteers aged between 23 and 65 years. The skin surface topography of the flexor forearm was measured on 14 volunteers, while the measurement was performed on the back of five volunteers. Mimic wrinkles were measured on the face of four women.

In vivo measurement of skin surface topography

The skin areas on the forearm or the back were carefully cleaned with a moist tissue and dried with a separate tissue. The measurements of the skin surface topography were performed using the PRIMOS optical 3D *in vivo* skin measurement device (GF Messtechnik, Teltow, Germany). This system is based on the so-called digital stripe projection technique, which is used as an optical measurement process. A parallel stripe pattern is projected onto the skin surface and depicted on the CCD chip of a shooting camera through a shooting optic (Fig. 1). The measurement system consists of a freely movable optical measurement

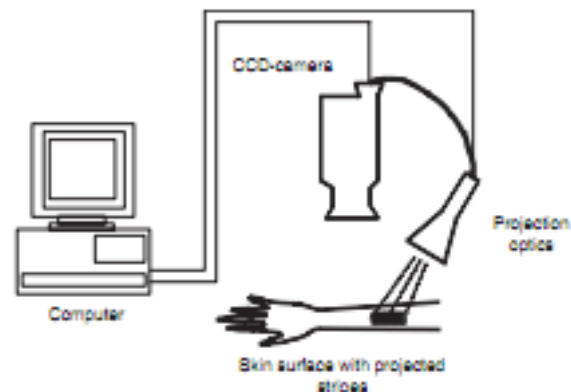


Fig. 1. Function principle of the PRIMOS optical 3D *in vivo* skin measurement device.

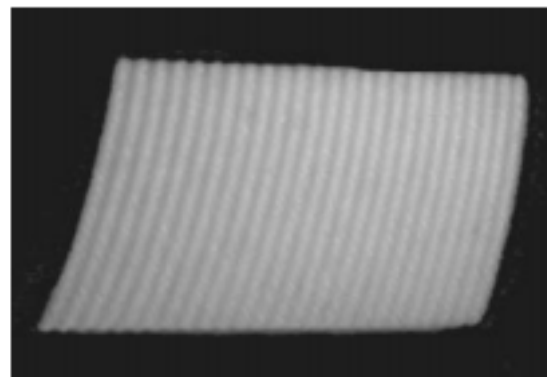


Fig. 2. Projection stripes on the forearm.

head (with an integrated micro-mirror projector, a projection optic, a shooting optic, and a CCD recording camera) together with an evaluation computer. The 3D effect is achieved by the minute elevation differences on the skin surface, which deflect the parallel projection stripes (Fig. 2). These deflections are a qualitative and quantitative measurement of the skin profile. They are digitalized and quantitatively evaluated using the software. Mathematical algorithms, originally developed and used for the highly precise optical measurement of optical and precision-mechanical parts, are employed for the evaluation providing highly precise 3D-profiles of the skin surface.

Before measurement, hairs were removed from the skin areas using scissors. The skin areas measured did not contain any visible change in the color of the skin. The area of measurement was $1.8\text{ cm} \times 1.3\text{ cm}$. Direct light from lamps or the sun was avoided. Standardized positions were used. The forearm was positioned realizing

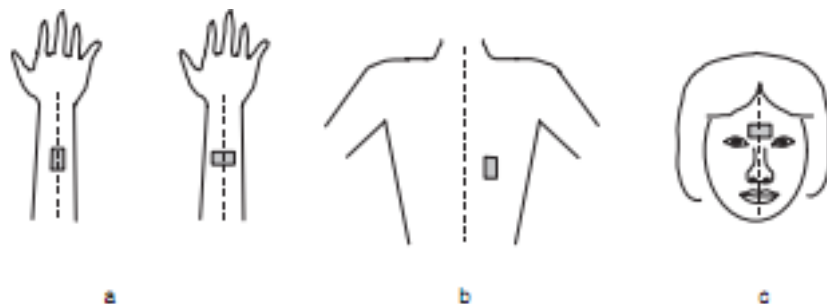


Fig. 3. Position of the measuring area (grey rectangle) to the arm, the body and the face axis (dotted line). a) On the forearm. b) On the back. c) On the face.

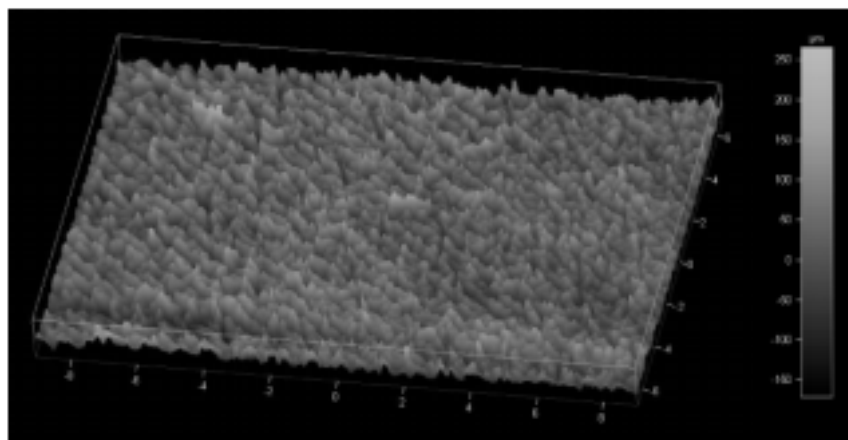


Fig. 4. 3D-profile of skin surface on the flexor forearm.

an angle of 90° between forearm and upper arm and stretched fingers. The back was measured on sitting volunteers realizing an angle of 90° between the upper part of the body and the legs. The mimic wrinkles were measured on the forehead above the nose. The chin was supported by a rest for these measurements.

For the measurements on the right flexor forearm, the rectangular measuring area was positioned parallel and longitudinal to the arm axis (Fig. 3a). On the back, the measuring area was arranged parallel to the body axis (Fig. 3b). On the forehead, the rectangular measuring area was positioned longitudinal to the face axis.

Determination of the skin surface parameters

The 3D skin profiles (Fig. 4) obtained were evaluated using the software Primos 4.0 (GF Messtechnik, Teltow, Germany). The surface parameters, maximum roughness (R_m), the mean roughness (R_a), the mean depth of roughness (R_z) and the waviness (W_t), were determined as

mean values of the parameters measured along 16 lines arranged circularly in relation to a fixed center (Fig. 5).

R_z is the arithmetical mean of the depths of roughness of five consecutive equally spaced profile sections of the same length following the same line. R_m is the maximum depth of different segment roughnesses. R_a is calculated as the mean deviation of the actual profile from the mean profile. W_t is the height of the maximum peak to valley tangents within the profile. In addition, the Fourier analysis of the 3D-profiles was performed. When applied to dermatological terms, R_a represents the roughness of the surface structure (11). R_z reflects roughness, minor furrows and anatomically preformed lines (18). W_t indicates deeper furrows and is an expression of the skin's wave structure (19).

The Fourier analysis of the profiles transforms information regarding the number, position and periodicity of the furrows into a 3D distribution of intensity. Parallel furrows show intensities in the same direction. This linear intensity

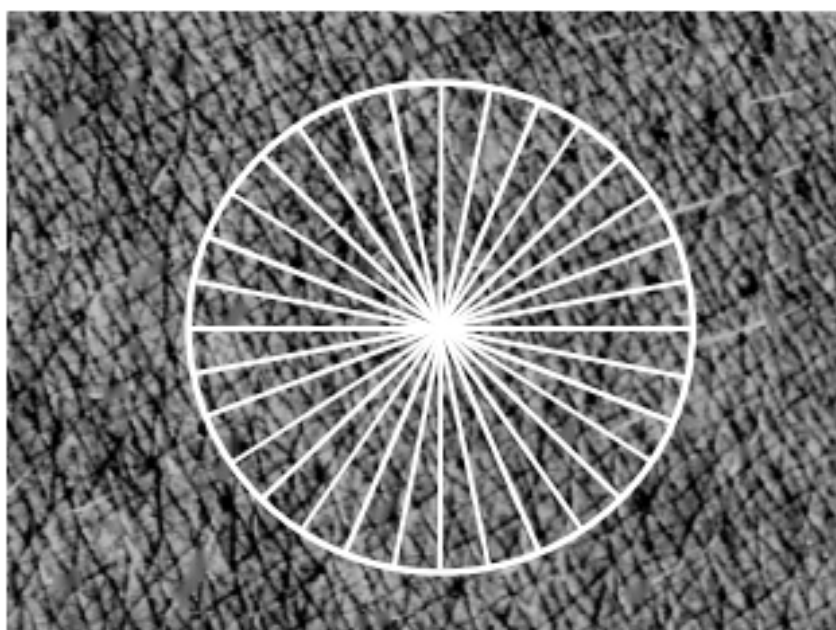


Fig. 5. Determination of the skin surface parameters.

distribution is vertically orientated to the parallel furrows on the 3D skin image. The length of the linear intensity distribution is proportional to the variance of distances between the parallel furrows.

Measurement of mimic wrinkles

In addition, the effect of injected botulinum Toxin A on mimic wrinkles was studied on four female volunteers. A total of 30–50 U botulinum Toxin A (Dysport[®], Fa. Ipsen Pharma GmbH, Ettlingen, Germany) was distributed in three to five injection points in the glabella area of each volunteer. *In vivo* measurements of the skin surface topography were performed. In addition, photographs of the forehead by a digital camera (Nikon Coolpix 990, Nikon Deutschland GmbH, Düsseldorf, Germany) were carried out as previously described in detail (20). All parameters were assessed pre- and post-treatment.

Statistics

Mean values and standard deviations were calculated using the software program Microsoft[®] Excel 97. Further statistical analysis was performed with the software program SAS[®]. Skin surface parameters R_n , R_x , R_m and W_i were analyzed between three age groups ($n = 14$, the Wilcoxon test, two-sided). In addition, the obtained skin surface parameters R_n , R_x , R_m and W_i

were compared between the forearm ($n = 6$) and the back ($n = 5$, the Wilcoxon test, two-sided). The influence of the position of the measuring area to the same body site was determined analyzing the skin surface parameters R_n , R_x , R_m and W_i ($n = 8$, the Wilcoxon test, two-sided, paired data).

Results

Measurements of skin surface topography on the forearm

3D-profiles of the skin surface topography were obtained *in vivo* (see Fig. 4). They were evaluated by determining the surface parameters R_n , R_x , R_m and W_i . The mean values for each parameter obtained on the right forearm are shown dependent on the age group (Fig. 6). The differences between these mean values are statistically significant for all parameters and all age groups ($P < 0.05$).

Additional measurements were performed by changing the position of the forearm with respect to the rectangular measuring area: from parallel to longitudinal position of the measuring area to the arm axis, as shown in Fig. 3a. A statistical analysis of the data sets (not shown) obtained for each parameter in both positions shows no significant differences ($P > 0.05$).

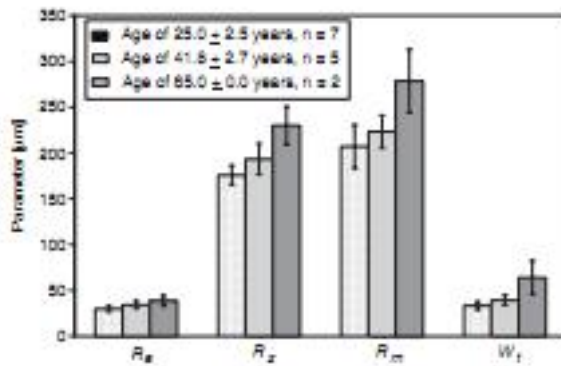


Fig. 6. Mean values and standard deviations of the skin surface parameters: mean roughness (R_a), mean depth of roughness (R_z), maximum roughness (R_m) and waviness (W_t), determined on the right flexor forearm.

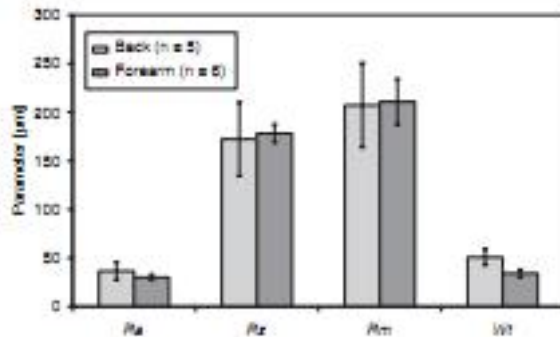


Fig. 7. Mean values and standard deviations of the surface parameters: mean roughness (R_a), mean depth of roughness (R_z), maximum roughness (R_m) and waviness (W_t), determined on the back (mean age of 27.0 ± 5.8 years) and the right flexor forearm of volunteers (mean age of 24.2 ± 1.3 years).

Measurements of skin surface topography on the back
 The surface topography of the back of young volunteers was measured. The mean values of the parameters R_a , R_z , R_m and W_t obtained are shown in Fig. 7 in comparison with the mean values observed on the forearms of volunteers at a similar age. The differences between the mean values of the parameters R_a , R_z and R_m are statistically non-significant ($P > 0.05$). There were only significant differences ($P < 0.05$) in the waviness W_t measured on both skin sites.

However, there are clear differences in the 3D-profiles of the back compared with those of the forearm (Fig. 8). Therefore, the profiles obtained for both body sites were additionally evaluated using the Fourier transformation. The intensity distribution showed an X-shaped form for the forearm (Fig. 9a), whereas on the back, the intensity was mainly distributed as a line (Fig. 9b).

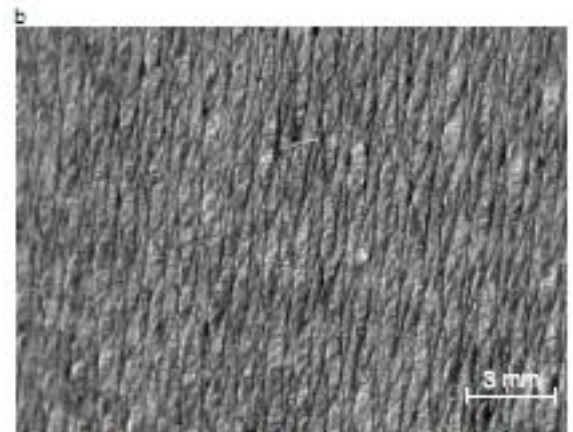
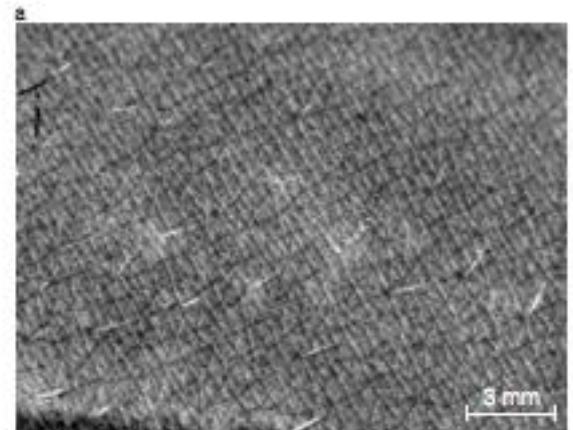


Fig. 8. 3D-profiles determined on the right flexor forearm and the back of male volunteers. (a) 3D-profile of the forearm of a male volunteer, 23 years old (\rightarrow direction of the arm axis). (b) 3D-profile of the back of a male volunteer, 30 years old (\rightarrow direction of the body axis).

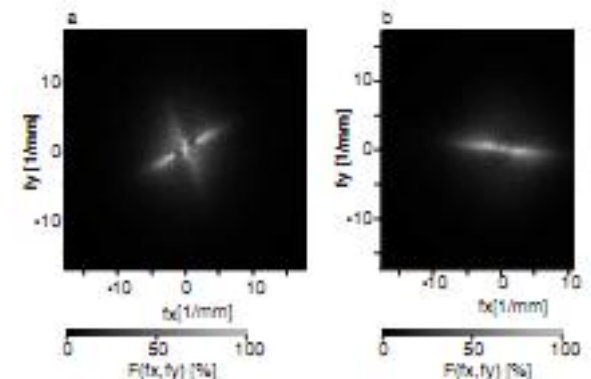


Fig. 9. Fourier analysis of the 3D-profiles determined on the right flexor forearm and the back of male volunteers. a) Fourier analysis of the 3D-profile of the forearm of a male volunteer, 23 years old (see Fig. 8a). b) Fourier analysis of the 3D-profile of the back of a male volunteer, 30 years old (see Fig. 8b).

Measurements of mimic wrinkles

The *in vivo* measurement of skin surface topography was used to evaluate the effect of a therapy with botulinum Toxin A against mimic wrinkles. Four female volunteers were treated with botulinum Toxin A to reduce mimic wrinkles in the face. Images were taken from the glabella area. The wrinkles were reduced or completely cleared. In Fig. 10, images of a volunteer pre- and post-treatment are shown in an exemplary manner. In Fig. 10a, two mimic wrinkles are clearly seen, which were diminished by the treatment with botulinum Toxin A (Fig. 10b).

The profiles of a wrinkle measured pre- and post-treatment with botulinum Toxin A are shown in Fig. 11. The depth of the wrinkle was



Fig. 10. Images taken from the forehead of a female volunteer. (a) Prior treatment with botulinum Toxin A. (b) Two weeks post-treatment with botulinum Toxin A.

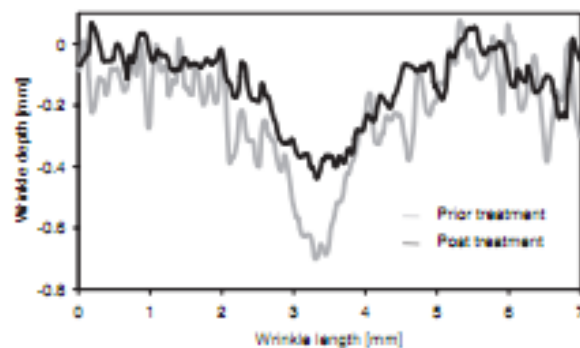


Fig. 11. Profiles of the glabella wrinkle of a female volunteer, measured prior and 4 weeks post-treatment with botulinum Toxin A.

decreased, while the width was nearly constant. This measurement could only be reproduced in one of the four volunteers, as this woman showed a small scar, which could be used as a mark for the wrinkle measurement. In the other volunteers, the treatment with botulinum Toxin A caused such a dramatic change in mimic furrows that the determination of exactly the same wrinkle profile was not possible without tattooed marks.

Discussion

The parameters R_x , R_z , R_m and W_i were determined for the 3D-profiles of the skin surface topography obtained *in vivo*. All these surface parameters determined on the forearm showed an increase with increasing age of the volunteer. A similar effect of age has also been described concerning the parameters R_x and R_m (3) and R_x , R_z and W_i (5). In contrast, no significant age-related differences in R_x (3) and in R_x , R_z and R_m (10) were observed. In these reports, for instance, the values of the parameters obtained on the forearm differ for R_z from 125 μm (3, 5) up to 240 μm (10). This variance could be caused by different factors: the posture of the body site measured influences the value of the surface parameters (3, 21), for example, by stretching the surface (5). Also, the angle between the forearm and the upper arm and the position of the fingers have influenced the results of measurements on the flexor forearm (3, 21). Therefore, standard postures should be used for the measurements and described in detail in the reports. In addition, the volunteers should be of the same age group, in order to exclude the influence of changes in the skin properties due to aging (2–5). In addition, there might be a difference between the parameters determined *in vivo* and *in vitro*. Lower values were measured at replica as performing the measurement *in vivo* using the same device (16).

Usually, the skin surface parameters were determined along parallel lines through the 3D skin surface profile. The orientation of these measuring lines to the orientation of the furrows can influence the values of the surface parameters (3). Therefore, the measurements should be performed in the same direction (3, 18). However, the preferred orientation differs depending on the volunteer and the age (22, 23). This effect can be avoided by determining the surface para-

meters following lines circularly arranged in relation to a fixed center (24). This method of evaluation has been applied in the present study. In this way, an influence on the values of the surface parameters by the orientation of these measuring lines to the orientation of the furrows could be avoided. Indeed, no significant differences between the parameters were obtained if the rectangular measuring area was positioned vertical to the arm axis, compared with the parallel orientation (see Fig. 3).

In other studies, intra-individual differences for different body sites were observed (5, 10). This can be caused by differences in the organization of the superficial dermis and the subcutaneous tissue depending on the body site (2), reflected by the surface topography (1). In the present study, the skin surface topography on the back was also evaluated. The 3D-profiles and the skin surface parameters obtained were compared with those obtained on the forearm. Particularly, the orientation of the deep furrows (primary lines classified by Hashimoto (25)) differed on both body sites. This is not reflected by the surface parameters R_x , R_z and R_m . Differences were only obtained concerning the waviness W_i .

The Fourier analysis of the profiles was performed in order to evaluate the differences. The obtained distribution of intensities (see Figs 9a, b) shows that the furrow network on the forearm is formed by two groups of primary and parallel furrows. Both groups of furrows are vertical to each other. On the back, the primary furrows are mainly located vertically to the body axis. These furrows are connected by flatter furrows that are parallel to the body axis. The variance of distances between the parallel furrows on both body areas was similar. In future, this analysis should be favored if different body sites are evaluated.

The effect of botulinum Toxin A on mimic wrinkles was additionally studied. The treatment with this substance led to a reduction in the mimic wrinkles (see Fig. 11). This result is in agreement with other studies where an improvement or diminution of facial lines could be observed after injection of botulinum Toxin A (26, 27). In the present study, this clinical wrinkle reduction was connected with a decrease in the depth of mimic wrinkles determined with the profilometric method. A change in wrinkle length, as reported by Funasaka et al. (28) was not measured due to the limited measuring area of the applied method.

Summarizing the obtained results, a simple, fast and objective *in vivo* evaluation of the skin surface topography was performed using the Primos 4.0 system. Therefore, the requirements for the evaluation of aging processes and the testing of therapies as shown for the wrinkle reduction by treatment with botulinum Toxin A are fulfilled. Besides, the normally used surface parameters and the Fourier analysis can be used satisfactorily to evaluate differences in skin surface topography.

References

1. Fischer TW, Wigger-Alberti W, Elsner P. Direct and non-direct measurement techniques for analysis of skin surface topography. *Skin Pharmacol Appl Skin Physiol* 1999; 12: 1-11.
2. Piérard GE. EEMCO guidance to the *in vivo* assessment of tensile functional properties of the skin. Part 1: relevance to the structures and ageing of the skin and subcutaneous tissues. *Skin Pharmacol Appl Skin Physiol* 1999; 12: 352-362.
3. De Paepe K, Lagarde J-M, Gall Y, Roseeuw D, Rogiers V. Microrelief of the skin using a light transmission method. *Arch Dermatol Res* 2000; 292: 500-510.
4. Levaque JL, Corcuff P, de Rigal J, Agache P. *In vivo* studies of the evolution of physical properties of the human skin with age. *Int J Dermatol* 1984; 23: 322-329.
5. Jaspers S, Hopermann H, Saueremann G, Hoppe U, Lunderstädt R, Ennen J. Rapid *in vivo* measurement of the topography of human skin by active image triangulation using a digital micromirror device. *Skin Res Technol* 1999; 5: 195-207.
6. Fenske NA, Lober CW. Structural and functional changes of normal aging skin. *J Amer Acad Dermatol* 1986; 15 (Part 1): 571-585.
7. Lévaque JL. EEMCO guidance for the assessment of skin topography. *J Eur Acad Dermatol Venereol* 1999; 12: 103-114.
8. Eberlein-König B, Schäfer T, Huss-Marp J, Darsow U, Mührenschlager M, Herbert O, Abeck D, Krämer U, Behrendt H, Ring J. Skin surface pH, stratum corneum hydration, trans-epidermal water loss and skin roughness related to atopic eczema and skin dryness in a population of primary school children. *Acta Derm Venereol* 2000; 80: 188-191.
9. Monti M, Bozzetti M, Motta S, Barbareschi M. Usefulness of surface topography analysis in psoriatic skin. *Acta Derm Venereol Suppl* 1989; 146: 81-83.
10. Piche E, Häfner H-M, Hoffmann J, Jünger M. Neue Ansätze zur 3-D-Rauhigkeitsanalyse menschlicher Haut. *Zeitschr f Haut* 2002; 77: 140-146.
11. Häbig J, Vocks E, Kautzky F, Ring J. Biophysical characteristics of healthy skin and nonlesional skin in atopic dermatitis: short-term effects of ultraviolet A and B irradiation. *Skin Pharmacol Appl Skin Physiol* 2000; 13: 174-181.
12. Linde YW, Bengtsson A, Lodén M. Dry skin in atopic dermatitis. II. A surface profilometry study. *Acta Derm Venereol* 1989; 69: 315-319.
13. Fournier N, Dahan S, Barneon G, Diridollou S, Lagarde JM, Gall Y, Mordon S. Nonablative remodeling: clinical, histologic, ultrasound imaging, and profilometric evaluation of a 1540 nm Er:glass laser. *Dermatol Sur* 2001; 27: 799-806.

Anexo B. Aplicaciones Industriales

De igual manera que en las aplicaciones médicas, las aplicaciones industriales y científicas son ampliamente usadas, algunas de éstas aplicaciones son: Análisis de vibraciones²³, Refractometría²⁴, Medición global de deformaciones en superficie libres²⁵, Medición del espesor local de pared en placas metálicas forzadas²⁶, Análisis de corrosión²⁷, Medición de rugosidad en superficies²⁸, Ingeniería inversa²⁹, Control de calidad en la manufactura de tablas de circuitos impresos³⁰, Visualización del flujo de calor³¹, entre otras.

Se anexará un trabajo enfocado a labores de ingeniería de gran importancia, éste trabajo es denominado: *FringeProjectionMeasurementSystem in Reverse Engineering*³².

²³S. T. Yilmaz, U. D. Ozugurel, K. Bulut, M. N. Inci, Vibration amplitude analysis with a single frame using a structured light pattern of a four-core optical fibre, *Opt. Commun.* 249 (4-6) (2005) 515–522.

²⁴ M. De Angelis, S. De Nicola, P. Ferraro, A. Finizio, G. Pierattini, Liquid refractometer based on interferometric fringe projection, *Opt. Commun.* 175 (4) (2000) 315–321.

²⁵ P. J. Cobelli, A. Maurel, V. Pagneux, P. Petitjeans, Global measurement of water waves by Fourier transform profilometry, *Experiments in Fluids* 46 (6) (2009) 1037–1047.

²⁶ R. V. Roger Ernst, Albert Weckenmann, Local wall thickness measurement of formed sheet metal using fringe projection technique, in: *Proceedings, XVII IMEKO World Congress, 1802–1805, 2003.*

²⁷ P. S. Huang, F. Jin, F. Chiang, Quantitative evaluation of corrosion by a digital fringe projection technique, *Opt. Laser Eng.* 31 (5) (1999) 371–380

²⁸ G. S. Spagnolo, D. Ambrosini, Diffractive optical element based sensor for roughness measurement, *Sensors and Actuators A: Physical* 100 (2-3) (2002) 180–186.

²⁹ J. Burke, T. Bothe, W. Osten, C. Hess, Reverse engineering by fringe projection, *Proc. SPIE* 4778 (2002) 312–324.

³⁰ J. Hecht, K. Lamprecht, M. Merklein, K. Galanulis, J. Steinbeck, Triangulation based digitizing of tooling and sheet metal part surfaces Measuring technique, analysis of deviation to CAD and remarks on us of 3D-coordinate fields for the finite element analysis, *Key EngineerinMaterials* 344 (2007) 847–853.

³¹ D. Ambrosini, D. Paoletti, Heat transfer measurement by a diffractive optical element fringe projection, *Opt. Eng.* 46 (9) (2007) 093606.

³² Tomado de <<http://www.springerlink.com/content/140224765p0x9847/fulltext.pdf>>

Fringe Projection Measurement System in Reverse Engineering

LIN Chao-hui(林朝辉)¹, HE Hai-tao(何海涛)¹, GUO Hong-wei(郭红卫)¹, CHEN Ming-yi(陈明仪)¹,
SHI Xuan(石璇)², YU Tao(俞涛)²

1. Applied Optics and Metrology Laboratory, School of Electromechanical Engineering and Automation, Shanghai University, Shanghai 200072, P. R. China

2. CIMS Center, Shanghai University, Shanghai 200072, P. R. China

Abstract Acquisition of physical data with high precision is a key step in reverse engineering (RE). It is an important stimulative for the progress of reverse engineering with which various digitizing devices are invented, developed and made applicable. This paper introduces a three dimensional optical measurement method based on digital fringe projection technique in RE to improve the technique through its application. A practical example is presented and the result demonstrates the applicability and feasibility of the measurement system as well as the reliability and validity of relevant methods and algorithms.

Key words reverse engineering, 3D optical measurement, fringe projection, multi-aperture overlap-scanning technique (MAOST), surface fitting.

1 Introduction

Reverse engineering as a shortcut for rapid shape design in the development of industrial products has received increasing attention^[1,2]. It is a new concept of shape design that starts from an existing physical model, which is different from the conventional forward engineering (FE). FE directly begins with an abstract form of the target product usually expressed as a blue print.

Generally, RE has three basic steps^[1,2] as shown in Fig.1, though these steps are often overlapping. Data acquisition is a crucial step of RE to digitize a physical model into a cloud of 3D coordinate points via a contact or non-contact measurement system^[3] and perform a connection operation according to the multi-view measurement if necessary. Then a process of 3D reconstruction is critical which includes data merging, data reduction, segmentation and surface fitting, etc. Finally, a CAD model is created and used to make an RE product commonly a rapid prototyped one.

Obviously, from such a development flow we can learn that acquisition of physical data with high precision is a key step in reverse engineering. As there are many methods of data acquisition and each has its strengths and weaknesses^[1-3], different digitization methods should be selected in accordance with different object attributes, application situations and precision demands. At present, the most noticeable and popular data acquisition methods are those based on optical noncontact methods. In this work, we introduce a noncontact method, *i. e.*, a 3D measurement system based on fringe projection technique^[4,5]. Since this method has many advantages such as high speed for data acquisition, full field, noncontactness, nondestructiveness, high data density and low cost, it has found more and more applications. The system setup is shown in Fig.2.

The main purpose of this paper is to present a description of application of a 3D measurement system based on fringe projection technique in reverse engineering. In Section 2, the basic principle of the measurement system is briefly introduced, and multi-view measurements and a connection method based on MAOST are described. Section 3 focuses on surface fitting method in RE. An example with a head statue as a source physical model for RE is given in Section 4. Section 5 concludes the paper.

Received Sep. 12, 2003; Revised Feb. 25, 2004

Project supported by the Science Foundation of Shanghai Municipal Commission of Science and Technology (Grant No. 011461059)

LIN Chao-hui, M.Sc. Candidate, E-mail: orientlin@163.com;

CHEN Ming-yi, Prof., E-mail: mych@sh.163.net

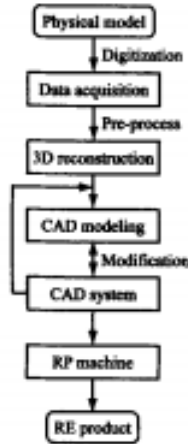


Fig.1 Common procedure of RE

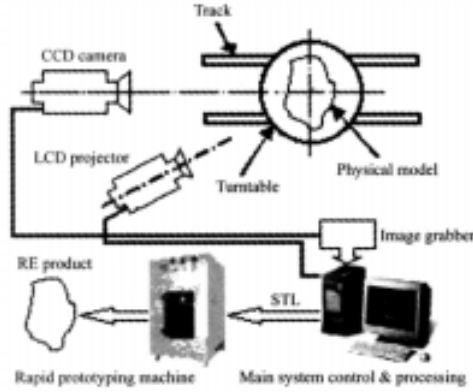


Fig.2 Layout of RE system

2 Data Acquisition via Measurement System Based on Fringe Projection Technique

2.1 Basic Principle

The data acquisition system used in this paper is a 3D measurement system based on fringe projection technique. Its basic principle is shown in Fig.3.

A deformed fringe pattern (Fig.4 (a)) captured by a CCD camera can be formulated mathematically as follows^[44]:

$$I(i, j) = a(i, j) + b(i, j)\cos[\Phi(i, j)], \quad (1)$$

where $I(i, j)$ is the intensity distribution of the fringe pattern, $a(i, j)$ the bias intensity, $b(i, j)$ the visibility or fringe contrast, and $\Phi(i, j)$ the optical phase.

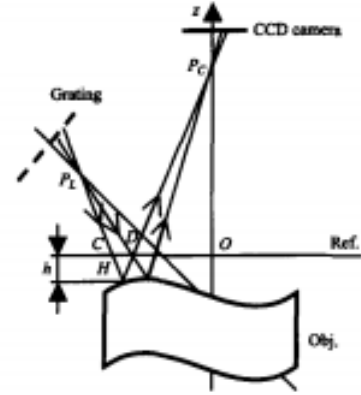


Fig.3 Optical geometry for grating projection profilometry

In this work, an N -step phase shifting algorithm^[45] is used for the optical phase distribution evaluation. The phase can be extracted by

$$\Phi(i, j) = -\arctan \frac{\sum_{k=0}^{N-1} I_k(i, j) \sin(\frac{2\pi k}{N})}{\sum_{k=0}^{N-1} I_k(i, j) \cos(\frac{2\pi k}{N})}, \quad (2)$$

where the phase is wrapped (Fig.4(b)). A phase unwrapping operation is needed to recover the absolute phase Φ_o of a tested object as well as Φ_r of a reference plane can be acquired. The phase difference (Fig.4(d)) is given by

$$\Delta\Phi(i, j) = \Phi_o(i, j) - \Phi_r(i, j), \quad (3)$$

where the phase difference $\Delta\Phi$ has a certain mapping relation with the object height distribution as follows^[4]:

$$z(i, j) = \frac{\Delta\Phi(i, j)}{c(i, j) + d(i, j)\Delta\Phi(i, j)}, \quad (4)$$

where $z(i, j)$ is the depth coordinates, and $c(i, j)$ and $d(i, j)$ are the geometry parameters of the measuring system shown in Fig.2 that is obtainable through a system calibration.

Moreover, the coordinates x and y can be determined through a calibration of the CCD camera.

2.2 Multi-view measurement and connection technique

As for multi-view measurement, because measuring results from different views are based on different local coordinates, they should be connected and accurately integrated into a global coordinate system to produce a complete result by means of coordinate

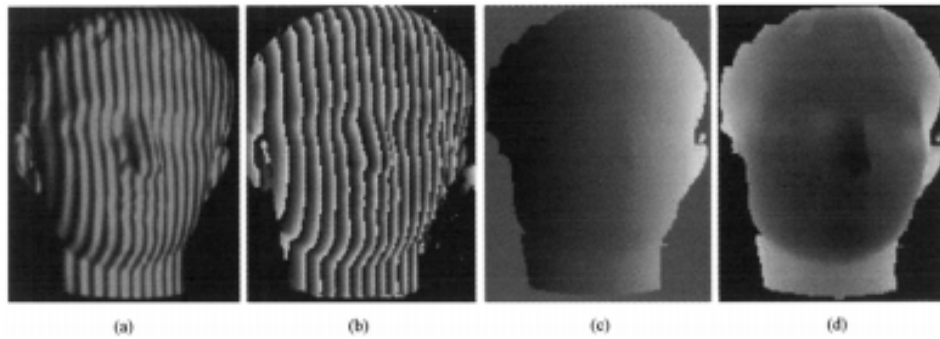


Fig.4 Single view measurement of a head status
 (a) Deformed fringe pattern; (b) Wrapped phase; (c) Unwrapped phase; (d) Phase difference

transformation. Then the key problem is to find an accurate coordinate transformation operator. For solving these problems, the multi-aperture overlap-scanning technique (MAOST)^[6-10] is an available method.

MAOST was first proposed in the Applied Optics and Metrology Laboratory, Shanghai University, in 1991 initially for the purpose of inspecting a large-scale optical flat with an interferometer^[7]. Its principle is to use the information in overlapping areas from adjacent sub-apertures or views by twos to determine their spatial relationship (relative piston and tilt). Through operations of coordinate transformation, all point data in different views are connected into a global coordinates system to remove errors generated by relative movement between the object and the sensor, and thus a full point set of the tested plane is obtained accurately. So far, application of MAOST in optical measurement has been extended to precision 3D measurement in 360°. Connection algorithm models based on Cartesian coordinates and cylindrical coordinates respectively

have been developed^[8-10].

By rotation and translation of the tested object mounted on a turntable (Fig. 2), the measurement is repeated at different views to cover the entire surface of the object. Fig. 5 shows the relative movement between the sensor and the tested object. Due to the movement errors, the real movement of the object is superposition of the nominal movement and the error movement, where the nominal movement parameters are already known, and only the error movement parameters unknown. So we can easily get a rough connection model through nominal parameter calculation and coordinate transformation. By expanding the coordinate transformation formulas into a Taylor series, we can utilize the error movement parameters to get a refined connection model. The operation is iterated until the algorithm converges or the remaining error decreases below a given threshold. Ref. [12] gives a detailed algorithm based on a cylindrical coordinate system.

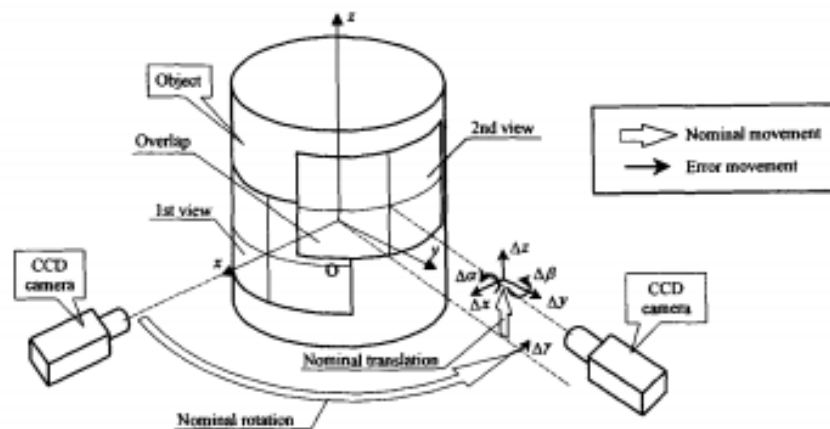


Fig.5 Movement of 3D optical sensor relative to tested object

In fact, MAOST has been proved to be useful and effective as a connection technique in 3D optical measurement^[18]. Improvement and development of relevant algorithms and methods are still in progress to meet the need for measuring more complex shapes.

3 Surface Fitting

It is essential to parameterize the discrete point cloud onto a smooth surface model for the purpose of CAD modeling in reverse engineering via a surface fitting process^[1,2].

There are three main steps to accomplish this. The one selected in this paper is shown in Fig.6. First, a surface fitting method combined with interpolation and approximation is selected.

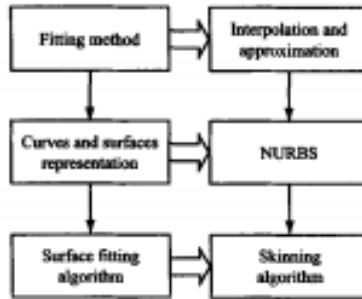


Fig.6 Main steps of surface fitting

Then the NURBS method^[2,14,15] is chosen to represent surface patch, which is mathematically expressed as follows:

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \quad (5)$$

where $P_{i,j}$ are control points; $w_{i,j}$ are weights; and $N_{i,p}(u) N_{j,q}(v)$ are B-spline basic functions.

Finally, an improved skinning algorithm^[14,16] is determined to finish surface fitting (Fig.7).

4 Experiment

A head statue made of casting plaster is employed as a physical model. Its full-digitalized point set is acquired by 10-view measurement (Fig.8) with 8 frames phase-shifting for every single-view, and connecting all views together. Afterwards the point set is re-

aligned by cross-sections along with the axial orientation of the statue and a surface construction strategy known as skinning algorithm is employed to obtain a NURBS-based CAD model used in rapid prototyping system. The ultimate work is done and the cloned product is shown in Fig.9(d).

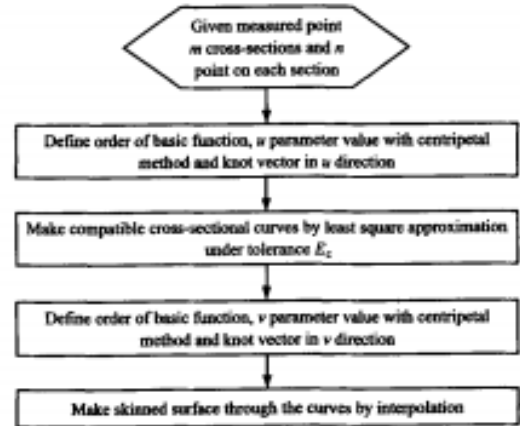


Fig.7 Flowchart of the improved skinning method

5 Conclusions

Reverse engineering has become an important tool in CAD/CAM model construction for an existing part from the measurement. It is a key step in reverse engineering to acquire physical data with high precision. And it is also an important stimulative for the progress of RE that various digitizing devices are invented, developed and made applicable. We have presented an available procedure for shape measurement of an object based on fringe projection technique and for the reconstruction of a model. The measurement method based on digital fringe projection technique in RE is especially suitable for data acquisition accurately from single-view. Multi-view measurements with the connection method based on MAOST is a costless and powerful method of measuring 3-D surface with high precision.

The measurement accuracy of the 3D optical measurement system is ± 0.025 mm/100 mm determined in comparison with a Coordinate Measuring Machine (Brown & Sharpe CMM) through measuring a cylinder^[6].

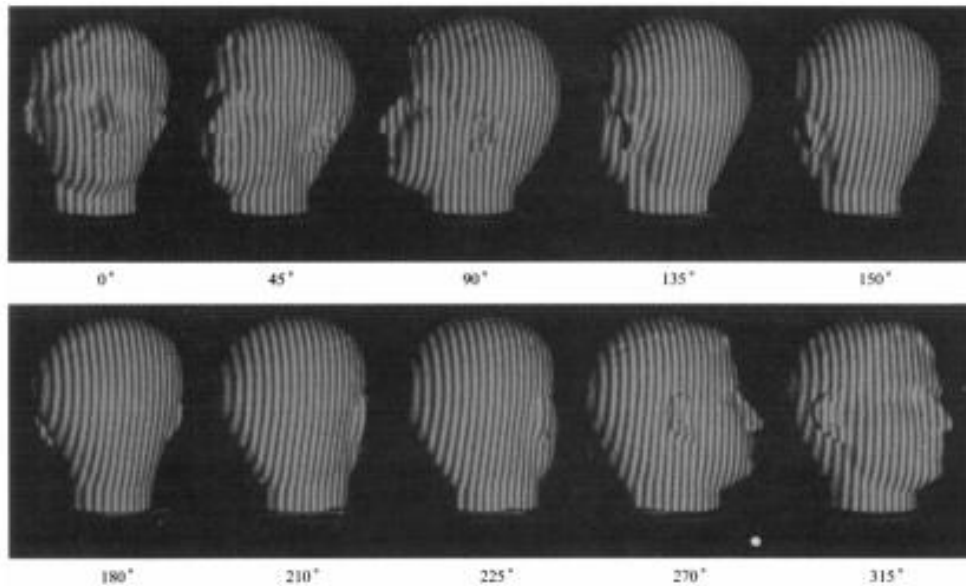


Fig.8 Deformed fringe patterns of 10 views

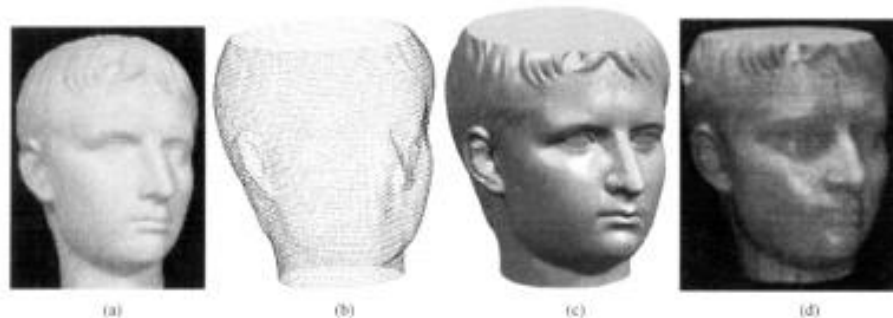


Fig.9 An example of reverse engineering (a) The original statue;(b) Acquired point cloud;(c) Result of CAD modeling;(d) The rapid prototyped part based on the measurement result

References

- [1] Chang D Y, Chang Y M. A freeform surface modeling system based on laser scan data for reverse engineering [J]. *Int. J. Adv. Manuf. Technol.*, 2002, 20(1):9-19.
- [2] Varady T, Martin R R, Cox J. Reverse engineering of geometric models - an introduction[J]. *Computer-Aided Design*, 1997, 29(4): 255-269.
- [3] Chen F, Brown G M, Song M. Overview of three dimensional shape measurement using optical method[J]. *Opt. Eng.*, 2000, 39(1):10-22.
- [4] Mitsuo Takeda, Kazuhiro Mutoh. Fourier transform profilometry for the automatic measurement of 3-D object shapes[J]. *Appl. Opt.*, 1983, 22(24):3977-3982.
- [5] Srinivasan V, Liu H C, Halloua M. Automated phase measuring profilometry of 3D diffuse objects [J]. *Appl. Opt.*, 1984, 23(18):3106-3108.
- [6] Guo Hongwei. 360-deg profilometry by multi-aperture overlap-scanning technique [D]. Doctoral Dissertation, Shanghai University, 2001, 25-33(in Chinese).

Anexo C. Herramientas de Matlab

MATLAB es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. MATLAB integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación tradicional.

MATLAB dispone también en la actualidad de un amplio abanico de programas de apoyo, especializados, denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'Toolbox' de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica.

De forma coherente y sin ningún tipo de fisuras, integra los requisitos claves de un sistema de computación técnico: cálculo numérico, gráficos, herramientas para aplicaciones específicas y capacidad de ejecución en múltiples plataformas. Esta familia de productos proporciona al estudiante un medio de carácter único, para resolver los problemas más complejos y difíciles.

ORIGEN

MATLAB nace como una solución a la necesidad de mejores y más poderosas herramientas de cálculo para resolver problemas de cálculo complejos en los que es necesario aprovechar las amplias capacidades de proceso de datos de grandes computadores.

El nombre MATLAB viene de "matrixlaboratory" (laboratorio matricial). MATLAB fue originalmente escrito para proveer acceso fácil al software matricial desarrollado por los proyectos LINPACK y EISPACK, que juntos representan el estado del arte e software para computación matricial. Hoy MATLAB es usado en una variedad de áreas de aplicación incluyendo procesamiento de señales e imágenes, diseño de sistemas de control, ingeniería financiera e investigación médica. La arquitectura abierta facilita usar MATLAB y los productos que lo acompañan para explorar datos y crear herramientas personalizadas que proveen visiones profundas tempranas y ventajas competitivas.

INICIACIÓN AL MATLAB

El Lenguaje de Computación Técnica MATLAB es un ambiente de computación técnica integrada que combina computación numérica, gráficos y visualización avanzada y un lenguaje de programación de alto nivel.

Sea cual fuere el objetivo, un algoritmo, análisis, gráficos, informes o simulación, MATLAB lo lleva allí. El lenguaje flexible e interactivo de MATLAB permite a ingenieros y científicos expresar sus ideas técnicas con simplicidad. Los poderosos y amplios métodos de cómputo numérico y graficación permiten la prueba y exploración de ideas alternativas con facilidad, mientras que el ambiente de desarrollo integrado facilita producir resultados prácticos fácilmente.

MATLAB es la fundación numérica y gráfica para todos los productos de TheMathWorks. MATLAB combina computación numérica, gráficos 2D y 3D y capacidades de lenguaje en un único ambiente fácil de usar.

Con su amplio rango de herramientas para modelar sistemas de control, análisis, simulación y procesamiento de prototipos, MATLAB es el sistema ideal para desarrollar sistemas avanzados de control. Usted puede modelar su sistema de control usando las cajas de herramientas para el diseño de controles avanzados de MATLAB - Control System, Robust Control, μ -Analysis and Synthesis, ModelPredictive Control, QTF Control Design y LMI control. Posteriores análisis y refinamientos pueden ser efectuados estableciendo una simulación interactiva en Simulink, y luego sintonizar automáticamente los parámetros usando el Nonlinear Control DesignBlockset. Finalmente, usted puede generar código C para correr en controladores incrustados con Real Time Workshop.

Combinando MATLAB con SignalProcessing Toolbox, Wavelet Toolbox y un conjunto de herramientas complementarias - tales como ImageProcessing, Neural Network, FuzzyLogic, Statistics y otras - usted puede crear un ambiente de análisis personalizado de señales y desarrollo de algoritmos DSP. Para simulación y desarrollo de prototipos usted puede agregar Simulink y el DSP Blockset para modelar y simular sus sistemas DSP, y luego usar Real-Time Workshop para generar código C para su hardware designado.

PROGRAMACION EN MATLAB

La elaboración de un programa para Matlab es muy sencilla. Basta abrir un documento de texto (se puede hacer con cualquier editor, Matlab trae uno incorporado), escribir las instrucciones tal como se pondrían en la ventana de comandos y guardar el documento con la extensión **.m** (por ejemplo: miprog.m). Matlab reconoce automáticamente los ficheros que tienen extensión **.m**: si en la ventana de comandos se introduce

»miprogram

las órdenes almacenadas en el fichero miprogram.m se ejecutarán, una tras otra. A ese recorrido a través de las instrucciones se le conoce con el nombre de *flujo*. Los comandos de un programa se denominan también *sentencias*.

El hecho de guardar una serie de instrucciones (el *código*) en un fichero tiene de por sí algunas ventajas: evita el repetir las órdenes en la ventana de comandos. Pero la potencia de un programa se pone de manifiesto si en él se contienen instrucciones que controlan qué sentencias se ejecutan en cada circunstancia. Así, un programa puede "decidir", llegado a un punto, ejecutar unos comandos u otros en función del valor que tome una variable, por poner un caso.

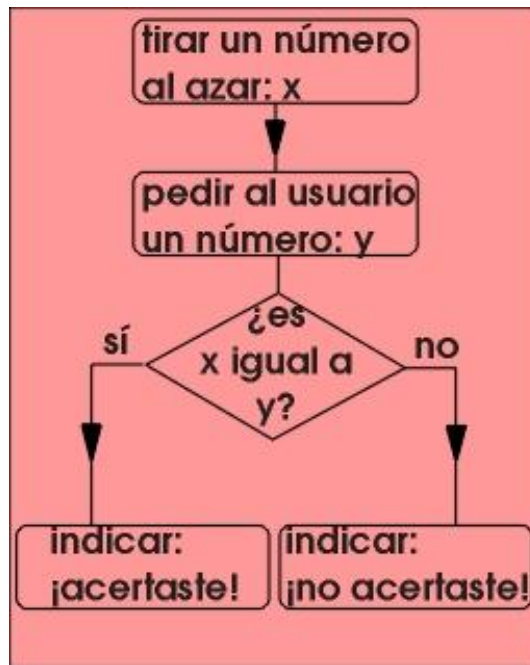
Control de flujo

Al proceso de toma de decisiones se le llama control del flujo

Antes de ponerse a escribir el programa, sobre todo si es un poco largo o complejo, es útil trazar un esquema, en lenguaje corriente, de lo que se quiere hacer. A eso se le denomina un **diagrama de flujo**. Si lo que se pretende está claro, escribir el programa se reduce a traducir el diagrama de flujo del castellano al lenguaje de Matlab.

Veamos un caso simple. Construyamos el diagrama de flujo de un programa que escoja un número al azar del 0 al 9 y nos pida que lo adivinemos. Una vez escogido el número, el programa debe informar si hemos acertado o no. El diagrama de flujo correspondiente es

Diagrama de flujo.



En un diagrama de flujo ha de buscarse la claridad, la concisión, el rigor. Con un buen diagrama de flujo, gran parte del trabajo está hecho. Puede ser tentador comenzar a escribir directamente el programa en el editor, con la esperanza de hacerlo todo bien a la primera. Suele ser más eficaz, sin embargo, escribir antes el diagrama de flujo, para luego intentar pasarlo al lenguaje informático elegido.

Condiciones

Son las preguntas básicas a las que se puede responder sí o no

Para implementar el control del flujo, son muy útiles unas expresiones que permiten comparar dos variables entre sí o una variable con un valor fijo. En un programa, a veces es necesario preguntarse: *¿es x mayor que y? Pues si x es mayor que yo, entonces..., y si no, ...*; o bien: *caso de que la variable x sea menor que cero, hacer tal cosa.*

Las condiciones se construyen con **operadores relacionales**, como son los siguientes:

>	mayor que
<	menor que
==	igual que
~=	diferente que
<=	menor o igual que
>=	mayor o igual que

Una condición, como se ha indicado, sólo admite dos respuestas: verdadero o falso. Así, a la pregunta ¿es x mayor o igual que y?, que en el lenguaje de Matlab se expresa $x \geq y$, sólo caben dos respuestas posibles: sí, es cierto, x es mayor o igual que y; o bien, no, es falso, x no es mayor o igual que y.

Dos observaciones. La primera: el operador relacional *igual que* se construye con dos símbolos ==. El motivo es el siguiente: si se escribe $x=3$, el resultado es que se le asigna a x el valor 3. No se hace ninguna pregunta. Si lo que se desea es preguntar ¿es x igual a 3?, no se puede escribir $x=3$, pues en este caso x tomaría el valor 3, independientemente de lo que valiera antes. En cambio, $x==3$ es el análogo en lenguaje informático a la pregunta: ¿es x igual a 3?. Tras la pregunta $x==3$ la variable x sigue valiendo lo mismo que antes. La segunda observación es una pequeña astucia: la tilde ~ del operador *diferente que* se consigue apretando la tecla Alt y -sin soltarla- escribiendo con el teclado el número 126. En algunos sistemas operativos, también sirve apretar AltGr y, sin soltarla, apretar la tecla 4. Las condiciones se pueden agrupar en construcciones lógicas: condición1 Y condición2, condición1 O condición2, etcétera. Consúltese el manual de Matlab en lo relativo a este tópico.

En resumen: con las expresiones relacionales se pueden realizar preguntas de respuesta unívoca (verdadero o falso), que permitan luego tomar una decisión.

Ramificaciones

Dependiendo de una condición, el programa decide el curso que seguirá

El resultado de una condición permite tomar una decisión, es decir, si la condición es verdadera se ejecutan determinadas sentencias. El diagrama de flujo se ramifica:

Ramificación if



La ramificación más simple, expresada en este diagrama de flujo, se obtiene con la siguiente sintaxis:

```
if (condición)  
    Sentencias  
end
```

(lo que va en cursiva, hay que sustituirlo por las expresiones adecuadas; if y end son *palabras clave* del lenguaje informático, y no se pueden utilizar para otra cosa, p. ej. una variable no puede -no debería- llamarse if).

Un caso concreto:

```
if(length(sitios)>1)  
    recta=polyfit(x,y,1);  
end
```

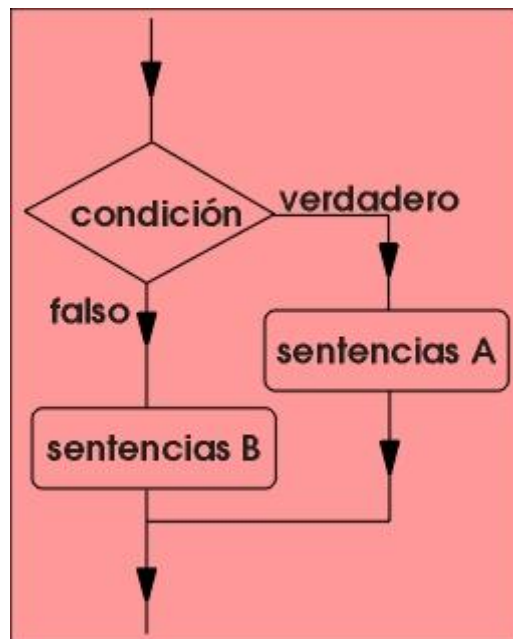
Leído en lenguaje corriente: si la longitud del vector sitios es mayor que 1, se realiza el ajuste lineal indicado en la instrucción `recta=polyfit(x,y,1)`. Caso contrario (si la longitud del vector sitios es menor o igual a 1) esa instrucción no se ejecuta (y el programa sigue en la instrucción que venga después de `end`).

Existe la posibilidad de ejecutar ciertas sentencias si la condición es verdadera, y otras diferentes si la condición es falsa:

```
if (condición)
    sentencias A
else
    sentencias B
end
```

Dicho de otra manera: si la condición se cumple, se ejecutan las *sentencias A*; si no, se ejecutan las *sentencias B*.

Diagrama de flujo de una ramificación if / else



Una vez se hayan ejecutado las sentencias que correspondan (las sentencias A si la condición es verdadera, las sentencias B en caso contrario) el programa continúa con las instrucciones que vengan después de `end`.

Es de señalar que las ramificaciones se pueden *anidar* (una condición dentro de otra: dentro de las *sentencias A* o de las *sentencias B* en el anterior diagrama de flujo). El código se vuelve más complejo. También se pueden construir ramificaciones con varias posibilidades: la sintaxis hace uso de las palabras clave `if / elseif / elseif / ... / else / end`. Se remite para una explicación más detallada a la ayuda de Matlab (término de búsqueda: `elseif`).

Otra posibilidad de ramificación múltiple la ofrece la construcción **switch**. La sintaxis es:

```
switch variable
case valor1,
    sentencias A
case valor2,
    sentencias B
case ...
...
end
```

(Como antes, lo escrito en cursiva debe sustituirse por las expresiones adecuadas). Las palabras clave son `switch`, `case`, `end`.

La ramificación `switch` opera de la siguiente manera. Al llegar a la expresión `switch variable`, si *variable* tiene el valor *valor1* se ejecutan las *sentencias A*; si *variable* toma el valor *valor2*, las *sentencias B*; y así sucesivamente. Es importante notar que la variable sólo debe tomar unos pocos valores: *valor1*, *valor2*, etc. para que el programa se ramifique en unas pocas ramas. No tiene sentido intentar una ramificación `switch` con una variable que pueda tomar un número infinito de valores.

Bucles

¿Varias secuencias que se ejecutan repetidas veces? Un bucle

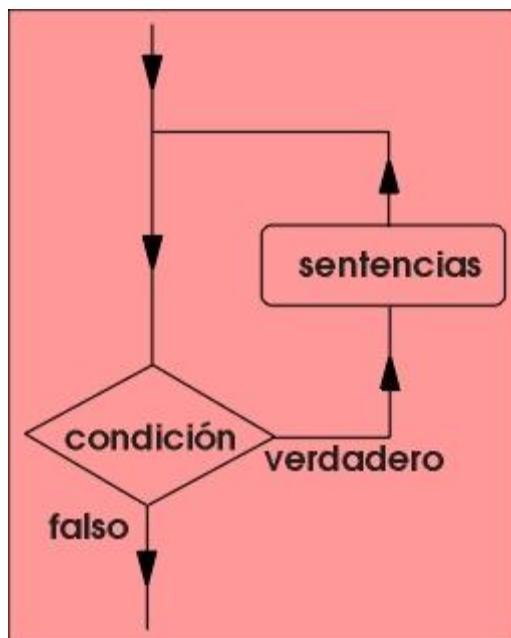
Hay ocasiones en las que es necesario repetir el mismo conjunto de instrucciones muchas veces, cambiando algunos detalles. Pongamos un caso. Sea un vector

Obsérvese que un bucle como el indicado se implementa un número fijo de veces: desde inicio hasta fin de paso en paso. En ocasiones, sin embargo, no se sabe de antemano cuántas veces habrá que ejecutar las sentencias del bucle. Por ejemplo: si es necesario repetir una serie de sentencias hasta que se cumpla una determinada condición, y no se sabe a priori cuántas veces será necesario realizar esas operaciones. En ese caso se emplea un bucle **while**:

```
while(condición),  
    sentencias  
end
```

Este bucle ejecuta las *sentencias* mientras la *condición* sea verdadera.

Diagrama de flujo de un bucle while



Es posible sustituir la condición por una variable. En efecto: una variable que toma el valor cero corresponde a una condición falsa. Si la variable toma un valor diferente de cero, es equivalente a una condición verdadera. Así, se puede escribir

```
x=10;
while(x)
    sentencias
    x=x-1;
end
```

Para $x=10$, la "condición" es verdadera puesto que x es diferente de cero. Nótese que el contador x hay que modificarlo manualmente (línea $x=x-1$) puesto que, al revés que lo que ocurre con el bucle `for`, este no gestiona ningún contador. En cuanto x tome el valor cero, la "condición" es falsa y el bucle acaba.

Atención: es fácil caer en bucles infinitos. En el ejemplo anterior, si falta la línea $x=x-1$ y las *sentencias* no modifican el valor de x , la "condición" siempre será cierta (pues $x=10$) y el programa nunca saldrá del bucle: ejecutará una y otra vez las *sentencias*. El programa se "cuelga", y hay que interrumpirlo desde el teclado apretando las teclas Ctrl+C.

Estilo

Un programa elegante suele ser un buen programa

Empecemos diciendo que, en general, para realizar una tarea particular se pueden emplear muchos programas diferentes que den el resultado correcto. ¿Cuál es el más eficiente? ¡No es una pregunta trivial! Hay que ir probando para mejorar la velocidad. Con la práctica -y el estudio- se descubren las maneras más rápidas de implementar las instrucciones entre todas las posibles.

La confección de programas se vuelve frecuentemente una tarea rutinaria donde el orden facilita las cosas. Repasar un programa una y otra vez, modificando poco a poco el código para depurarlo, exige una cierta disciplina. Aunque cada uno tenga sus preferencias, es conveniente desarrollar algunos hábitos. Me atrevo a recomendar algunos:

- El programa debe resultar de **fácil lectura**. Así, pasado el tiempo, es más sencillo revisarlo. Los nombres de las variables, en la medida de

lo posible, deben ser auto-explicativos. La siguiente porción de código, escrita de dos maneras equivalentes, lo muestra. ¿Por qué llamar x o N a una variable? Los nombres abstractos restan claridad al programa.

<pre>N=10; x=rand(1,N); y=zeros(1,N); for i=1:N, y(i)=sum(x(1:i)); end</pre>	<pre>num_pts=10; prob=rand(1,num_pts); prob_acum=zeros(1,num_pts); for indice=1:num_pts, prob_acum(indice)=sum(prob(1:indice)); end</pre>
--	---

Ambos códigos son equivalentes y se ejecutarán igual de rápido (los nombres largos de las variables no influyen en la velocidad de ejecución).

- Introducir **comentarios** en el programa. Los comentarios son líneas que no se ejecutan, en las que se escriben aclaraciones explicativas. Para que una línea no se ejecute, basta escribir al principio de ella el símbolo %. Así:

```
%escala: 102 pixels = 8 mm
d_m=8/102;
```

Un programa con comentarios es más fácil de entender.

- Suele ser bueno **definir las variables al principio**. Ello evita tener que buscarlas a lo largo del código para cambiar su valor cuando sea necesario. Además, si es posible, es mejor definir los vectores y matrices al principio con su dimensión adecuada. En el caso de que haya que ir rellenando los valores de un vector, el

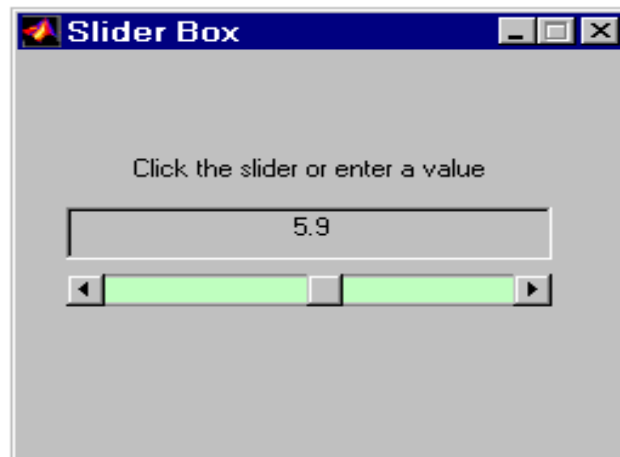
programa va más rápido si se define el vector vacío al principio (con el comando ones o zeros) que ir añadiendo componentes al vector conforme se van calculando.

- Para ver dónde empiezan y dónde acaban los bucles y ramificaciones, es costumbre **intentar** (introducir una sangría) en las líneas contenidas en ellos. Eso resulta de especial ayuda en los bucles y condiciones anidados.

GUIDE EN MATLAB

Matlab Guide* es un entorno de programación visual que ofrece Matlab para poder realizar y ejecutar programas de Simulación a medida de forma simple , tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

Por ejemplo una aplicación se puede ver de la siguiente manera



Ejecución: Desde la ventana de comando del Matlab se debe ejecutar el comando *guide*.

Esto abre la consola de edición de la parte grafica de la aplicación a implementar (.fig), es decir, colocar botones, cuadros de dialogo, graficas, texto, etc.

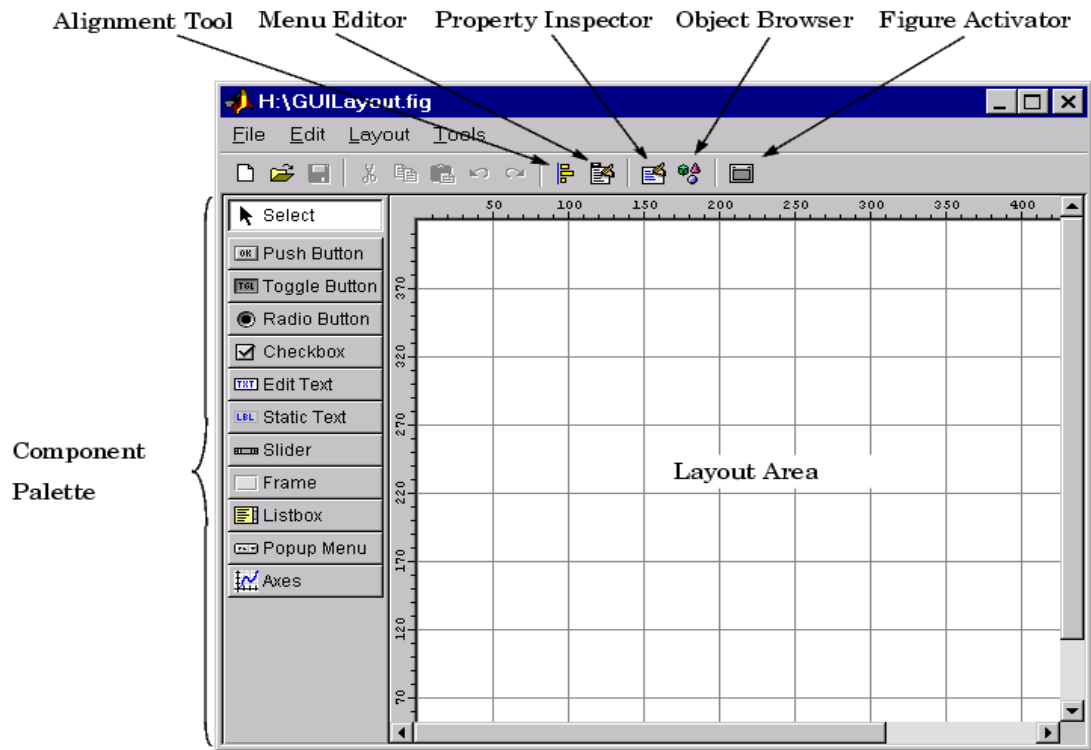
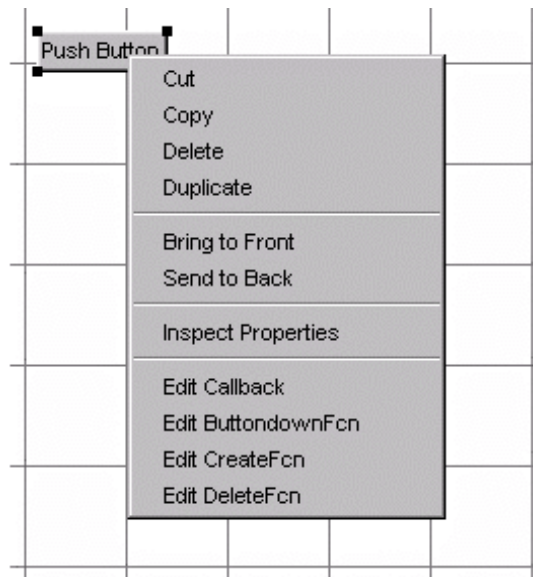


Figura 2

Cada uno de estos elementos tienen un conjunto de propiedades a las cuales podemos acceder con el botón derecho del mouse, una vez clickeado este aparece el siguiente cuadro:



Para editar las propiedades de cada elemento seleccionamos la opción *Properties Inspector* y se abre una consola (la cual variará según que elemento se esté editando) con todas las propiedades que podemos editar, ej color, posición, tamaño, font, etc.

Una de las opciones de mayor interés para nosotros en la figura anterior es *EditCallback*. Esta última abre el archivo .m asociado (ejecutable Matlab) y nos posiciona en la sección del programa que corresponde a la subrutina que se ejecutara cuando se realice una determinada acción sobre el elemento que estamos editando.

Por ejemplo para el botón 1, EditCallback nos posiciona en la siguiente parte del programa:

```
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
disp('pushbutton1 Callback not implemented yet.')
```

¿Cómo funciona una aplicación Guide? Consta de dos archivos uno .m (ejecutable) y otro. fig la parte gráfica. Las dos partes están unidas a través de las subrutinas callback. Una vez que se graba los archivos desde la consola de

emisión (si salvamos la .fig automáticamente lo hace el .m asociado) podemos ejecutar el programa en la ventana de comando de Matlab solamente escribiendo el nombre del archivo solamente. Por ejemplo si guardamos un archivo ej.fig y ej.m escribiendo *ej* y presionando *enter* se ejecuta el programa.

El archivo .m que se crea tiene una estructura predeterminada. Consta de un encabezado y a continuación viene el código correspondiente a las siguientes subrutinas. Por ejemplo una aplicación cuya figura tenga 3 botones, un gráfico y un cuadro de edición tendrá un archivo .m con la siguiente estructura inicial (todavía no se agrega el código de la subrutina) como la siguiente:

```
function varargout = untitled1(varargin) ← Encabezado
% UNTITLED1 Application M-file for untitled1.fig
% FIG = UNTITLED1 launch untitled1 GUI.
% UNTITLED1('callback_name', ...) invoke the named callback.
```

```
% Last Modified by GUIDE v2.0 20-Aug-2002 19:57:33
```

```
if nargin == 0 % LAUNCH GUI
```

```
    fig = openfig(mfilename,'reuse');
```

```
    % Use system color scheme for figure:
```

```
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
```

handles, Puntero a todas variables y elementos de la aplicación

```
    % Generate a structure of handles to pass to callbacks, and store it.
```

```
    handles = guihandles(fig);
```

```
    guidata(fig, handles);
```

guidata(), comando para guardar las variables de la aplicación

```
if nargin > 0
```

```
    varargout{1} = fig;
```

```

        end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
    catch
        disp(lasterr);
    end
end
end

```

%| ABOUT CALLBACKS:

*%| GUIDE automatically appends subfunction prototypes to this file, and
 %| sets objects' callback properties to call them through the FEVAL
 %| switchyard above. This comment describes that mechanism.*

%|

%| Each callback subfunction declaration has the following form:

%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)

%|

%| The subfunction name is composed using the object's Tag and the

%| callback type separated by '_', e.g. 'slider2_Callback',

%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.

%|

%| H is the callback object's handle (obtained using GCBO).

%|

%| EVENTDATA is empty, but reserved for future use.

%|

%| HANDLES is a structure containing handles of components in GUI using

%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This

%| structure is created at GUI startup using GUIHANDLES and stored in

%| the figure's application data using GUIDATA. A copy of the structure

%| is passed to each callback. You can store additional information in

<p>Comentario, se puede borrar</p>
--

```

%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>(<SUBFUNCTION_NAME>, gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.

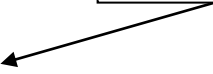
```

```

% -----
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
disp('pushbutton1 Callback not implemented yet.')

```

Subrutina Boton 1



```

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
disp('pushbutton2 Callback not implemented yet.')

```

Subrutina Boton 2

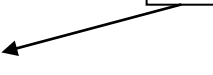


```

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
disp('pushbutton3 Callback not implemented yet.')

```

Subrutina Boton 3



```

% -----
function varargout = edit1_Callback(h, eventdata, handles, varargin)

```

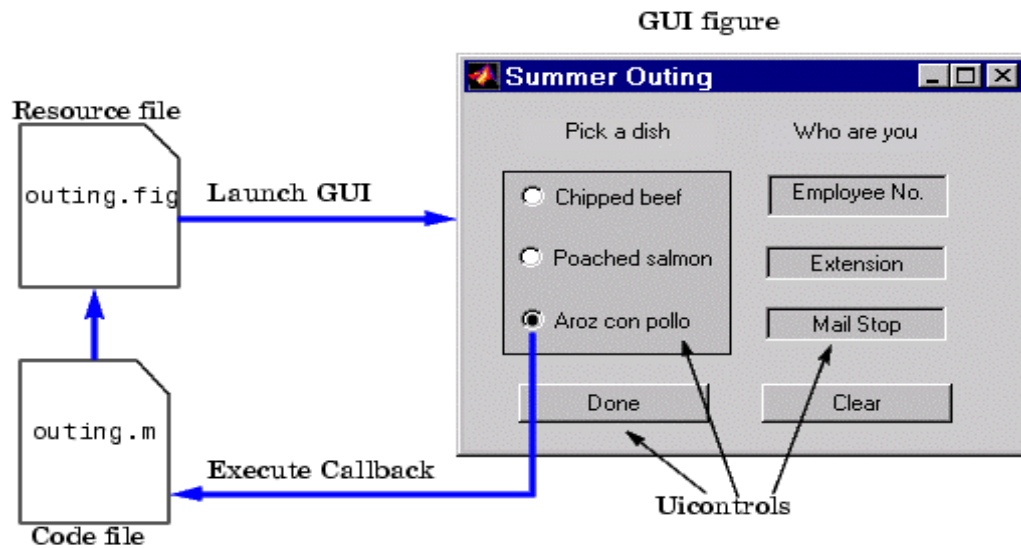
Subrutina Cuadro de Edición 1



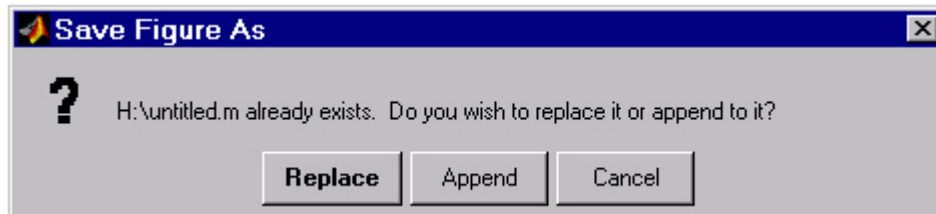
```
% Stub for Callback of the uicontrol handles.edit1.
```

```
disp('edit1 Callback not implemented yet.')
```

La siguiente figura muestra la interacción entre la figura (.fig) y el archivo (.m):



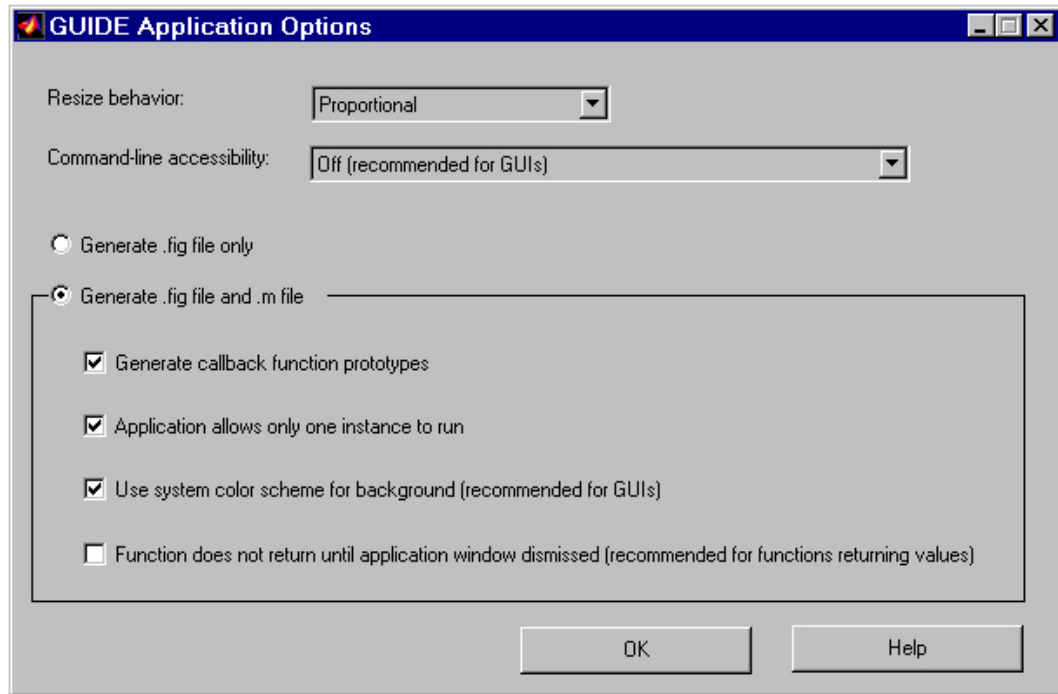
Observación sobre Salvar: Si guarda una modificación del archivo cuando aparece el cuadro de dialogo (ver figura)que pregunta si hacemos Replace o Append debemos elegir Append, si se selecciona Replace borra todo el archivo .m es decir lo pone a cero, no utilizar Replace.



Configuración Estándar de la Aplicación:

Sobre las propiedades del elemento Figure:

ApplicationOptions: Al seleccionar esta opción se abrirá la siguiente ventana:



Se debe configurar como:

ResizeBehavior: UserSpecified

Command Line Accesibility=on

Generate .fig and .m files

Generate Cllaback Prototype Functions

Applications allows only one instance to run

Use system color for background

Property Inspector:

Para hacer una aplicación que ocupe toda la pantalla y que nos deje ver la barra de tareas de Windows debemos seleccionar:

Position : x= 0.0

y=2.5

width= 205.6

height=54.8

Si se desea utilizar varias graficas:

Nextplot=add

Tag: Nombre interno que se le asigna

Sobre las propiedades del elemento Axes (Grafica):

Color: Red=1

Green=1

Blue=0.75

Nextplot=Replace Children

Tag: nombre interno que se le asigna.

Sobre las propiedades del elementoEdit (Cuadro de Edición):

BackgroundColor: Red=1

Green=1

Blue=0.75.

FontAngle : Italic

FontSize: 14.00

Font Name : MS Sans Serif

Fontwight :bold

ForegrounColor:

Red=0.0

Green=0.0

Blue=0.75

Tag:Nombre interno que se le asigna.

Sobre las propiedades del elementoButton(Botón):

BackgroundColor: Red=.75

Green=.75

Blue = .75

String= Es el texto que aparece dentro del botón

Tag: Nombre interno que se le asigna.

Sobre las propiedades del elemento Slider (Barra de desplazamiento):

BackgroundColor: Red=.75

Green=.75

Blue = .75

Tag: Nombre interno que se le asigna.

Slider Step: Con esta propiedad se define el valor del paso al clickear en la fecha de incremento o decremento.

Manejo de datos entre los elementos de la aplicación y el archivo .m

Todos los valores de las propiedades de los elementos (ej, color, valor, posición, string, etc.) y los valores de las variables transitorias del programa se guardan en una estructura (estructuramatlab: puede almacenar matrices, string, arreglos, vectores, etc.) los cuales son accedidos mediante un único y mismo puntero para todos estos. El nombre del puntero se asigna en el encabezado del archivo .m, tomando por ejemplo el programa listado anteriormente el puntero se asigna en

handles = guihandles(fig);

handles , es entonces nuestro puntero a los datos de la aplicación .

Esta definición de puntero es salvada con la siguiente instrucción

guidata(fig, handles);

guidata, es entonces la función para salvar los datos de la aplicación.

Importante: `guidata` es la función que guarda las variables y propiedades de los elementos en la estructura de datos de la aplicación, entonces como regla general en cada subrutina se debe escribir en la última línea lo siguiente,

```
guidata(gcbo,handles);
```

Esto nos garantiza que cualquier cambio o asignación de propiedades o variables quede salvado.

Por ejemplo si dentro de una subrutina una operación dio como resultado una variable *fi* para poder utilizarla desde el programa u otra subrutina debemos salvarla de la siguiente manera,

```
handles.fi=fi;  
guidata(gcbo,handles);
```

la primera línea crea la variable *fi* a la estructura de datos de la aplicación apuntada por *handles* y la segunda graba el valor.

get y set

La transferencia u obtención de los valores de las propiedades de los elementos se realiza mediante las funciones `get` y `set`. Por ejemplo si queremos que la variable *fi* tenga el valor del slider escribimos

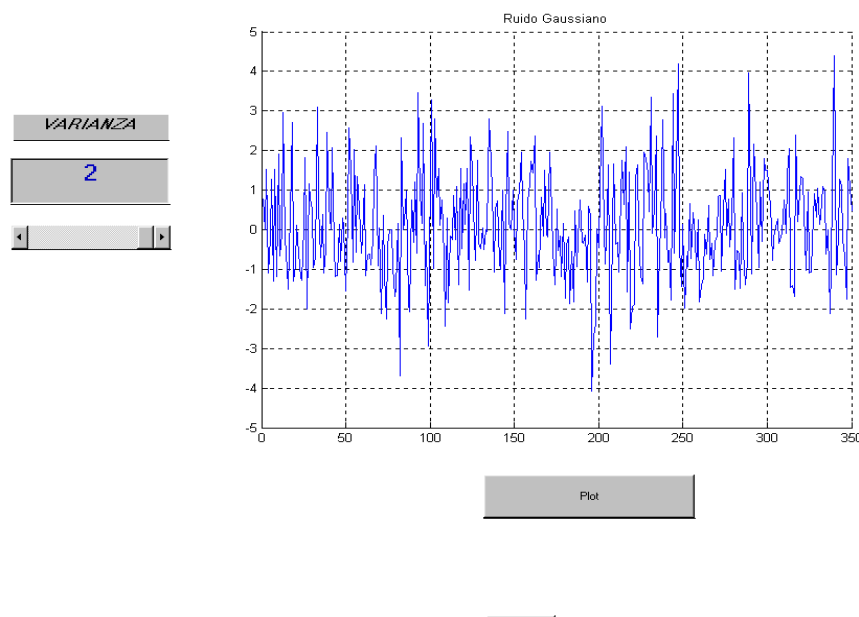
```
fi=get(handles.slider,'value');
```

Observar que siempre se obtienen los datos a través del puntero *handles* .

Ahora si quisiéramos hacer al revés y asignarle el valor la variable *fi* al slider debemos escribir

```
set(handles.slider,'value',fi);
```

Para aclarar todos los conceptos anteriores veremos una aplicación de ejemplo cuya presentación y archivo .m se listan a continuación. Lo que realiza este programa es graficar un ruido Gaussiano (350 muestras) cuya varianza se elige con la barra de desplazamiento, el valor de esta se muestra en un cuadro de edición y luego con un botón denominado *plot* se grafica.



```
function varargout = tut(varargin)
% TUT Application M-file for tut.fig
% FIG = TUT launch tut GUI.
% TUT('callback_name', ...) invoke the named callback.
```

```
if nargin == 0 % LAUNCH GUI
```

```
fig = openfig(mfilename,'reuse');
set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
```

Inicialización, aquí se pueden agregar otros valores de inicialización

```

% Generate a structure of handles to pass to callbacks, and store it.
handles = guihandles(fig);
guidata(fig, handles);

if nargout > 0
    varargout{1} = fig;
end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
    catch
        disp(lasterr);
    end
end

end

% -----

function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
var = handles.var;
noise = randn(1,350);
noise = noise * sqrt(var);
% -----
% instrucciones para imprimir en la gráfica del programa y que no se abra una
ventana nueva
h = gca;
axes(h);
% instrucciones para imprimir en la gráfica del programa y que no se abra una
ventana nueva
% esto es válido cuando solo tenemos una gráfica, por ahora está bien

```

subrutinabo tonplot

```
%-----  
plot(noise);  
title('RuidoGaussiano');  
grid on%  
guidata(gcbo,handles);  
%-----
```

```
function varargout = edit1_Callback(h, eventdata, handles, varargin)
```

```
%-----  
function varargout = slider1_Callback(h, eventdata, handles, varargin)  
var=get(handles.slider1,'value');  
var=var*2;  
set(handles.edit1,'string',num2str(var));  
handles.var=var;  
guidata(gcbo,handles);  
%-----
```

Subrutina slider