



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA MECÁNICA

# **REDUCCIÓN DEL ERROR EXPERIMENTAL EN LOS CAMPOS DE DEFORMACIÓN OBTENIDOS POR UN SISTEMA DE CORRELACIÓN DE IMÁGENES DIGITALES**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

**EDGAR LEONARDO GEERLING GAMBOA**

PROFESOR GUÍA:  
ELENA ATROSHCHENKO

MIEMBROS DE LA COMISION:  
ROGER BUSTAMANTE PLAZA  
VIVIANA MERUANE NARANJO

Este trabajo ha sido parcialmente financiado por proyecto FONDECYT de iniciación 11130259

SANTIAGO DE CHILE  
2015

**RESUMEN DE LA MEMORIA PARA OPTAR AL  
TÍTULO DE:**

Ingeniero Civil Mecánico

**POR:** Edgar Leonardo Geerling Gamboa

**FECHA:** 10/12/2015

**PROFESOR GUÍA:** Elena Atroshchenko

**REDUCCIÓN DEL ERROR EXPERIMENTAL EN LOS CAMPOS DE DEFORMACIÓN  
OBTENIDOS POR UN SISTEMA DE CORRELACIÓN DE IMÁGENES DIGITALES**

Este trabajo consiste en obtener el campo de deformación de una probeta de acero SAE 1020 sometida a flexión, a partir de mediciones del campo de desplazamiento a través de correlación de imágenes digitales con cámaras DIC (Digital Image Correlation). Este sistema DIC, es un método óptico no destructivo para medir el campo de desplazamientos y deformaciones en todo punto. Sin embargo, el proceso de captura digital de imágenes está sujeto a muchas fuentes de ruido y error. Esta situación, define el principal foco de estudio de reducir el efecto del ruido experimental y los errores numéricos en la obtención del campo de deformaciones, en donde los resultados obtenidos son comparados con la solución del problema de flexión, utilizando elementos finitos. De esta forma, el trabajo realizado es necesario como fase preliminar para el estudio de fenómenos más complejos, tales como, el estudio de propagación de grietas en materiales considerando el efecto microestructural, así como también en materiales frágiles, por mencionar algunos.

La primera parte del trabajo, contempla una revisión detallada de la bibliografía y estudio de la teoría, junto con la obtención de desplazamientos utilizando correlación de imágenes digitales. Para esto, se propusieron dos técnicas de preparación de las probetas para la correlación de imágenes digitales. La primera, por medio del salpicado de tinta con un cepillo y tinta en spray negra. La segunda propone el estampado del patrón de manchas con tinta gráfica por medio de un timbre fabricado por cortado laser. Esta última, implicó los mejores resultados en la obtención de desplazamientos. Posterior a la preparación de probetas, se definen los parámetros de operación de la máquina del ensayo, para luego obtener los desplazamientos de las muestras utilizando el software propio de las cámaras DIC. Los parámetros de correlación de mejores resultados fueron un sub set de referencia de 17x17 pixeles y distancia de paso de 5 pixeles, presentando un error promedio de dos órdenes de magnitud menor a la escala de los desplazamientos medidos.

La segunda parte, busca la reducción del ruido en los campos de deformación por medio de tres métodos diferentes. El primer método propuesto busca suavizar el campo de desplazamientos de forma previa a la obtención del campo de deformación por diferenciación numérica. Este método alcanza una reducción de un 88.5% del error promedio, con respecto a los campos de deformación obtenidos directamente sin suavizar. El segundo, utiliza el método Pointwise Least-Squares, por ajuste de mínimos cuadrados, alcanzando una reducción del 80,5%. Finalmente, el último método, busca la disminución del ruido inherente en el campo de deformación, definiendo sobre él una función de costos regularizada, cuyo mínimo implica el campo de mejor rendimiento. Este último método alcanza una reducción del 72,28% del error promedio.

Del trabajo realizado, se observa que el montaje experimental y la calibración de las cámaras tienen la mayor influencia en los errores de los desplazamientos medidos. En cuanto a los métodos de reducción del error en el campo de deformación, el más efectivo corresponde al suavizado de los desplazamientos mediante la aproximación de la transformada de coseno discreta.

# Tabla de Contenido

---

1	Introducción .....	1
1.1	Principios Básicos - Correlación de Imágenes Digitales (DIC).....	2
1.2	Motivación .....	4
1.3	Objetivos .....	4
1.3.1	Objetivo Principal .....	4
1.3.2	Objetivos Específicos.....	4
1.3.3	Alcances.....	5
2	Marco Teórico .....	6
2.1	Desplazamientos y Deformación Bidimensional .....	6
2.2	Errores Sistemáticos y Ruido .....	9
2.2.1	Errores de Correlación .....	9
2.2.2	Errores de Calibración y Montaje Experimental .....	10
2.3	Reducción del error .....	12
2.3.1	Atenuación de ruido en desplazamiento .....	12
2.3.2	Método de Obtención Deformaciones Pointwise Least-Squares (PLS) .....	12
2.3.3	Atenuación de ruido en deformaciones.....	14
3	Metodología .....	16
3.1	Insumos .....	18
3.2	Infraestructura .....	19
3.2.1	Equipos .....	19
3.2.2	Software .....	19
4	Montaje experimental.....	20
4.1	Preparación de Probetas .....	20
4.2	Ensayo de Flexión .....	21
4.3	Cámaras y Software DIC.....	22

5	Resultados y Análisis .....	24
5.1	Modelo Elementos Finitos .....	24
	Análisis .....	26
5.2	Medición de Desplazamientos .....	26
	Análisis .....	29
5.3	Método de Suavizado de Desplazamientos y Aplicación del Gradiente.....	30
	Análisis .....	35
5.4	Método PLS.....	36
	Análisis .....	39
5.5	Reducción del Ruido en el Campo de Deformación, Utilizando una Función de Costos Regularizada.....	39
	Análisis .....	42
5.6	Resumen de Resultados.....	42
6	Conclusiones .....	45
7	Bibliografía.....	48
	Anexos.....	
	Anexo A: Consideraciones del Montaje Experimental .....	
	Anexo B: Definición de malla de Elementos Finitos .....	
	Anexo C: Muestras No Incluidas En Resultados .....	
	C.1 Distribución de Desplazamientos Horizontales y Desviación Estándar Probeta tipo Cepillo	
	C.2 Distribución de Desplazamientos Horizontales y Desviación Estándar Probeta con estampado timbre .....	
	Anexo D: Códigos MATLAB .....	
	D.1 Función Importar_datos(), get_disp_x(),get_disp_y(),get_disp_z() .....	
	D.2 Método de suavizado de desplazamientos “smoothn.m” .....	
	D.3 Método PLS.....	
	D.4 Reducción del Ruido en el Campo de Deformación, a través de Minimización de Función de Costos Regularizada .....	
	D.5 Script para cargar datos en Matlab .....	
	D.6 Función ajuste del origen sistema de coordenadas minimizando desviación estandar para vectores de traslación $tx,ty$ y de rotación $alfa$ . .....	

D.7 Función de obtención de error promedio de desplazamientos .....	
D.8 Scripts para obtención de gráficos.....	

## Índice de Tablas

---

Tabla 5.1. Valores máximos y mínimos de los desplazamientos y deformaciones, obtenidos por el modelo de elementos finitos, para los casos de estudio.....	25
Tabla 5.2. Resumen del porcentaje de reducción de error, para los métodos de obtención de campos de deformación suavizados. ....	44

# Índice de Figuras

---

Figura 1.1 Parámetros de análisis software DIC.....	2
Figura 1.2. Seguimiento del desplazamiento de un subconjunto de píxeles, definidos en la imagen original en el tiempo $t_0$ y rastreados en la imagen sometida a deformación en el tiempo $t_2$ .....	2
Figura 1.3 Representación gráfica de las transformaciones que define una función de forma .....	3
Figura 2.1 Esquema de deformación teórica de un cuerpo sometido a fuerzas externas.....	7
Figura 2.2: Métodos de interpolación de imágenes, utilizados por los algoritmos de correlación de imágenes digitales, para obtención de un espacio continuo en escala de grises.....	9
Figura 2.3 Ejemplos de patrones de mancha con características de tipo: a) Repetitivos; b) No isotrópicos; c) Bajo contraste; d) Ideal. ....	10
Figura 2.4: Efecto de Aliasing producido por tamaño de manchas muy pequeño y baja resolución de imagen. ....	11
Figura 2.5: Expresión del efecto Aliasing, en la medición de desplazamientos en eje Z. ....	11
Figura 2.6. Ejemplo de cálculo utilizando el método PLS .....	13
Figura 3.1: Esquema representativo de los pasos que definen la metodología de trabajo. ....	16
Figura 4.1: Técnica de aplicación del patrón de manchas con cepillo y tinta en spray. ....	20
Figura 4.2: Método de Estampado del Patrón de Manchas con Timbre. En a) Elementos necesarios y (b) Resultado.....	21
Figura 4.3 Montaje del ensayo de flexión de tres puntos. ....	21
Figura 4.4 Vistas del montaje experimental. ....	22
Figura 4.5: Área seleccionada para el cálculo de desplazamientos .....	23
Figura 5.1: Mallado, carga y condiciones de borde consideradas en el modelo de elementos finitos. ....	24
Figura 5.2: Distribución espacial de desplazamientos horizontales $u_x$ del modelo en elementos finitos. ....	25
Figura 5.3: Distribución espacial de la deformación normal horizontal $\epsilon_x$ del modelo en elementos finitos. ....	25
Figura 5.4. Resultados de la medición DIC, para la probeta con patrón tipo “Cepillo”, sub set de referencia 23x23, y distancia de paso 5 píxeles.....	27
Figura 5.5: Error promedio y desviación estándar de probetas tipo “cepillo” en el cálculo de desplazamientos, en función del tamaño de sub set.....	27
Figura 5.6. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre”, sub set de referencia 17x17, y distancia de paso 5 píxeles.....	28
Figura 5.7. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre”, sub set de referencia 27x27, y distancia de paso 5 píxeles.....	28
Figura 5.8: Error promedio y desviación estándar para las probetas tipo “timbre”, para el cálculo de desplazamientos, en función del tamaño de sub set.....	29

Figura 5.9. Líneas de contorno de los desplazamientos horizontales $u_x$ , para distintos valores de suavizado, para la probeta con patrón tipo “Timbre”, sub set de referencia 27x27, y distancia de paso 5 pixeles.....	31
Figura 5.10. Resultados de la aplicación del gradiente direccional, sobre desplazamientos sin suavizar.....	32
Figura 5.11. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método smoothn.m y valor de suavizado s=1.....	32
Figura 5.12. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método "smoothn.m y valor de suavizado s=10.....	33
Figura 5.13. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método smoothn.m y valor de suavizado s=100.....	33
Figura 5.14. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método smoothn.m y valor de suavizado s=1000.....	34
Figura 5.15: Error promedio y desviación estándar del cálculo del campo de deformación, a partir de desplazamientos horizontales, para distintos valores de suavizado s del método “Smoothn.m” .....	34
Figura 5.16: Error promedio y desviación estándar del cálculo del campo de deformación, a partir de desplazamientos horizontales, para distintos valores de suavizado s del método “Smoothn.m”, considerando solo los valores de la probeta entre $x=-95$ y $x=95$ .....	36
Figura 5.17: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo de 5x5 datos.....	37
Figura 5.18: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo de 17x17 datos.....	37
Figura 5.19: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo 33x33 datos.....	38
Figura 5.20: Resumen de errores promedios y desviación estándar para cada campo de deformación obtenido a través del método PLS .....	38
Figura 5.21: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro $\alpha = 0.01$ .....	40
Figura 5.22: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro $\alpha = 0.1$ .....	40
Figura 5.23: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro $\alpha = 0.3$ .....	41
Figura 5.24: Resumen de errores promedios y desviación estándar para cada campo de deformación obtenido a través del método de reducción de error, utilizando una función de costos regularizada.....	41
Figura 5.25: Resumen de resultados del error promedio y desviación estándar para los tres métodos propuestos de reducción del ruido y errores en el campo de deformación.....	43

# 1 Introducción

La correlación de imágenes digitales (Digital Image Correlation - DIC), es un método óptico no destructivo para medir el campo completo de desplazamientos sobre la superficie de un objeto que se deforma. Este método ha sido ampliamente utilizado e investigado en el área de la mecánica experimental los últimos 6 años [1],[2],[3],[4], evolucionando y buscando nuevos métodos que propicien la obtención de resultados más precisos y confiables. El método de medición consiste en captar con cámaras de sensores CCD<sup>1</sup>, imágenes de una muestra sometida a algún tipo de ensayo (generalmente tracción, flexión o vibración). Esta muestra debe tener impresa sobre ella un patrón de manchas, el cual finalmente permitirá comparar las imágenes obtenidas luego del ensayo con la muestra original, para así obtener los desplazamientos para cada punto de interés, en cualquiera de los ejes de coordenadas x,y,z requeridos.

Este proceso de captura digital está sujeto a muchas fuentes de ruido y error, lo que finalmente se traduce en desviaciones en los resultados de campos de desplazamiento, y en consecuencia, en los campos de deformación. Debido a que el campo de deformación se obtiene a partir del gradiente de los desplazamientos, el ruido experimental se ve amplificado, lo que incide en que su valor como información útil sea baja. La medición de esfuerzos y deformaciones, representan información realmente útil en el área de mecánica de sólidos, lo que explica la necesidad actual de los investigadores de definir y aplicar métodos que permitan extraer esta información de manera confiable.

En este trabajo, se desarrollan dos técnicas para la impresión del patrón de manchas sobre las probetas, siendo estas sometidas a un ensayo de flexión y luego medidas por un sistema de correlación de imágenes, para posteriormente, analizar el error de los desplazamientos experimentales obtenidos con un modelo de elementos finitos. Además de lo anterior, se implementan tres métodos diferentes para reducir el error en el campo de deformación. El primero, supone el suavizado de los desplazamientos y posterior aplicación del gradiente sobre estos datos para la obtención del campo de deformación. El segundo, propone la obtención del campo de deformación por un método alternativo a la aplicación del gradiente, utilizando una aproximación por mínimos cuadrados. El último, plantea la eliminación del ruido directamente en el campo de deformación obtenido mediante el gradiente de desplazamientos no suavizados. El error promedio de cada método será obtenido al comparar los resultados con un modelo de elementos finitos de deformaciones previamente desarrollado.

De esta manera, el trabajo y observaciones realizadas en esta memoria, sirven como herramienta para futuros proyectos en los que se requiera la obtención de deformaciones de una muestra, tales como, validación de modelos teóricos o estudios sobre el comportamiento mecánico de diversos materiales.

---

<sup>1</sup> Por sus siglas en inglés Charged Coupled Device

## 1.1 Principios Básicos - Correlación de Imágenes Digitales (DIC)

El algoritmo de correlación de imágenes digitales (DIC), calcula los desplazamientos para todos los *Puntos de Interés*, definidos dentro de una *Región de Interés* particular de una *Imagen de Referencia* (ver Figura 1.1). Para realizar una medición utilizando un algoritmo DIC, el primer paso es tener una probeta adecuada a los requerimientos del algoritmo para el reconocimiento de desplazamientos de la muestra estudiada. Esto es, utilizar un patrón de manchas sobre la probeta, el cual permita la búsqueda y reconocimiento de un subconjunto<sup>2</sup> de píxeles en la imagen sometida a deformación, previamente definidos en la imagen original, como se grafica en la Figura 1.2

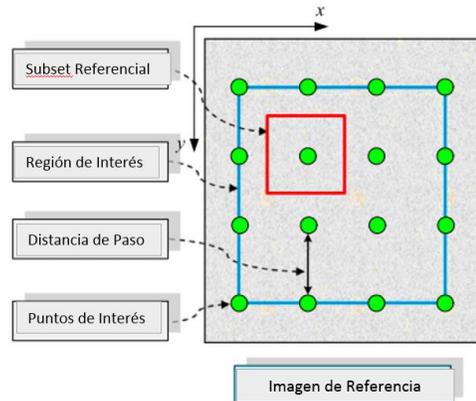


Figura 1.1 Parámetros de análisis software DIC  
Fuente: Imagen adaptada de [5]

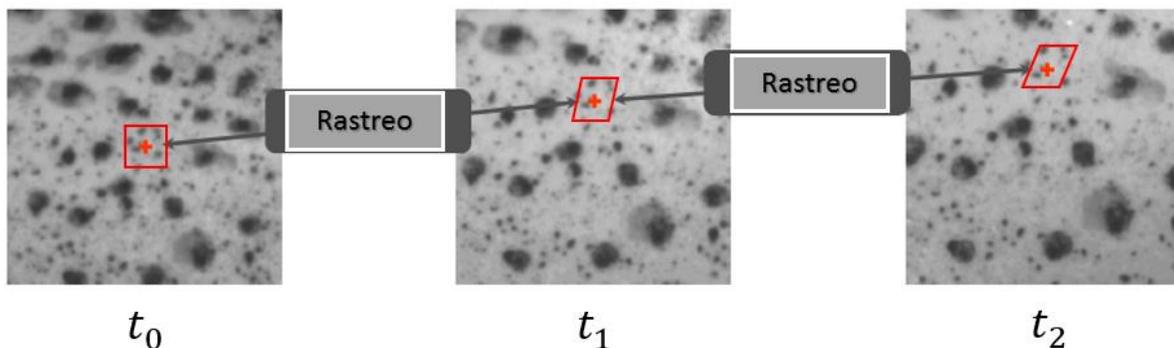


Figura 1.2. Seguimiento del desplazamiento de un subconjunto de píxeles, definidos en la imagen original en el tiempo  $t_0$  y rastreados en la imagen sometida a deformación en el tiempo  $t_2$ .  
Fuente: Imagen adaptada de [6].

<sup>2</sup> También son denominados sub sets.

Cada subconjunto de pixeles definidos en la imagen de referencia, tiene un valor característico asociado, el cual lo diferencia de otros subconjuntos. Para encontrar la nueva ubicación del subconjunto de la imagen de referencia, se debe definir este valor y algún criterio de similitud que se maximice al reconocerlo en la imagen objetivo, obteniendo así el desplazamiento. Por lo general, se utiliza el valor de la escala de grises para reconocimiento del subconjunto y como criterio se utilizan funciones de correlación. La ecuación 1.1 corresponde a una función de correlación cruzada de error promedio cero, utilizada por los algoritmos DIC para evaluar la similitud de dos sub sets:

$$C = \frac{\sum_{x=-N}^N \sum_{y=-N}^N [f(x,y) - \bar{f}(x,y)][g(x',y') - \bar{g}(x',y')]}{\sqrt{\sum_{x=-N}^N \sum_{y=-N}^N [f(x,y) - \bar{f}(x,y)]^2} \sqrt{\sum_{x=-N}^N \sum_{y=-N}^N [g(x',y') - \bar{g}(x',y')]^2}} \quad 1.1$$

En donde  $f(x, y)$ , corresponde al valor en escala de grises para el punto  $(x, y)$  de la imagen de referencia, y  $g(x', y')$  el valor en escala de grises para el punto  $(x', y')$  de la imagen objetivo. Así,  $\bar{f}(x, y) = (1/(2N + 1)^2) \sum_{x=-N}^N \sum_{y=-N}^N [f(x, y)]$  es la intensidad promedio del sub set de referencia y  $\bar{g}(x', y') = (1/(2N + 1)^2) \sum_{x=-N}^N \sum_{y=-N}^N [g(x', y')]$  es la intensidad promedio del sub set objetivo.  $N$  es el tamaño del sub set utilizado para el cálculo.

Por otra parte, la geometría del subconjunto de pixeles puede variar debido a la deformación de la muestra. Debido a esto, el algoritmo se vale de las funciones de forma (descritas en las ecuaciones 1.2 y 1.3) que describen cambios geométricos del subconjunto de pixeles original.

$$x^* = a_0 + a_1x + a_2y + a_3xy \quad 1.2$$

$$y^* = a_4 + a_5x + a_6y + a_7xy \quad 1.3$$

En donde,  $(x^*, y^*)$  representa las coordenadas en la imagen deformada del punto  $(x, y)$ , de la imagen original, calculado a partir de las constantes  $a_1, a_2, a_3, a_4, a_5, a_6$  y  $a_7$ . Dichas constantes representan las transformaciones posibles a partir de la combinación de movimientos de traslación, rotación, acortamiento, distorsión y deformación de corte, representado en la Figura 1.3.

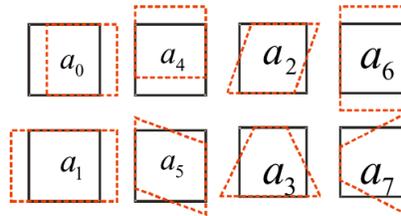


Figura 1.3 Representación gráfica de las transformaciones que define una función de forma  
Fuente: Imagen adaptada de [7]

De esta forma, comparando numéricamente las imágenes iniciales con las finales, a través defunciones de correlación, el algoritmo entrega los campos de desplazamiento de las imágenes de muestra con precisión sub-pixel.

## 1.2 Motivación

La principal dificultad para la obtención de los campos de deformación, radica en que cada paso, aproximación e interpolación utilizada por el algoritmo DIC conlleva en mayor o menor grado a un error sistemático sobre las mediciones. Así también, lo hacen las imperfecciones en la implementación del experimento, como lo son los errores por calibración de los equipos, de contaminación (e.g. polvo en los sensores), de cambios de iluminación, etc. [8]. Por lo tanto, el gradiente de los desplazamientos (deformaciones), es seriamente corrompido por estos errores, especialmente, en deformaciones pequeñas, lo que conlleva que su valor como información útil sea muy bajo.

Actualmente, en el Departamento de Ingeniería Mecánica de la Universidad de Chile, se han utilizado las cámaras DIC para medir fenómenos de vibraciones, sin embargo, aún no se han realizado estudios sobre esto, ni tampoco existe una metodología clara para la obtención de los campos de deformación. Esto, explica la necesidad actual del Departamento de definir y aplicar algún método para poder extraer esta información de manera confiable. Por lo que, este trabajo de título, es el primer paso para el Departamento en la medición de deformaciones, con el fin de facilitar estudios relacionados posteriores.

## 1.3 Objetivos

### 1.3.1 Objetivo Principal

El objetivo de este trabajo es determinar y reducir el efecto del ruido experimental y los errores numéricos para tres métodos diferentes en la obtención del campo de deformaciones. Esto, será estimado mediante un sistema de correlación digital de imágenes.

### 1.3.2 Objetivos Específicos

Los objetivos específicos son:

- Definir una metodología para imprimir un patrón de manchas adecuado para mediciones con el método. Posteriormente, realizar ensayos de flexión de tres puntos con la máquina de ensayos Zwick Roell Z-100 sobre probetas de acero SAE 1020 de 38x38x400 mm, con el patrón de manchas impreso en su superficie. Los desplazamientos deben ser obtenidos por el software propio del sistema DIC, para luego importar los datos obtenidos al entorno MATLAB para así ser procesados.
- Reducir los errores en los campos de deformación, a partir de la diferenciación numérica de los campos de desplazamientos suavizados, utilizando la transformada coseno discreta y la aproximación de mínimos cuadrados penalizados.
- Reducir el error de los campos de deformación, a partir del método alternativo de obtención de los campos de deformación, por medio de la aproximación de mínimos cuadrados locales (método Pointwise Least-Squares).

- Reducir el error inherente del campo de deformaciones obtenido por diferenciación numérica de los desplazamientos no suavizados, minimizando una función de costos regularizada sobre el campo de deformaciones.
- Determinar la medida del error de los desplazamientos medidos y los campos de deformación obtenidos, comparando los resultados con un modelo del experimento en elementos finitos realizado en el software ANSYS.

### **1.3.3 Alcances**

Para este trabajo, solo se analizará la efectividad de los métodos para los desplazamientos y deformaciones normales en el eje  $x$ , puesto que los errores de los desplazamientos y deformaciones normales en el eje  $y$  son del mismo orden [4], por lo que se desprenden conclusiones similares con respecto a los métodos estudiados.

Además, no se considerará el estudio de la influencia en la elección y definición de los algoritmos y métodos numéricos asociados a la correlación de imágenes, y se utilizarán los métodos preestablecidos por el software Istra 4D para la obtención de los desplazamientos. Por lo tanto, la reducción de los errores sistemáticos, se limitará a la relación de estos con el tamaño del sub set de referencia, a la variable dependiente que define cada uno de los métodos para la obtención de deformación y a un montaje experimental adecuado.

## 2 Marco Teórico

### 2.1 Desplazamientos y Deformación Bidimensional

En cualquier cuerpo sometido a algún esfuerzo que propicie deformación del mismo (Figura 2.1 izquierda), se producirán desplazamientos de los puntos  $P(x,y)$  originales que conforman dicho cuerpo. En este cuerpo, el campo de desplazamiento puede ser definido como el campo vectorial que para cada punto  $(x,y)$  indica cuanto se mueve dicho punto al deformarse a su nueva ubicación en  $P(x^*,y^*)$ . Formalmente, esto se define según las ecuaciones 2.1 y 2.2,

$$x^* = x + u(x, y) \quad 2.1$$

$$y^* = y + v(x, y) \quad 2.2$$

donde  $x$  e  $y$ , corresponden a las coordenadas de los puntos originales;  $x^*$  e  $y^*$  corresponden a las coordenadas de los puntos originales;  $u(x, y)$  y  $v(x, y)$ , definen los desplazamientos horizontales y verticales respectivamente. Realizando una expansión de Taylor en torno al punto  $P(x_o, y_o)$ , las ecuaciones 2.1 y 2.2 se pueden reescribir como:

$$x^* = x_o + u_o + \frac{\partial u}{\partial x} \Delta x + \frac{\partial u}{\partial y} \Delta y + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Delta x^2 + \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \Delta y^2 + \frac{1}{2} \frac{\partial^2 u}{\partial x \partial y} \Delta x \Delta y \quad 2.3$$

$$y^* = y_o + v_o + \frac{\partial v}{\partial x} \Delta x + \frac{\partial v}{\partial y} \Delta y + \frac{1}{2} \frac{\partial^2 v}{\partial x^2} \Delta x^2 + \frac{1}{2} \frac{\partial^2 v}{\partial y^2} \Delta y^2 + \frac{1}{2} \frac{\partial^2 v}{\partial x \partial y} \Delta x \Delta y \quad 2.4$$

donde  $x_o$  e  $y_o$  son las coordenadas del punto inicial que se evalúa en la serie de Taylor;  $\Delta x$  corresponde a la diferencia entre  $x - x_o$  y  $\Delta y$  a la diferencia  $y - y_o$ ;  $u_o$  y  $v_o$  son los componentes que definen la traslación del punto  $P(x_o, y_o)$ . Como en general se trabaja con deformaciones pequeñas, los términos de mayor orden se desprecian, quedando así la siguiente expresión que define los desplazamientos horizontales y verticales respectivamente:

$$x^* - x_o = u_o + \frac{\partial u}{\partial x} \Delta x + \frac{\partial u}{\partial y} \Delta y \quad 2.5$$

$$y^* - y_o = v_o + \frac{\partial v}{\partial x} \Delta x + \frac{\partial v}{\partial y} \Delta y \quad 2.6$$

Para el caso bidimensional esquematizado en la Figura 2.1, se puede analizar un elemento infinitesimal de forma rectangular, el cual sufrirá deformaciones debido a las fuerzas externas y a restricciones de desplazamiento a las que está sometido. De esta forma, el elemento  $ABCD$  ilustra el cuadrado diferencial antes de ser sometido a esfuerzos y el elemento romboideo  $A'B'C'D'$  de líneas segmentadas lo describe luego de ser sometido a esfuerzos.

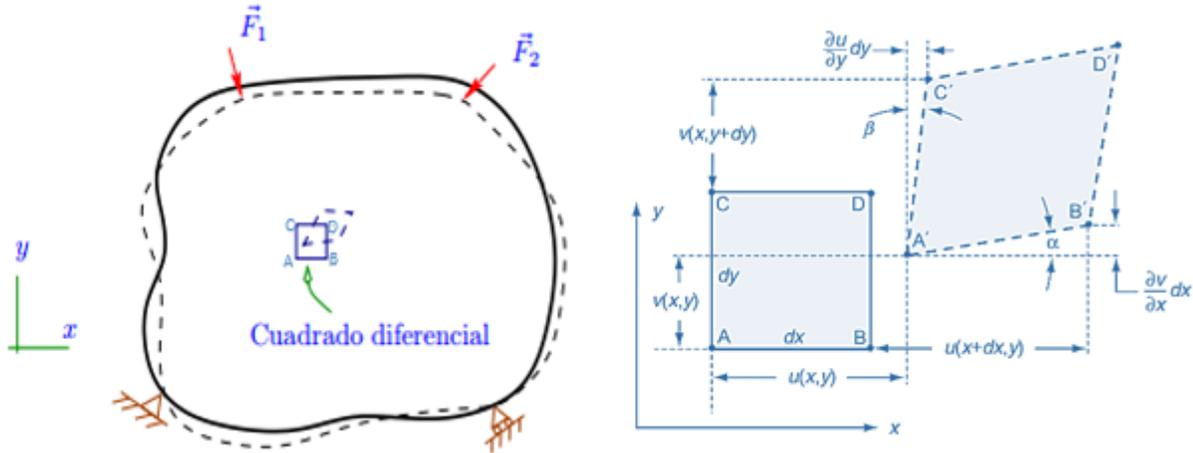


Figura 2.1 Esquema de deformación teórica de un cuerpo sometido a fuerzas externas.  
Fuente: - Imagen Adaptada de [9] y [10]

El punto de referencia  $A$  se sitúa en las coordenadas  $(x, y)$ , así como los desplazamientos horizontales  $u(x, y)$  y verticales  $v(x, y)$ . Por otra parte, para el punto  $B$  ubicado en  $(x + dx, y)$ , los desplazamientos horizontales corresponden a  $u(x + dx, y)$  y los verticales a  $v(x + dx, y)$ . De manera análoga, se pueden definir los desplazamientos de los otros puntos de referencia. De acuerdo a la teoría de pequeños desplazamientos, se puede aproximar  $u(x + dx, y) \approx u(x, y) + \frac{\partial u}{\partial x} dx$  y de manera análoga se puede hacer para el resto de los puntos.

En la definición de esfuerzos pequeños, la deformación normal horizontal es el cambio en el largo del elemento dividido por su largo inicial, lo que se puede traducir en la expresión 2.7, la cual es análoga para las otras dimensiones. Esta se define como:

$$\varepsilon_{xx} = \frac{A'B' - AB}{AB} \quad 2.7$$

donde  $A'B'$  y  $AB$  corresponden a los elementos de línea representados en la Figura 2.1. Haciendo un análisis geométrico y siendo consecuente con la aproximación de pequeñas deformaciones (despreciando términos de mayor orden), se tiene la siguiente expresión:

$$\begin{aligned} A'B' &= \sqrt{\left(dx + \frac{\partial u}{\partial x} dx\right)^2 + \left(\frac{\partial v}{\partial x} dx\right)^2} = \sqrt{1 + 2\frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2} dx \\ &\approx \left(1 + \frac{\partial u}{\partial x}\right) dx \end{aligned} \quad 2.8$$

por lo que, como  $AB$  corresponde a  $dx$  (Figura 2.1), la expresión 2.7 se puede reescribir como:

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} \quad 2.9$$

donde  $\varepsilon_{xx}$  corresponde a las deformaciones normales horizontales, y de forma análoga se tiene que la deformación normal vertical queda definida como:

$$\varepsilon_{yy} = \frac{\partial v}{\partial y} \quad 2.10$$

donde  $\varepsilon_{yy}$  corresponde a las deformaciones normales verticales, por lo que la derivada parcial viene a ser el gradiente del desplazamiento, es decir, la representación de cómo varían los desplazamientos a través del material.

Por otra parte, las deformaciones de corte están relacionadas con el cambio de ángulos que sufre el elemento diferencial. Si de nuevo se analiza la Figura 2.1 en la derecha, se puede observar que los ángulos  $\alpha$  y  $\beta$  describen la deformación de corte, por lo que su definición queda acotada por el promedio de ambos como:

$$\varepsilon_{xy} = \frac{\alpha + \beta}{2} \quad 2.11$$

Bajo el supuesto de que las deformaciones son pequeñas, también los ángulos lo son, por lo que  $\alpha \approx \tan(\alpha)$  y  $\beta \approx \tan(\beta)$ , así también los términos derivativos  $\frac{\partial u}{\partial x} \ll 1$ . Considerando lo anterior y haciendo un análisis geométrico se tiene que:

$$\tan(\alpha) = \frac{\frac{\partial v}{\partial x} dx}{dx + \frac{\partial u}{\partial x} dx} = \frac{\frac{\partial v}{\partial x}}{1 + \frac{\partial u}{\partial x}} \approx \frac{\partial v}{\partial x} \quad 2.12$$

de manera análoga se puede demostrar:

$$\tan(\beta) \approx \frac{\partial u}{\partial y} \quad 2.13$$

donde finalmente:

$$\varepsilon_{xy} = \frac{1}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad 2.14$$

por lo que las deformaciones de corte quedan definidas como  $\varepsilon_{xy}$ .

## 2.2 Errores Sistemáticos y Ruido

Como se mencionó anteriormente, los datos de las coordenadas y desplazamientos de la superficie de un objeto, obtenidos utilizando el sistema DIC, están sujetos a cierta medida de incertidumbre en mayor magnitud en el campo de deformaciones. Estos errores pueden ser divididos en dos categorías, en primer lugar, están los errores de correlación, los cuales están asociados a los algoritmos y parámetros que definen el algoritmo DIC. En segundo lugar, están los errores del montaje experimental, que corresponden a aquellos originados por un montaje experimental no adecuado, entre los cuales se encuentran:

### 2.2.1 Errores de Correlación

#### Errores Asociados al Método de Interpolación de Imágenes Correlacionadas:

Para identificar los desplazamientos en escalas más detalladas que la que definen los píxeles que describen una imagen discreta, el algoritmo DIC precisa interpolar las imágenes a funciones continuas que representen lo que ocurre entre píxeles (escala sub-píxel). Los métodos de interpolación al ser una aproximación de la realidad, introducen la primera medida de error en el algoritmo DIC. La Figura 2.2 ilustra cómo se puede obtener una escala más detallada de una imagen a través de dos métodos de interpolación diferentes sobre una imagen en resolución de píxeles, en donde el eje  $z$  de los gráficos, representa los valores en escala de grises para los puntos definidos por las coordenadas  $x, y$  de las imágenes.

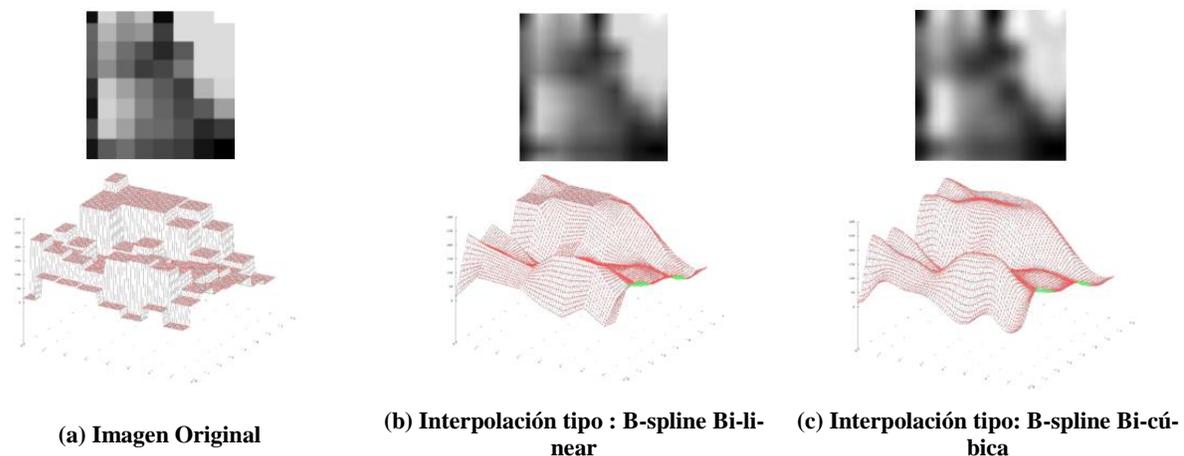


Figura 2.2: Métodos de interpolación de imágenes, utilizados por los algoritmos de correlación de imágenes digitales, para obtención de un espacio continuo en escala de grises.

Fuente: Imagen adaptada de [6]

#### Errores asociados a la elección de función de correlación y función de forma

Por una parte, la elección de la función de correlación que identifica el rastreo de los sub sets de referencia (como la descrita en la ecuación 1.1, en la sección 1.1), tiene influencia en el error de los desplazamientos medidos [4]. Por otra parte, la elección de la función de forma, implica una medida de error con respecto a la precisión con que describe el sub set de referencia utilizado para

la correlación de imágenes. Esto, debido a que para aumentar la precisión de la correlación se pueden incluir términos cuadráticos a las ecuaciones 1.2 y 1.3 (descritas en la sección 1.1) a costa de mayor gasto computacional [4].

## 2.2.2 Errores de Calibración y Montaje Experimental

### Errores de calibración

El procedimiento de calibración corresponde a la determinación de los parámetros de imagen de cada una de las cámaras (parámetros intrínsecos), así como las posiciones externas y orientaciones de estas con respecto a un sistema de coordenadas global. Se requiere la calibración del sistema para la transformación de las posiciones de los puntos de la superficie de una muestra, representados en las imágenes obtenidas por las dos cámaras, a las coordenadas 3D correspondientes de ese punto. Los errores de calibración son una importante fuente potencial de errores sistemáticos que limitan la resolución de los resultados [7], por lo que una calibración no adecuada puede tener como consecuencia la distorsión de las imagen adquiridas [8], [11]. El procedimiento de calibración en el software Istra 4D es parte del módulo de adquisición, para el cual el Anexo A describe detalles y consideraciones de este proceso.

### Errores asociados al patrón de manchas

Para lograr una correlación óptima con el fin de minimizar los errores sistemáticos, el patrón de manchas debe ser: no repetitivo, isotrópico<sup>3</sup> y de alto contraste [12]. La Figura 2.3 representa las características menos adecuadas de un patrón de manchas para los casos a), b) y c), mientras que el caso d) ilustra el caso ideal.

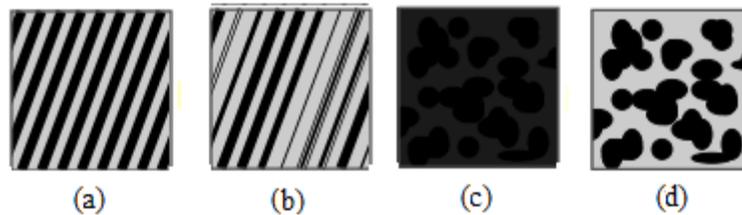


Figura 2.3 Ejemplos de patrones de mancha con características de tipo: a) Repetitivos; b) No isotrópicos; c) Bajo contraste; d) Ideal.

Fuente: Imagen adaptada de [6]

De lo anterior se deduce que, los patrones de manchas no deben ser ni muy grandes ni muy pequeño. Si las manchas del patrón son muy grandes, puede ocurrir que algunos sub sets encierren una región completamente negra o completamente blanca. Es por esto que lo anterior, no permite hacer un rastreo adecuado del sub set, ya que se tendrían coincidencias de esas regiones en muchas otras partes de la imagen. Se puede compensar este efecto con la definición un sub set más amplio y así asegurar mayor confiabilidad en los resultados, pero este, no será capaz de describir deformaciones heterogéneas grandes con precisión[4]. Por el contrario, si las manchas del patrón son más pequeñas, la correlación de los sub sets de referencia es más precisa, pero su resultado estará

<sup>3</sup> Propiedad que indica que la distribución de manchas en la muestra, no depende de la dirección en que es examinada.

dominado por la granularidad<sup>4</sup> del patrón de manchas y esto puede llevar a errores sistemáticos [13]. Por otra parte, si el tamaño de manchas es muy pequeño, la resolución de la cámara puede no ser suficiente para representar la muestra con precisión, ocurriendo el efecto de Aliasing<sup>5</sup>. Este efecto significa que, a medida que se hace la correlación, el sub set de referencia contendrá alteraciones oscilantes, producto de las pequeñas manchas y la incapacidad de las cámaras para captarlas en detalle, debido a una baja resolución de imagen (ver Figura 2.4). Este efecto, producirá una desviación del valor en la escala de grises para la correlación, por lo que el rastreo de los sub set no será preciso. El Aliasing, generalmente se expresa en los resultados como patrones del tipo *moiré* [12], como se muestra en la Figura 2.5.

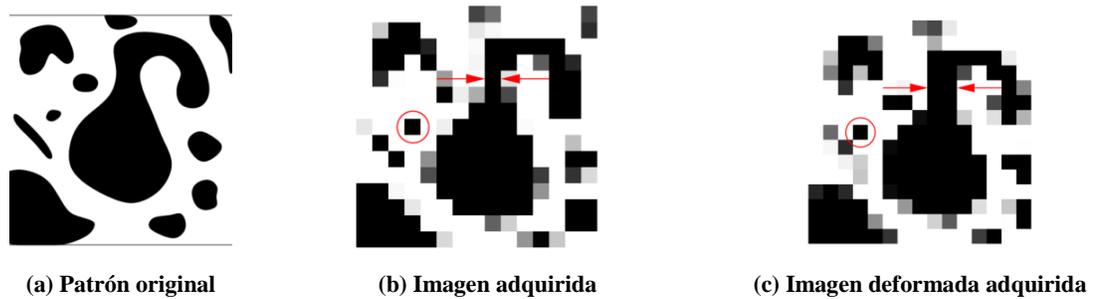


Figura 2.4: Efecto de Aliasing producido por tamaño de manchas muy pequeño y baja resolución de imagen. Fuente: Imagen adaptada de [6]

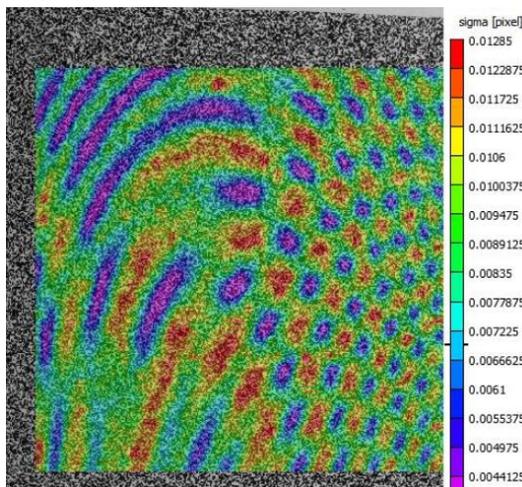


Figura 2.5: Expresión del efecto Aliasing, en la medición de desplazamientos en eje Z. Fuente: Imagen adaptada de [8]

El tamaño óptimo de sub set no está claramente establecido, pues depende de la granularidad del patrón de manchas y del grado de heterogeneidad del mismo [4], pero en general, se sugiere que las manchas deben ser al menos entre 3 o 4 píxeles de tamaño, con el fin de evitar estos problemas.

<sup>4</sup> Término asociado al nivel de detalle de un modelo. Cuanto mayor es la granularidad de un modelo, más profundo es el nivel de detalle. A su vez, este concepto se utiliza generalmente para caracterizar la escala o nivel de detalle en un conjunto de datos.

<sup>5</sup> Fenómeno que ocurre cuando una señal (en este caso, una imagen) contiene componentes de altas frecuencias, las cuales no pueden ser representadas por la frecuencia de muestreo, (en este caso, la resolución en píxeles de la imagen).

**Errores varios:** Otros factores, que inciden en desviaciones de los resultados obtenidos, son los efectos de contaminación de las imágenes (e.g. polvo en el sensor de adquisición), pérdida del contraste por degradación de la pintura, cambios en la iluminación, por mencionar algunos.

Los efectos de los errores sistemáticos en la medición de los desplazamientos, son inevitables [3], sin embargo, pueden ser disminuidos en gran medida, si se realiza un montaje experimental adecuado, en el cual pequeños cambios pueden significar una gran contribución a la reducción del error [11].

## 2.3 Reducción del error

### 2.3.1 Atenuación de ruido en desplazamiento

Como ni el ruido ni los errores sistemáticos pueden ser completamente eliminados, los autores que investigan el área concuerdan en la necesidad de tratar los datos obtenidos [1] [2], debido a que la diferenciación numérica directa (para la obtención de las deformaciones) supone la amplificación exagerada de los ruidos y errores en los desplazamientos. Para resolver la problemática anterior, por lo general, se filtran los datos de los desplazamientos con métodos de suavizado y posteriormente se obtienen los campos de deformación, ya sea por diferenciación numérica o por aproximación de mínimos cuadrados. Sin embargo, estos métodos generalmente suponen un compromiso entre el suavizado de los datos y la precisión de los valores encontrados.

En la actualidad, se han estudiado numerosos métodos para obtener el campo de deformaciones a partir de la diferenciación numérica sobre campos de desplazamientos suavizados. Estos métodos de suavizado se pueden clasificar como métodos de elementos finitos [5], [13], [14] y como métodos de ajuste a funciones continuas (locales y globales) [3], [2], [15]. Los métodos de aproximación de los desplazamientos a funciones globales continuas, se sugieren cuando el foco de las mediciones apunta a la deformación completa de la imagen. Por su parte, los métodos de aproximación a funciones locales continuas, se utilizan en general, para los estudios en donde el objeto presenta regiones localmente deformadas, las cuales precisan de mayor atención [3].

Este trabajo, considera el suavizado de los datos mediante el algoritmo basado en la aproximación de los desplazamientos sobre mínimos cuadrados penalizados, utilizando la transformada coseno discreta [16]. Esto permite un suavizado robusto para datos equiespaciados, en donde la intensidad del suavizado depende del parámetro de suavizado ' $\alpha$ '.

### 2.3.2 Método de Obtención Deformaciones Pointwise Least-Squares (PLS)

Además de la obtención de las deformaciones a través de una simple diferenciación numérica, es posible obtener una solución de las deformaciones por medio del método PLS, de mínimos cuadrados locales [2]. El cálculo de este método, requiere la elección de una ventana cuadrada de cálculo de deformaciones, que contiene  $(2m + 1) \times (2m + 1)$  puntos discretos. La Figura 2.6 ejemplifica la definición de la ventana de cálculo:

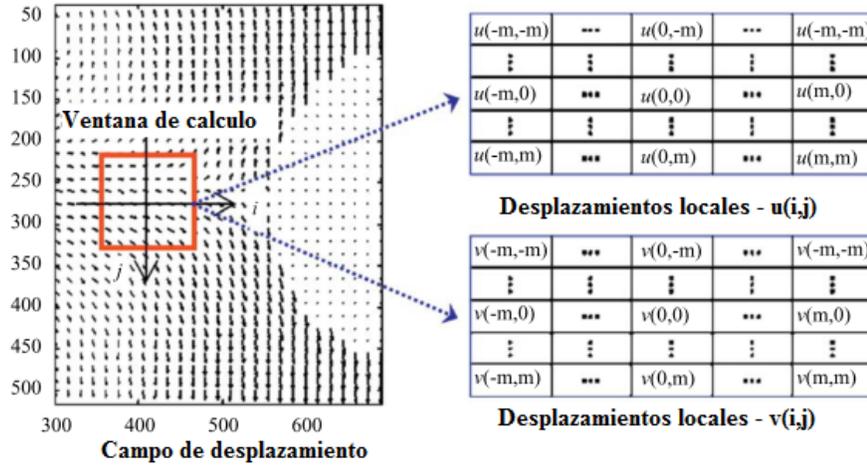


Figura 2.6. Ejemplo de cálculo utilizando el método PLS  
Fuente: Figura adaptada de [2]

Si la ventana es comparativamente pequeña con respecto a los desplazamientos obtenidos, éstos últimos se pueden aproximar a un comportamiento lineal, definido en las ecuaciones 2.15 y 2.16.

$$u(i, j) = a_0 + a_1 x + a_2 y \quad 2.15$$

$$v(i, j) = b_0 + b_1 x + b_2 y \quad 2.16$$

En donde,  $i, j$  son las coordenadas locales de la ventana de cálculo que toman los valores de  $-m$  a  $m$ ,  $u(i, j)$  y  $v(i, j)$  los valores de los desplazamientos horizontales y verticales en  $(i, j)$  obtenidos mediante DIC y  $a_{i=1,2,3}$  y  $b_{i=1,2,3}$  son los coeficientes del polinomio a determinar. A modo de simplificar el trabajo en MATLAB, las ecuaciones 2.15 y 2.16, pueden ser reescritas de manera matricial, como se muestra en la expresión 2.17.

$$\begin{bmatrix} 1 & -m & -m \\ 1 & -m+1 & -m \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & m-1 & m \\ 1 & m & m \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} u(-m, -m) \\ u(-m+1, -m) \\ \vdots \\ u(0, 0) \\ \vdots \\ u(m-1, m) \\ u(m, m) \end{pmatrix} \quad 2.17$$

Comparando las ecuaciones 2.15 y 2.16 con las ecuaciones 2.5 y 2.6 se tiene que, los coeficientes  $a_{i=1,2,3}$  y  $b_{i=1,2,3}$  representan las deformaciones normales horizontales y verticales, así como, las de corte (ecuaciones 2.9, 2.10 y 2.14 respectivamente) según las siguientes expresiones:

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} = a_1 \quad 2.18$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial y} = b_1 \quad 2.19$$

$$\varepsilon_{xy} = \frac{1}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = \frac{1}{2} (a_2 + b_2) \quad 2.20$$

Así, los valores de los coeficientes pueden ser encontrados utilizando aproximación por mínimos cuadrados. Esto supone una primera eliminación del ruido contenido en los desplazamientos, ya que la dispersión de los datos disminuye al ajustar los datos localmente. Sin embargo, el comportamiento lineal de la aproximación de los desplazamientos, supone por tanto que, la deformación encontrada será constante para cada ventana de cálculo. Entonces, la homogeneidad de la solución así como la precisión de los datos, queda determinada por el tamaño de cálculo de esta ventana.

Para cálculos en donde la deformación se prevé ser homogénea, el autor recomienda ventanas de cálculo grandes, sin embargo, para comportamientos no homogéneos se sugiere la elección cuidadosa del tamaño de la ventana. Esto, debido a que es necesario tener en cuenta el compromiso de la solución, pues si el tamaño de la ventana es muy pequeño, los ruidos inherentes al desplazamiento no son suficientemente suavizados. Por otra parte, si la ventana es muy grande, el supuesto de aproximación lineal puede hacerse no razonable. En un principio se sugiere un tamaño de entre 11 x 11 y 21 x 21 puntos.

### 2.3.3 Atenuación de ruido en deformaciones

Por otra parte, y a diferencia de los métodos anteriores, en los cuales se reduce el ruido en los desplazamientos, se pueden aplicar otros métodos en los cuales la eliminación del ruido se realiza en el campo de deformaciones de manera iterativa [1]. Éstos, han tenido mucha aceptación debido a su simpleza, alta precisión y efectividad, por lo que su uso es propuesto en este estudio.

Para esta estimación, se asume que las deformaciones obtenidas directamente del campo de desplazamiento denotadas como  $\varepsilon_r^n$ , son una versión ruidosa de las deformaciones genuinas  $\varepsilon_r$ . Ésto, se expresa en la ecuación 2.21.

$$\varepsilon_r^n = \varepsilon_r + n_r \quad 2.21$$

En donde  $n_r$  es la componente del error representada como una distribución gaussiana con una cierta distribución de probabilidad.

La filosofía de este método, consiste en encontrar el campo de deformaciones tal que minimice una función apropiada de energía, según un criterio de convergencia sobre el error de los cálculos iterativos. Esta función es definida como una función de costo regularizada, con campos aleatorios de Markov, en donde su representación matemática define que la influencia en la deformación sobre el punto considerado, es solo relevante para los cuatro puntos vecinos discretos [1]. Esta función compuesta (definida en la ecuación 2.22), está por dos componentes, en la cual la primera penaliza los residuos de la función (similitud entre el campo ruidoso original y el filtrado) y la segunda penaliza discontinuidades (asociada a la suavidad del campo filtrado):

$$U(\varepsilon) = \lambda \sum_{r \in L} (\varepsilon_r - \varepsilon_r^n)^2 + (1 - \lambda) \sum_{r \in L_s} \sum_{\varepsilon \in N_r} (\varepsilon_r - \varepsilon_s)^2 \quad 2.22$$

En donde L denota el ensamblaje de todos los puntos validos de cálculo; s representa los cuatro puntos vecinos de cálculo (i.e. izquierdo, derecho, superior e inferior) en torno al punto de cálculo actual;  $\lambda$  es el parámetro de suavizado que toma valores entre 0 y 1, el cual define el compromiso entre suavizado absoluto y similitud completa entre el campo original y el filtrado.

Ahora bien, la idea es encontrar el campo de deformaciones de mejores características, i.e. minimizar  $U(\varepsilon)$  utilizando MATLAB. Para esto, es necesario expresar la ecuación 2.22 de manera discreta (ecuación 2.23), así como su gradiente en la ecuación 2.24. Lo anterior, se explica pues se quiere encontrar el campo de deformaciones  $\varepsilon_{ij}$  de mejor confiabilidad que minimice  $U(\varepsilon_{ij})$ , que ocurre cuando  $\frac{\partial U(\varepsilon_{ij})}{\partial \varepsilon_{ij}} = 0$ .

$$U(\varepsilon_{ij}) = \sum_{(i,j) \in L} \left\{ \lambda (\varepsilon_{ij} - \varepsilon_{ij}^n)^2 + (1 - \lambda) \left[ (\varepsilon_{ij} - \varepsilon_{(i-\Delta L)j})^2 + (\varepsilon_{ij} - \varepsilon_{(i+\Delta L)j})^2 + (\varepsilon_{ij} - \varepsilon_{i(j+\Delta L)})^2 + (\varepsilon_{ij} - \varepsilon_{i(j-\Delta L)})^2 \right] \right\} \quad 2.23$$

$$\frac{\partial U(\varepsilon_{ij})}{\partial \varepsilon_{ij}} = 2\lambda(\varepsilon_{ij} - \varepsilon_{ij}^n) + 2(1 - \lambda)(4\varepsilon_{ij} - \varepsilon_{(i-\Delta L)j} - \varepsilon_{(i+\Delta L)j} - \varepsilon_{i(j+\Delta L)} - \varepsilon_{i(j-\Delta L)}) \quad 2.24$$

La raíz de la ecuación 2.24 se puede resolver utilizando el método iterativo de Newton Rhapson, lo que se traduce en la formulación iterativa descrita a continuación:

$$\varepsilon_{ij}^{k+1} = \varepsilon_{ij}^k - \frac{\nabla U(\varepsilon_{ij}^k)}{\nabla \nabla(\varepsilon_{ij}^k)} \quad 2.25$$

En donde  $\nabla U(\varepsilon_{ij}^k)$  es definido en la ecuación 2.24;  $\nabla \nabla(\varepsilon_{ij}^k)$  es la derivada de segundo orden de la función de costos cuadrática regularizada, definida como  $\nabla \nabla(\varepsilon_{ij}^k) = 8 - 6\lambda$ ;  $\varepsilon_{ij}^k$ , y  $\varepsilon_{ij}^{k+1}$  son las soluciones  $k$  y  $(k+1)$ -ésimas de la iteración. La primera iteración, se puede iniciar con los valores de las deformaciones ruidosas inicialmente obtenidas ( $\varepsilon_{ij}^n$ ). Además, se debe fijar un criterio que asegure la convergencia de la iteración. La referencia [1], sugiere  $\frac{|\varepsilon_{ij}^{k+1} - \varepsilon_{ij}^k|}{|\varepsilon_{ij}^{k+1}|} = \leq 0.1\%$  o fijar un máximo de 1000 iteraciones.

### 3 Metodología

Los pasos a realizar en el presente trabajo se muestran en el esquema de la Figura 3.1, sus respectivas explicaciones se detallan a continuación:

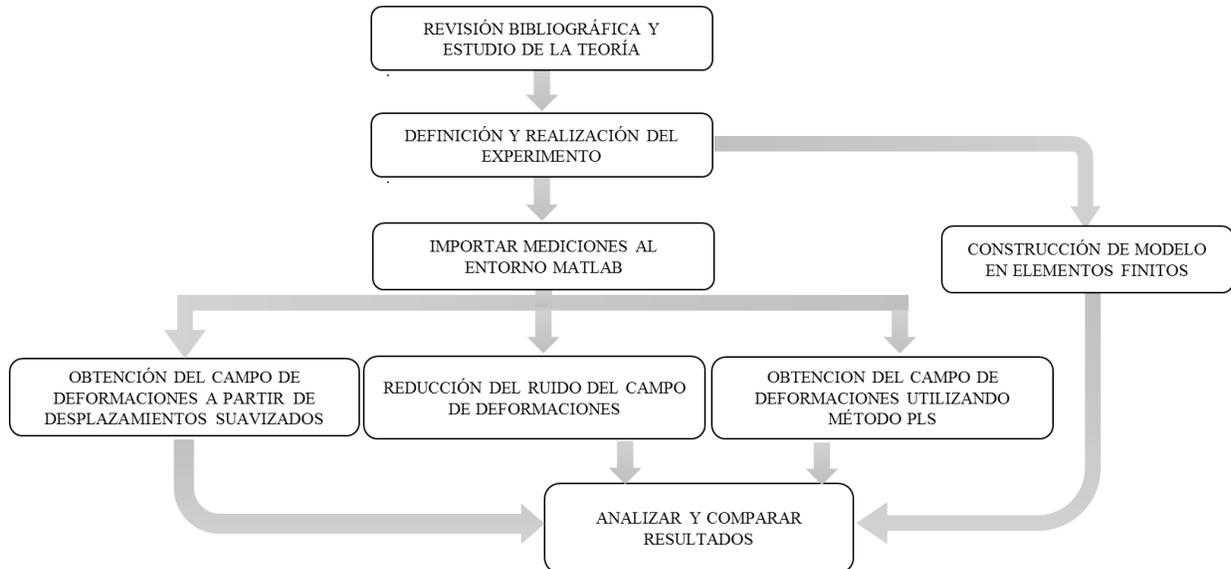


Figura 3.1: Esquema representativo de los pasos que definen la metodología de trabajo.  
Fuente: Elaboración propia

- **Revisión bibliográfica y estudio de la teoría:** Estudio detallado de los antecedentes y marco teórico necesarios para el trabajo.
- **Definición y realización del experimento:** Preparar las probetas con el patrón de manchas negras por medio de dos técnicas diferentes: la del salpicado de tinta negra en spray utilizando un cepillo, por un lado; y el estampado de patrón de manchas con un timbre y tinta gráfica negra, por otro. Posteriormente, se deben calibrar las cámaras utilizando el software ISTR4 4D y luego, en el mismo programa, definir los parámetros de correlación para la obtención de los desplazamientos y sus respectivas coordenadas en formato '.hdf5'. Finalmente, se deben definir parámetros de operación de la máquina del ensayo de flexión y realizar las mediciones de la experiencia.
- **Importar las mediciones al entorno MATLAB:** Implementar un script en el lenguaje de programación de MATLAB para obtener la información de las mediciones correspondientes a: Coordenadas  $x, y, z$  de los puntos de interés medidos y el valor del desplazamiento horizontal  $u_x$  asociado a estos.
- **Obtención del campo de deformaciones utilizando método PLS:** Implementar en MATLAB el método Pointwise Least Squares, y luego obtener los campos de deformación para un rango de valores del tamaño de la ventana de cálculo ' $M$ ', valor que define la aproximación/resultado del método.

- **Obtención del campo de deformaciones a partir de desplazamientos suavizados:** Implementar el método de suavizado '*smoothn.m*', basado en la aproximación de los desplazamientos sobre mínimos cuadrados penalizados, utilizando la transformada de coseno discreta. Aplicar este algoritmo para un rango de valores del parámetro '*s*', propio del método que define la cantidad del suavizado sobre los desplazamientos. Posteriormente, obtener el campo de deformación para cada valor del parámetro '*s*' con el método de diferenciación numérica previamente implementado.
- **Reducción del error del campo de deformaciones:** Implementar en MATLAB un método que reduzca el error en el campo de deformaciones obtenido por diferenciación numérica de los desplazamientos no suavizados. Para esto, se debe definir una función de costos regularizada (parametrizada por el valor '*α*') sobre el campo de deformación, cuyo mínimo se obtiene para el campo de deformación de menor error, a través del método Newton-Rhapson. Posteriormente, calcular los campos de deformación para un rango de valores de '*α*'.
- **Construcción de modelo en elementos finitos:** Encontrar la solución al problema con el método de elementos finitos utilizando el software ANSYS. Posteriormente, exportar los datos al entorno MATLAB en forma de matrices, estructuradas de tal manera que permita la comparación con los puntos de interés obtenidos de las mediciones con el software DIC.
- **Analizar y comparar resultados:** Comparar los resultados de las mediciones del campo de desplazamiento y los métodos de obtención de deformaciones, con respecto al modelo en elementos finitos. Para investigar cuantitativamente la medida del error, se utilizará el error promedio absoluto de los desplazamientos  $\overline{\mu_u}$  y el error promedio absoluto de las deformaciones  $\overline{\mu_\varepsilon}$  (definido en las ecuaciones 3.1 y 3.2, respectivamente), junto la desviación estándar para los desplazamientos  $s_u$  y para los campos de deformación  $s_\varepsilon$  (definidas en las ecuaciones 3.3 y 3.4 respectivamente),

$$\overline{\mu_u} = \frac{\sum_{i=1}^N \Delta u_i}{N} \quad 3.1$$

$$\overline{\mu_\varepsilon} = \frac{\sum_{i=1}^N \Delta \varepsilon_i}{N} \quad 3.2$$

en donde  $N$  representa el número de puntos de los datos;  $\Delta u_i$  representa el valor absoluto de la diferencia entre el valor del desplazamiento horizontal medido con el modelo en elementos finitos para el mismo punto  $i$  y  $\Delta \varepsilon_i$  representa el valor absoluto de la diferencia entre el valor de la deformación normal horizontal obtenida, con el modelo en elementos finitos para el mismo punto  $i$ .

$$s_u = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |\Delta u_i - \overline{\mu_u}|^2} \quad 3.3$$

$$s_{\varepsilon} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |\Delta\varepsilon_{x_i} - \overline{\mu_{\varepsilon}}|^2} \quad 3.4$$

en donde  $N$  representa el número de puntos de los datos;  $\Delta u_i$  representa el valor absoluto de la diferencia entre el valor del desplazamiento horizontal medido con el valor del mismo punto  $i$  en el modelo en elementos finitos;  $\overline{\mu_u}$  es el promedio de todos los valores  $\Delta u_i$ ;  $\Delta\varepsilon_{x_i}$  representa el valor absoluto de la diferencia entre el valor de la deformación horizontal obtenida con el valor del mismo punto  $i$  en el modelo de elementos finitos;  $\overline{\mu_{\varepsilon}}$  es el promedio de todos los valores  $\Delta\varepsilon_{x_i}$ .

### 3.1 Insumos

Para la realización del experimento y posterior análisis de resultados se requiere la compra y adquisición de los siguientes insumos:

- Probetas de acero SAE1020 trefilado, de sección cuadrada de 38 [mm] y largo 400 [mm].
- Pintura de base para las probetas, blanca, opaca y en spray.
- Pintura para las manchas en las probetas, negra opaca y en spray.
- Pintura gráfica negra, off-set, para impresión de patrón a través de timbre de goma.
- Timbre de goma con modelo de estampado de patrón de manchas digital.

## 3.2 Infraestructura

Los equipos y softwares, necesarios para la realización del experimento y posterior análisis de resultados se detallan a continuación:

### 3.2.1 Equipos

- Focos de iluminación led de alta potencia.
- Cámaras de Correlación de Imágenes Digitales Phantom, 1 Mega pixel de resolución de Dantec Dynamics.
- Máquina de ensayos Zwick Roell Z-100, con módulo para ensayos de flexión de tres puntos.

### 3.2.2 Software

- MATLAB 2013b
- Istra 4D
- ANSYS 14.5

## 4 Montaje experimental

### 4.1 Preparación de Probetas

Las probetas fueron pintadas con pintura blanca, opaca y en spray como base. Por su parte, para la lograr la impresión del patrón de manchas sobre las probetas, se probaron tres métodos distintos, de los cuales solo dos fueron aplicadas. Estos métodos son descritos a continuación:

- **Pintura Negra en Spray:** La técnica más utilizada en los estudios con cámaras DIC corresponde a pintar las muestras con pintura en lata spray. Esta consiste en presionar de manera sutil la boquilla del spray, haciendo un movimiento de barrido por sobre la muestra, a modo de dispersar de manera homogénea la pintura por la superficie. Esto generará manchas que en general cumplen con las 3 cualidades descritas anteriormente, con la desventaja de que las manchas serán de tamaño relativamente variable y la técnica dependerá mucho de la habilidad de quien aplique la técnica. Debido a lo anterior, esta técnica no fue aplicada en el estudio.
- **Salpicado de Pintura con Cepillo:** Como otra alternativa para conseguir un patrón de manchas adecuado se utilizó un cepillo dental común y tinta negra en spray. La pintura es aplicada en la cabeza del cepillo, desde la cual utilizando un dedo para desplazar hacia atrás las cerdas del cepillo, se salpica la probeta con la tinta negra como se muestra en la Figura 4.1. Se repite el proceso hasta que la región de interés quede aproximadamente con un 50% de cobertura con pintura negra, creando una distribución homogénea de las manchas (evitando concentraciones de manchas y manchas de tamaño desigual). El método es de fácil aplicación, pues no requiere mayor técnica y el tamaño de las manchas varía entre los 0.9 mm a 3.1 mm, siendo su distribución relativamente homogénea.

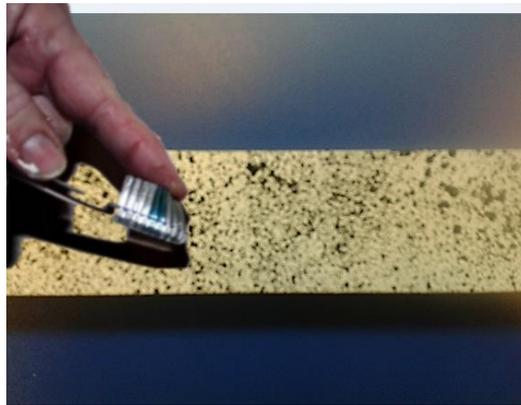


Figura 4.1: Técnica de aplicación del patrón de manchas con cepillo y tinta en spray.  
Fuente: Elaboración propia

- **Estampado del Patrón de Manchas con Timbre:** Para normalizar, sistematizar la preparación de las muestras y disminuir la posibilidad de agregar errores sistemáticos en las mediciones, se fabricó un timbre de goma de 40 mm x 80 mm que permite estampar el patrón de manchas por contacto. El patrón del timbre fue elaborado por cortado laser, a partir de un patrón de manchas digital recomendado por el proveedor del sistema DIC.

Debido a que la superficie de la probeta metálica no es absorbente, el uso de tinta común para estampar no es posible, pues tiene tendencia a escurrirse y el resultado es insatisfactorio. Para sortear esta condición, se utiliza un rodillo de impresión el cual se impregna con tinta gráfica, para luego transferir de manera uniforme una capa de pintura al timbre de goma, para finalmente estampar el patrón sobre la probeta. El tamaño de las manchas obtenido, varía entre los 3,1 mm a 3,3 mm de ancho. Los elementos anteriormente descritos se muestran en la Figura 4.2 a), mientras que el resultado del método se muestra en b).

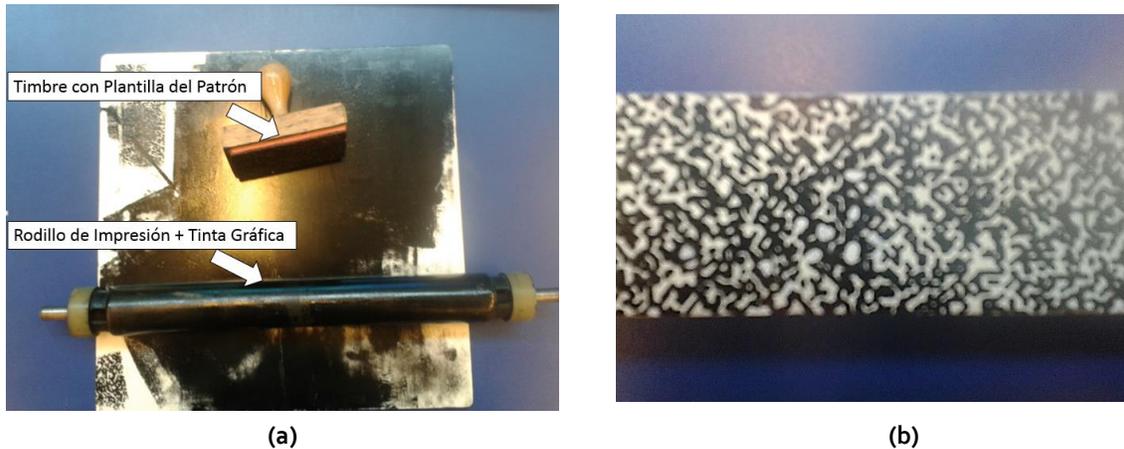


Figura 4.2: Método de Estampado del Patrón de Manchas con Timbre. En a) Elementos necesarios y (b) Resultado  
Fuente: Elaboración propia

## 4.2 Ensayo de Flexión

Para el ensayo de flexión, se utiliza un modelo simple (representado en la Figura 4.3) de dos apoyos, separados por un largo  $l = 200$  mm, sometido a flexión por la carga que ejerce el sistema en el centro de la probeta. Los ensayos son realizados de manera tal que no se alcance la fluencia del material, a modo de asegurar la repetitividad de los ensayos para la misma probeta.

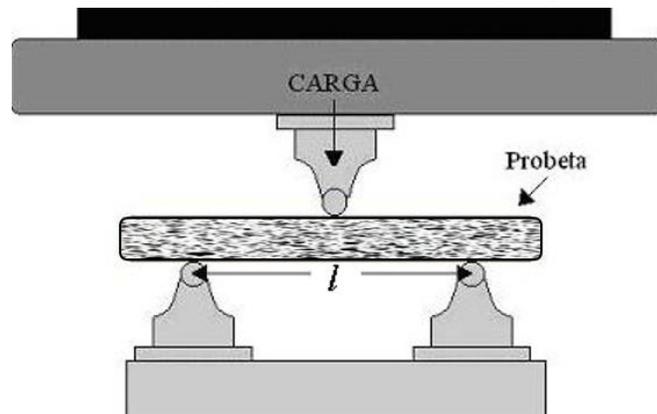


Figura 4.3 Montaje del ensayo de flexión de tres puntos.  
Fuente: Elaboración propia

El límite de la carga de flexión viene dado por la ecuación 4.1 y el límite de fluencia del acero SAE 1020,  $\sigma_y = 27 \left[ \frac{kg}{mm^2} \right]$ .

$$\sigma_f = \frac{3FL}{2a^3} \quad 4.1$$

En donde  $\sigma_f$  equivale al esfuerzo de flexión en el centro de la probeta, F es la carga central aplicada sobre la barra, a es el tamaño de la sección cuadrada de la barra y L es la distancia entre apoyos. Haciendo  $\sigma_y = \sigma_f$  y despejando se tiene que el límite de la carga a aplicar en flexión equivale a:

$$F_{limite} = 48,4 \text{ [kN]} \quad 4.2$$

Se fija en el software de la máquina de ensayos de flexión, la velocidad de avance de la cuña en 2 mm/min y la fuerza límite de aplicación en 30 [kN]. La Figura 4.4 muestra el montaje del módulo de flexión en la máquina de ensayos de tracción Zwick Roell Z-100.



(a) Disposición espacial de las cámaras DIC (carcasa blanca) y focos de iluminación (carcasa negra).

(b) Módulo de ensayo de flexión de tres puntos y probeta con patrón de manchas

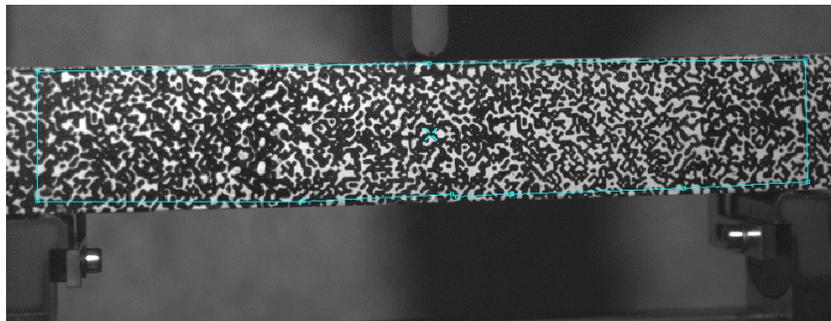
Figura 4.4 Vistas del montaje experimental.  
Fuente: Elaboración propia

### 4.3 Cámaras y Software DIC

Los desplazamientos fueron determinados a través de una correlación de imágenes digitales, utilizando el equipo de Dantec Dynamics, junto con el software Istra 4D. Se fija una distancia entre el lente de las cámaras y el centro de la probeta de 441 [mm], junto con un ángulo entre las cámaras de 25,1°, y se realiza la calibración de las cámaras con el módulo de calibración propio del software Istra 4D. Se realizaron una amplia cantidad de pruebas con el software Istra 4D, hasta encontrar una configuración óptima de las condiciones de iluminación y orientación de las cámaras. De esta forma, una vez definidas las condiciones del montaje, se realizaron las mediciones del ensayo de flexión.

Las imágenes de las probetas sometidas al ensayo de flexión, fueron capturadas por las cámaras CCD Phantom Q-450 de 1 Mega pixel de resolución. Debido a que el caso de estudio solo considera el estado inicial y final de las probetas sometidas a deformación, la frecuencia de adquisición de las cámaras se fija en 4 fotogramas por segundo. El tiempo de exposición de las cámaras, se define en  $30\mu\text{s}$ .

El área seleccionada para el cálculo de los desplazamientos se muestra en la Figura 4.5, en la cual el polígono celeste define la región entre los apoyos de la probeta y es donde se hará la correlación de imágenes. Como se mencionó en la sección 2.2, no hay un tamaño óptimo de sub set definido a priori, por lo que se seleccionan los valores para el sub set de referencia de  $17\times 17$ ,  $19\times 19$ ,  $21\times 21$ ,  $23\times 23$ ,  $25\times 25$ ,  $27\times 27$ ,  $29\times 29$  y  $33\times 33$  pixeles, junto con una distancia de paso de 5 pixeles entre los puntos de interés para el cálculo de los desplazamientos de las probetas con el patrón de manchas tipo *timbre* y tipo *cepillo*.



**Figura 4.5:** Área seleccionada para el cálculo de desplazamientos  
Fuente: Elaboración propia a partir de ISTR4 4D

El sistema de coordenadas global se fija en el centro de la probeta con el módulo de análisis de los datos propio en el software Istra 4D. Los datos se exportan al formato “.*hdf5*”, que contiene los datos de las coordenadas del sistema y del desplazamiento de los puntos de interés medidos. Finalmente, los datos son importados al entorno MATLAB, con las funciones “*Importar\_datos.m*”, “*get\_disp\_x().m*”, “*get\_disp\_y().m*” y “*get\_disp\_z().m*”, adjuntas en la sección de Anexo D.1, al final de este informe.

El proceso de montaje de las cámaras y del cálculo de los desplazamientos mediante el software Istra 4D, se explicita en el Anexo A, el cual contiene información y recomendaciones de cómo realizar mediciones con el sistema DIC y con el software Istra 4D para la obtención de desplazamientos.

## 5 Resultados y Análisis

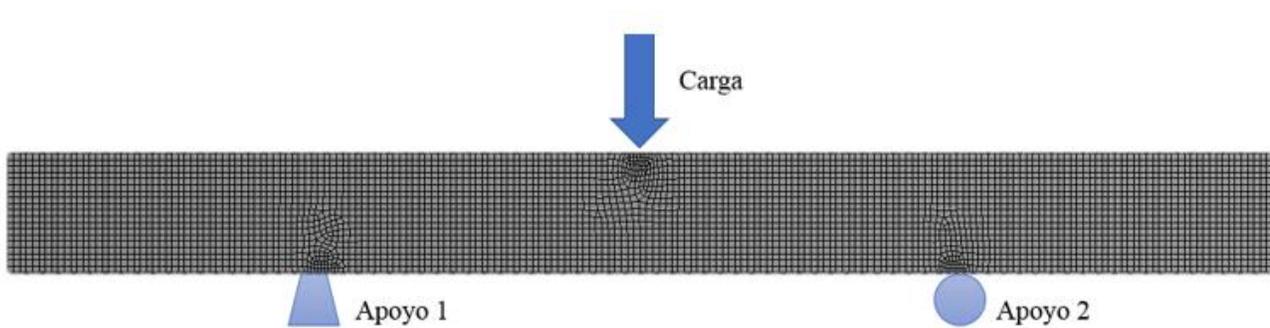
Esta sección será llevada a cabo mediante el análisis respectivo de cada resultado obtenido, y luego se hará un análisis general que englobe todas las discusiones de los resultados previos. En primer lugar, se describe la solución del modelo de elementos finitos desarrollado en ANSYS que se utiliza como solución referencial al problema de flexión estudiado. En la siguiente sección, se muestran las mediciones de los desplazamientos experimentales obtenidos, junto con su respectivo error para los distintos valores de los parámetros de correlación. Finalmente, se presentan los campos de deformación obtenidos y su error asociado, para cada uno de los valores que definen los métodos de:

- Suavizado de desplazamientos y posterior aplicación del gradiente.
- Aproximación de la deformación por mínimos locales cuadrados (Método PLS).
- Reducción del ruido en el campo de deformación, utilizando una función de costos regularizada.

### 5.1 Modelo Elementos Finitos

La geometría utilizada para el análisis corresponde a una extrusión de 400 [mm] de una sección cuadrada de 38 [mm]. Para el cálculo del software, se utilizó un mallado del tipo hexa-dominante con refinación local en la zona de aplicación de fuerza y apoyos (ver Figura 5.1), con un total 74.173 elementos, la calidad de la malla promedio es de 0,9869 con un máximo de 1. Los parámetros principales y estadísticas que definen el mallado, se resumen en la sección de Anexo B. La carga y condiciones de borde consideradas corresponden a:

- Cargas de 30,2 [kN], en donde la fuerza es aplicada en el centro de la barra y a lo largo de un elemento de línea en la cara superior de la barra.
- Apoyos simples ubicados a 100 [mm] del centro de la barra. La restricción de movimiento se aplica a lo largo de un elemento de línea, en la cara inferior de la barra.

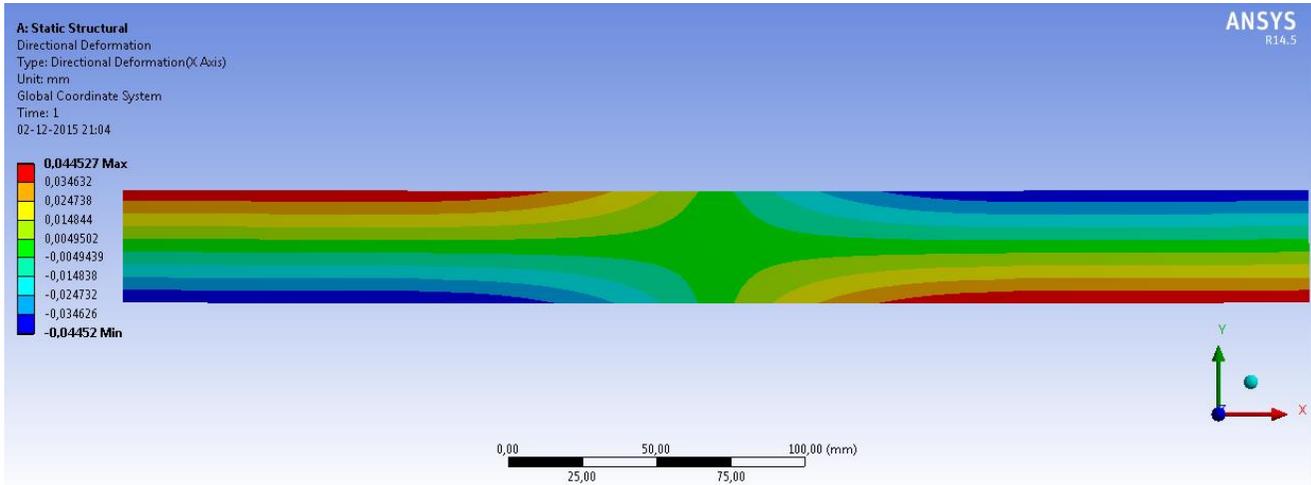


**Figura 5.1: Mallado, carga y condiciones de borde consideradas en el modelo de elementos finitos.**

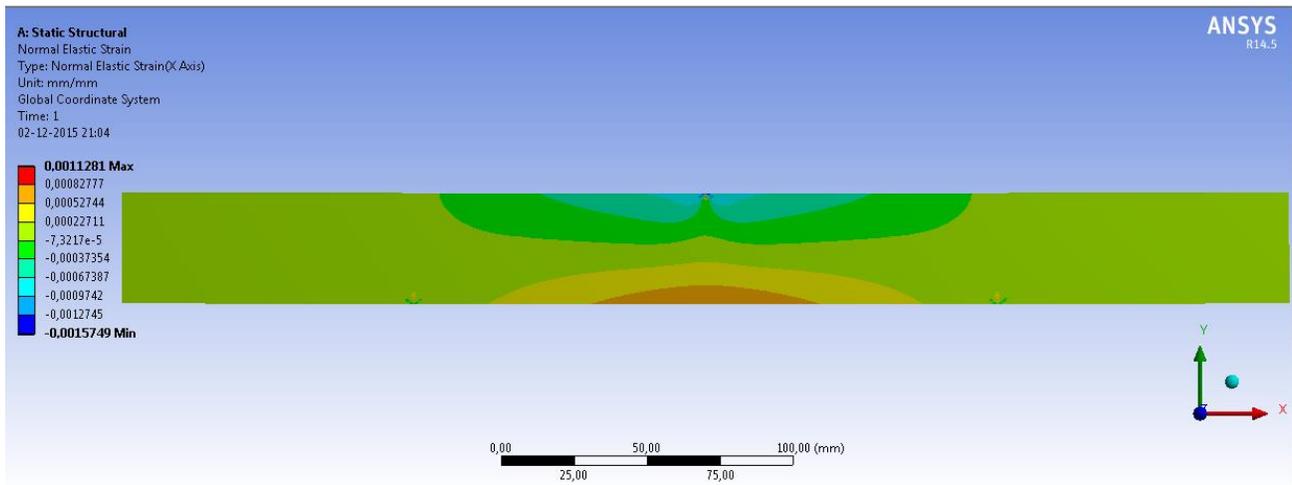
**Fuente: Elaboración propia a partir de ANSYS v.14.5.**

La distribución espacial de los desplazamientos horizontales ( $u_x$ ), se muestra en la Figura 5.2, mientras que la distribución espacial de la deformaciones normales horizontales ( $\epsilon_x$ ) se mues-

tran en la Figura 5.3. En ambas figuras, los valores máximos son representados en rojo y los mínimos en azul. La Tabla 5.1 resume los valores máximos y mínimos de resultados obtenidos para el modelo de distribución de desplazamientos y deformaciones en elementos finitos.



**Figura 5.2:** Distribución espacial de desplazamientos horizontales  $u_x$  del modelo en elementos finitos.  
**Fuente:** Elaboración propia a partir de ANSYS v.14.5.



**Figura 5.3:** Distribución espacial de la deformación normal horizontal  $\epsilon_x$  del modelo en elementos finitos.  
**Fuente:** Elaboración propia a partir de ANSYS v.14.5.

**Tabla 5.1.** Valores máximos y mínimos de los desplazamientos y deformaciones, obtenidos por el modelo de elementos finitos, para los casos de estudio.

Carga [kN]	máx( $u_x$ ) [mm]	mín( $u_x$ ) [mm]	máx ( $\epsilon_x$ )	mín ( $\epsilon_x$ )
30,2	0,041940	-0,041906	0,0007887	-0,001064

**Fuente:** Elaboración propia

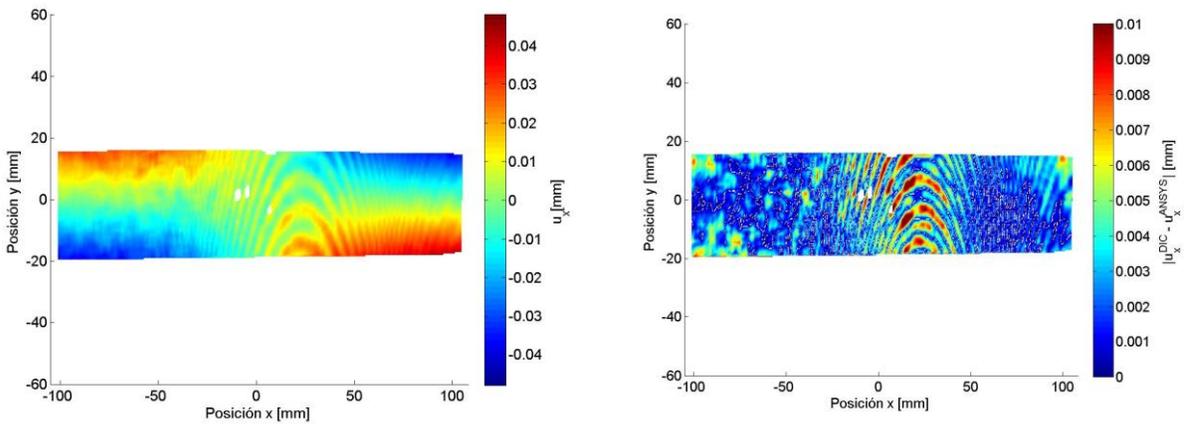
## Análisis

Al ver la Figura 5.2 , se aprecia que los desplazamientos horizontales siguen una distribución simétrica en torno a los ejes vertical y horizontal del sistema de coordenadas. Dicha distribución se explica por una parte, debido a que bajo del eje neutro, al alejarse de la línea de acción de la carga aplicada (centro de la probeta), los vectores de desplazamiento aumentan en magnitud en dirección hacia fuera de la probeta, por lo que esta tiende a extenderse. Por otra parte, sobre el eje neutro, los vectores de desplazamiento aumentan en magnitud a medida que se alejan del centro de la probeta, pero en dirección opuesta, por lo que la probeta tiende a contraerse. Este fenómeno, se refleja en la distribución de la deformación normal horizontal (ver Figura 5.3), en donde la zona inferior de la probeta concentra los valores de deformación positivos y la zona superior de la probeta concentra los valores de deformación negativos. La geometría de línea, en donde se aplica la carga y las condiciones de borde, genera áreas de concentración de esfuerzos, lo que explica que, en el campo de deformación se generen pequeñas perturbaciones. Estas últimas áreas, dependerán de la malla utilizada por el modelo, específicamente, si el tamaño de los elementos logra captar la totalidad de los desplazamientos en estas zonas. Para evitar este efecto, otros estudios proponen la realización de ensayos de flexión de cuatro puntos (como el realizado por X.Dai [3]), de modo que se evitan las zonas con influencia de concentraciones de esfuerzo por contacto, en la región de interés analizada.

La buena calidad de la malla, sumado al hecho de que la región de interés no contiene zonas en que la solución dependa de la malla, permite concluir que el modelo de elementos finitos es una buena solución base para la medición del error de los resultados, a través de la comparación de la medición experimental de los desplazamientos y también de los métodos de obtención de deformación considerados

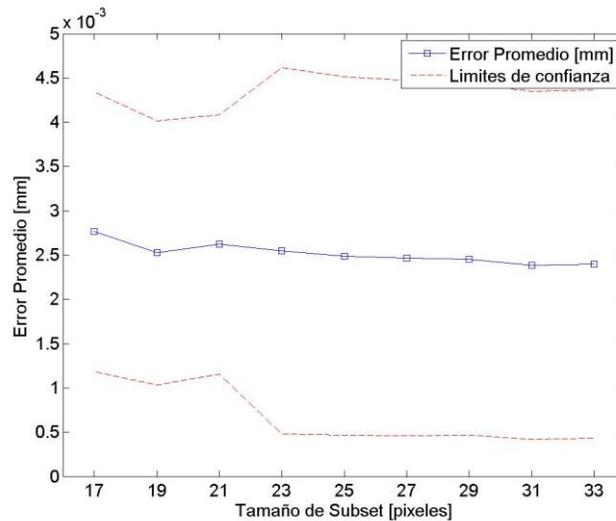
### 5.2 Medición de Desplazamientos

La Figura 5.4, muestra los resultados para las probeta obtenida con la técnica de salpicado de pintura spray y cepillo (patrón de manchas tipo “*cepillo*”), para un sub set de referencia de 23x23 y distancia de paso 5 pixeles. En esta figura, la escala de colores en (a) grafica los desplazamientos horizontales  $u_x$ , a su vez, la escala de colores en (b), ilustra la medida del error, representado como el valor absoluto de la diferencia entre el modelo de elementos finitos con los desplazamientos horizontales experimentales. Los espacios blancos de la Figura 5.4 corresponden a la ausencia de datos proporcionados por el software Istra 4D, producto del no reconocimiento de un patrón de manchas adecuado en esas áreas.



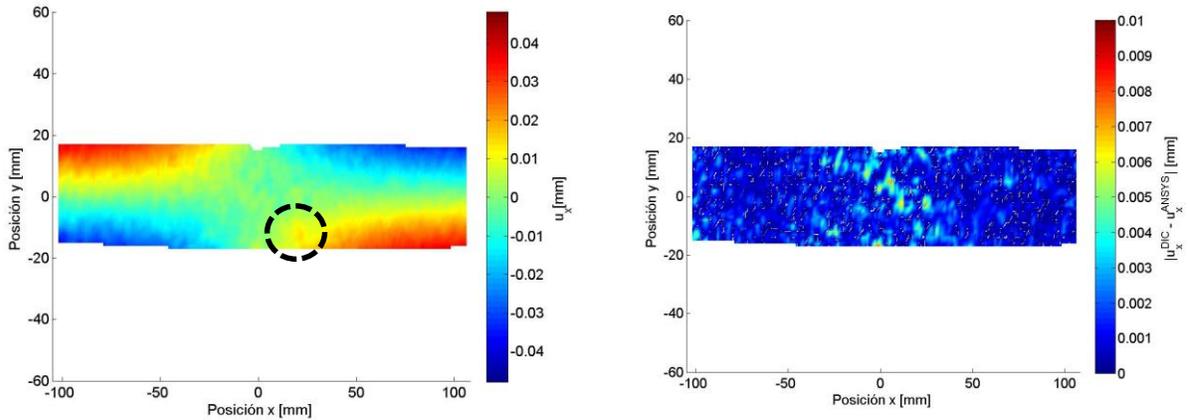
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura 5.4. Resultados de la medición DIC, para la probeta con patrón tipo “Cepillo”, sub set de referencia 23x23, y distancia de paso 5 pixeles.**  
**Fuente: Elaboración propia**

Las mediciones restantes para la probeta con el patrón de manchas tipo “cepillo”, no se incluyen en esta sección, estos resultados pueden ser consultados en la sección de Anexos C.1. No obstante, la totalidad de los resultados para la probeta con el patrón estampado con timbre se resumen en la Figura 5.5 , en donde se muestra el gráfico del error promedio (línea azul) en función del tamaño de sub-set, calculado como el promedio del valor absoluto de la diferencia entre los desplazamientos y el modelo en ANSYS. Las líneas rojas representan el intervalo de confianza de los datos, calculado como la desviación estándar descrita en la ecuación 3.3.



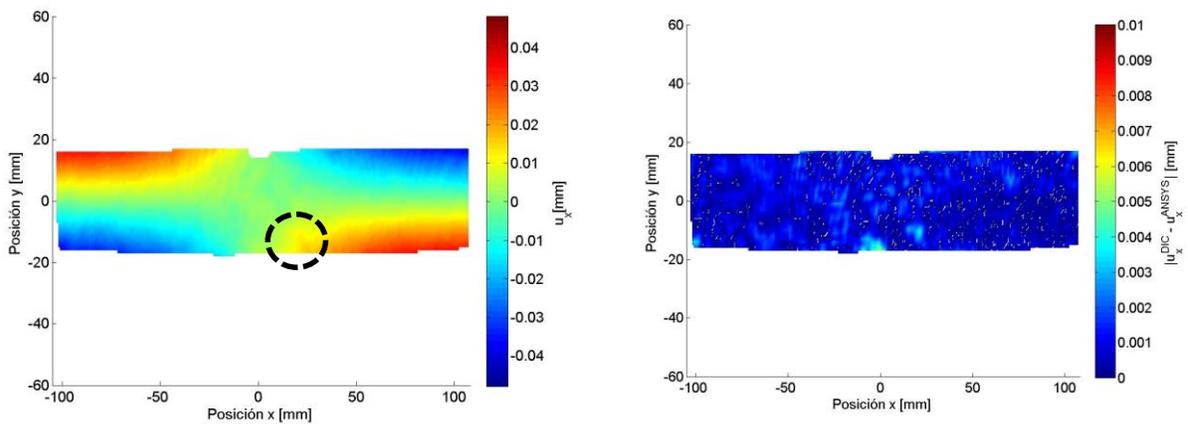
**Figura 5.5: Error promedio y desviación estándar de probetas tipo “cepillo” en el cálculo de desplazamientos, en función del tamaño de sub set.**  
**Fuente: Elaboración propia a partir de MATLAB R2013b.**

La Figura 5.6 y Figura 5.7, grafican los resultados de las mediciones de los desplazamientos para las probetas para con el patrón de manchas tipo “*timbre*”, obtenidas con sub sets de referencia de 17x17 y 27x27 respectivamente. En ambas figuras, la escala de colores en (a) grafica los desplazamientos horizontales  $u_x$ , a su vez, la escala de colores en (b), ilustra la medida del error, representado como el valor absoluto de la diferencia entre el modelo de elementos finitos con los desplazamientos horizontales experimentales.



(a) Desplazamientos horizontales  $u_x$  (b)Desviación estándar de los datos (MATLAB)  
**Figura 5.6. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre”, sub set de referencia 17x17, y distancia de paso 5 pixeles.**

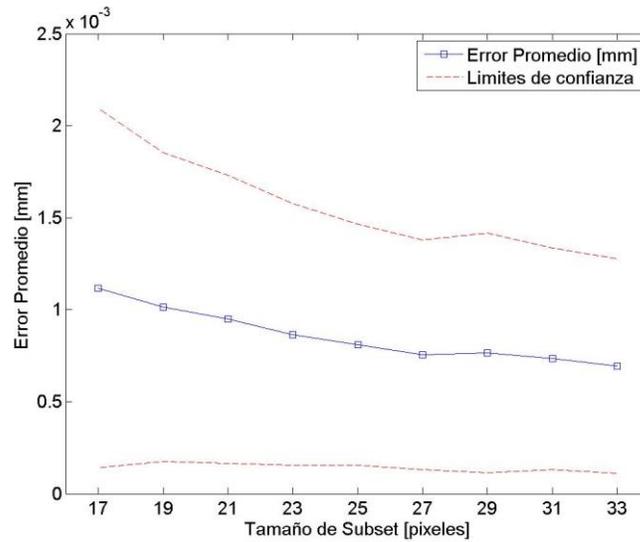
Fuente: Elaboración propia a partir de MATLAB R2013b



(a) Desplazamientos horizontales  $u_x$  (b)Desviación estándar de los datos (MATLAB)  
**Figura 5.7. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre”, sub set de referencia 27x27, y distancia de paso 5 pixeles.**

Fuente: Elaboración propia a partir de MATLAB R2013b

Las mediciones obtenidas, para los parámetros de correlación restantes, pueden ser consultadas en la sección de Anexos C.2. No obstante, la totalidad de los resultados para la probeta con el patrón estampado con timbre, se resumen en la Figura 5.8, en donde se muestra el gráfico del error promedio (línea azul) en función del tamaño de sub set, calculado como el promedio del valor absoluto de la diferencia entre los desplazamientos y el modelo en ANSYS. Las líneas rojas representan el intervalo de confianza de los datos, calculado como la desviación estándar descrita en la ecuación 3.3.



**Figura 5.8: Error promedio y desviación estándar para las probetas tipo “timbre”, para el cálculo de desplazamientos, en función del tamaño de sub set.**

**Fuente: Elaboración propia a partir de MATLAB R2013b.**

## Análisis

Los resultados para el patrón del tipo cepillo presentan efectos de Aliasing, lo que se evidencia al notar que las desviaciones de los desplazamientos de la Figura 5.4 a) siguen un patrón tipo *moire*. Lo anterior se hace evidente, al comparar la Figura 5.4 b) con la Figura 2.5. Este efecto, incide en el error promedio y desviación estándar exageradas para todos los sub sets de referencia considerados en el cálculo de los desplazamientos (ver Figura 5.5). Por el contrario, este efecto no se observa en las mediciones realizadas en la probeta con el patrón de manchas tipo “Timbre”, por lo que los análisis de todos los resultados se centran en los desplazamientos obtenidos para esta muestra (Figura 5.6 y Figura 5.7).

Por un lado, comparando la Figura 5.6 (b) y la Figura 5.7 (b), se puede apreciar cómo la distribución del error se hace más homogénea para sub sets de referencia de mayor tamaño, especialmente en las zonas del centro de la probeta, cuyos valores de desplazamiento son cercanos a 0. Se observa además, la presencia de concentraciones de error del orden de  $10^{-3}$  en la esquina inferior izquierda y la parte inferior central de la probeta. Además, para los desplazamientos obtenidos con el sub set de 17x17 píxeles, existe una pequeña perturbación aislada en la zona demarcada por una circunferencia negra en la Figura 5.6 a). Esta perturbación, se atenúa para sub sets referenciales de mayor tamaño (ver circunferencia negra Figura 5.7 (b)).

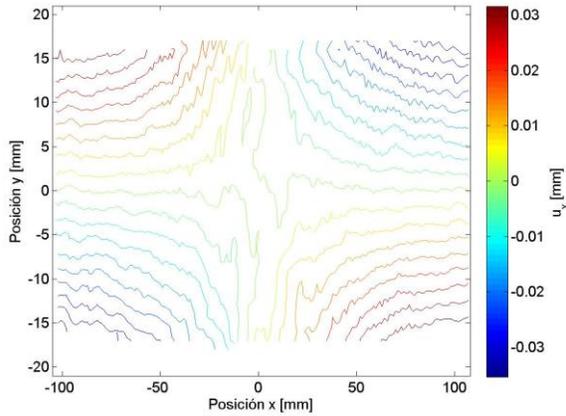
Por otra lado, se observa en la Figura 5.8, que para sub sets de referencia más amplios se reduce la escala del error y la variabilidad del mismo, situación que confirma lo establecido en el Marco Teórico. Además, la escala del error promedio para todos los sub set de referencia es de menor orden de magnitud en comparación con los desplazamientos horizontales máximos y mínimos del modelo de elementos finitos (Tabla 5.1).

Para tamaños de sub sets más pequeños, se producen errores en la correlación asociados al tamaño de las manchas del patrón, debido a que existen sub sets que encierran zonas completamente negras o completamente blancas. Lo anterior, se explica debido a que al disminuir el tamaño del sub set, se produce una tendencia a aumentar el error promedio de los datos, así como sus límites de confiabilidad. Por el contrario, al aumentar el tamaño de sub set, más allá de los 27x27 píxeles, no se observa una tendencia clara en la disminución del error (como la que existe entre los 17x17 y 27x27 píxeles). Si bien, la deformación a la que se somete la probeta no es de gran magnitud, la afirmación que se hace sobre los tamaños mayores de los sub set, no describiría con precisión zonas con deformaciones heterogéneas, por lo que se explicaría que el error no disminuya más.

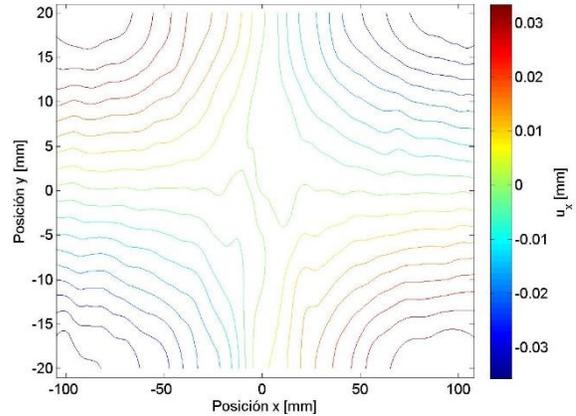
Entonces, considerando lo mencionado en el apartado anterior, se selecciona la probeta con el patrón tipo “*timbre*”, utilizando un tamaño de sub set de 27 píxeles y una distancia de paso de 5 píxeles, como referencia para el cálculo de deformación de los tres métodos propuestos.

### 5.3 Método de Suavizado de Desplazamientos y Aplicación del Gradiente

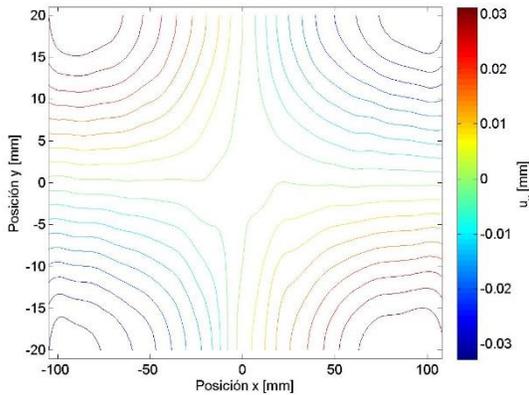
Se implementó en MATLAB, el método obtención de deformaciones por medio del suavizado de los desplazamientos y posterior obtención del campo de deformación por medio de la aplicación del gradiente, cuyo código se encuentra en la sección de Anexo C.3. La intensidad del suavizado de los datos se define por el valor escalar ‘s’, cuya influencia se grafica en la Figura 5.9, en donde se muestran las líneas de contorno que describen los valores de los desplazamientos horizontales  $u_x$ . En la Figura 5.9 (a), se muestran los desplazamientos sin suavizar, mientras que en (b), (c) y (d) se muestran los desplazamientos suavizados para los valores de suavizado ‘s’ de 10, 100 y 1000 respectivamente.



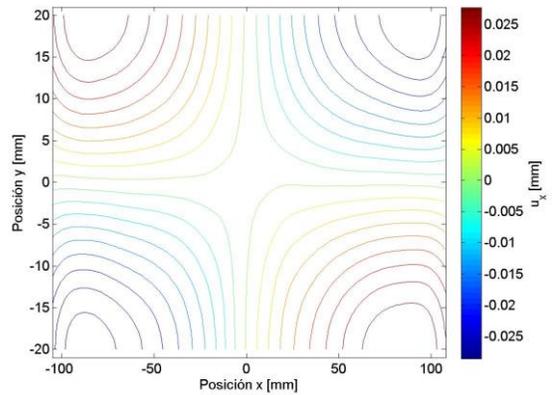
(a) Desplazamientos  $u_x$  sin suavizar.



(b) Desplazamientos  $u_x$  suavizados para  $s = 10$ .



(c) Desplazamientos  $u_x$  suavizados para  $s = 100$ .



(d) Desplazamientos  $u_x$  suavizados para  $s = 1000$ .

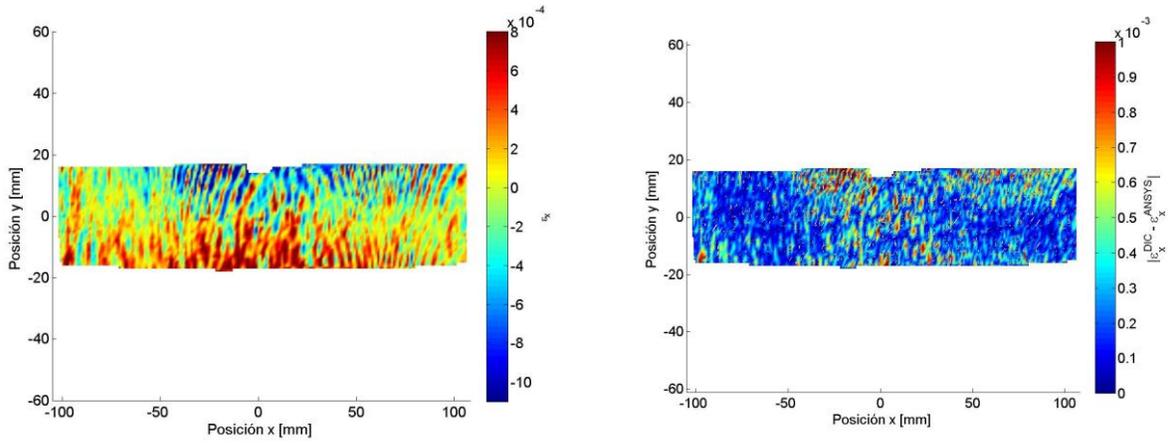
Figura 5.9. Líneas de contorno de los desplazamientos horizontales  $u_x$ , para distintos valores de suavizado, para la probeta con patrón tipo “Timbre”, sub set de referencia 27x27, y distancia de paso 5 píxeles.

Fuente: Elaboración propia

La Figura 5.10 muestra el campo de deformación obtenido por aplicación del gradiente directamente en los desplazamientos experimentales, cuyo error absoluto promedio equivale a  $2.549 \times 10^{-4}$  y desviación estándar de  $2.235 \times 10^{-4}$ .

Los campos de deformación resultantes, para los valores de suavizado ‘s’ de 10, 100, 1000 y 1000, junto con la distribución del error asociada, se muestran en Figura 5.11, Figura 5.12, Figura 5.13 y Figura 5.14, respectivamente. En estas figuras, la escala de colores en (a) grafica la deformación normal horizontal  $\epsilon_x$ , mientras que la escala de colores en (b), ilustra la medida del error, representado como el valor absoluto de la diferencia entre el modelo de elementos finitos con la deformación normal horizontal  $\epsilon_x$  obtenida. Los resultados de los campos de deformación obtenidos, para una serie de valores ‘s’ entre 0.1 y 1000, se resumen en la Figura 5.15, en donde se muestra el gráfico del error promedio (puntos azules), en función del parámetro de intensidad de

suavizado  $S$ , calculado como el promedio del valor absoluto de la diferencia entre las deformaciones y el modelo en ANSYS. Las líneas rojas representan el intervalo de confianza de los datos, calculado como la desviación estándar descrita en la ecuación 3.4.

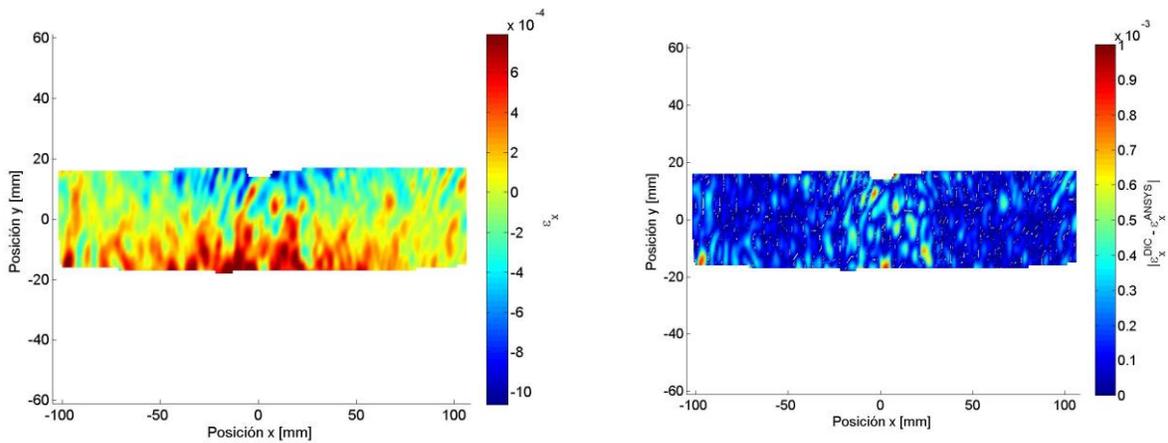


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.10. Resultados de la aplicación del gradiente direccional, sobre desplazamientos sin suavizar.**

Fuente: Elaboración propia a partir de MATLAB R2013b

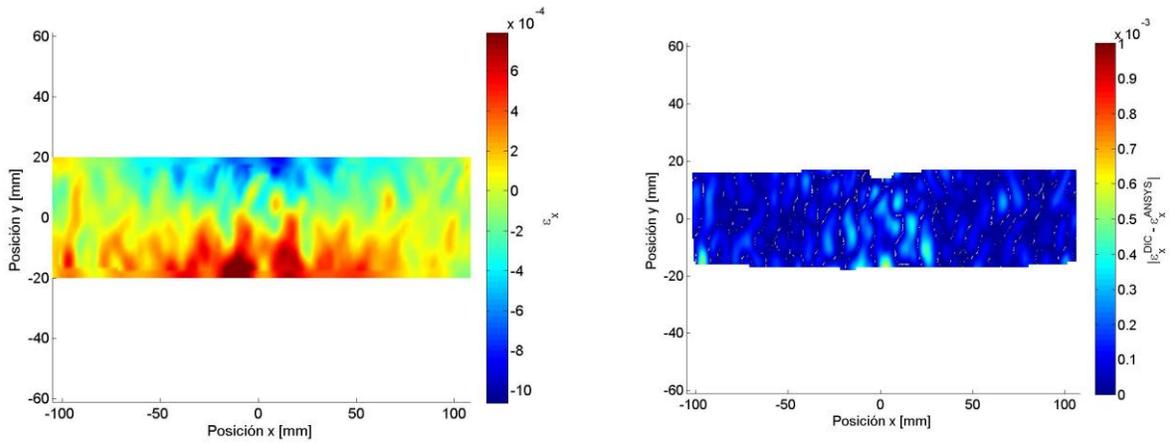


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.11. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método smoothn.m y valor de suavizado  $s=1$ .**

Fuente: Elaboración propia a partir de MATLAB R2013b

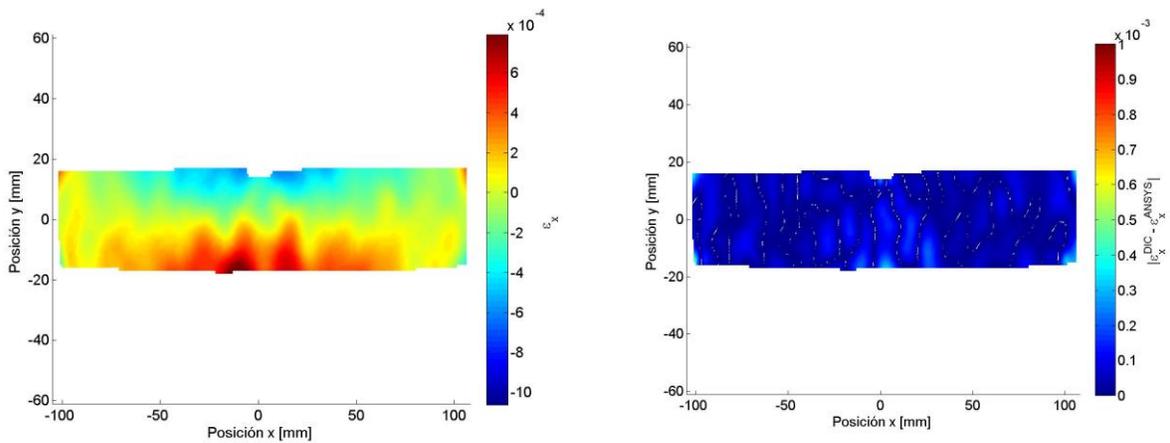


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.12. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método "smoothn.m" y valor de suavizado s=10.**

**Fuente: Elaboración propia a partir de MATLAB R2013b**

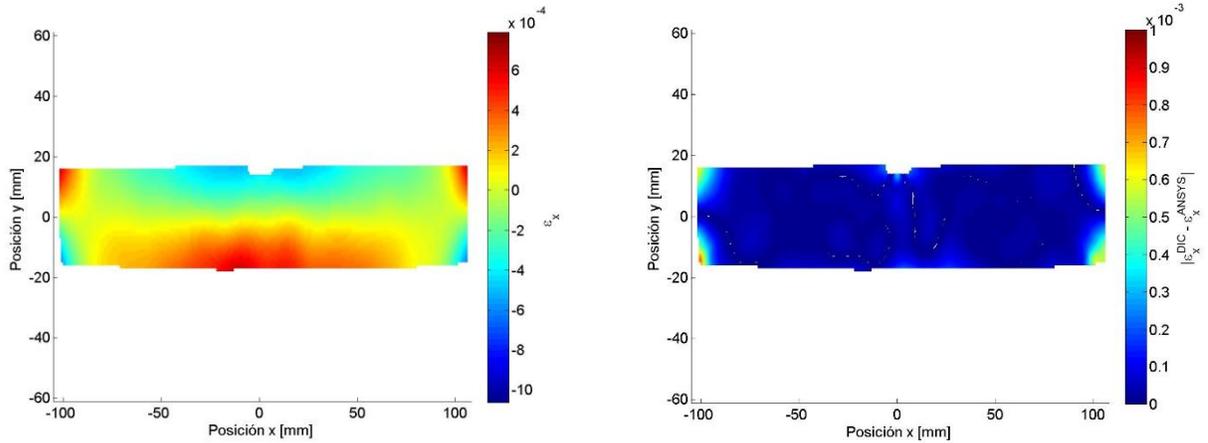


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.13. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método smoothn.m y valor de suavizado s=100.**

**Fuente: Elaboración propia a partir de MATLAB R2013b**

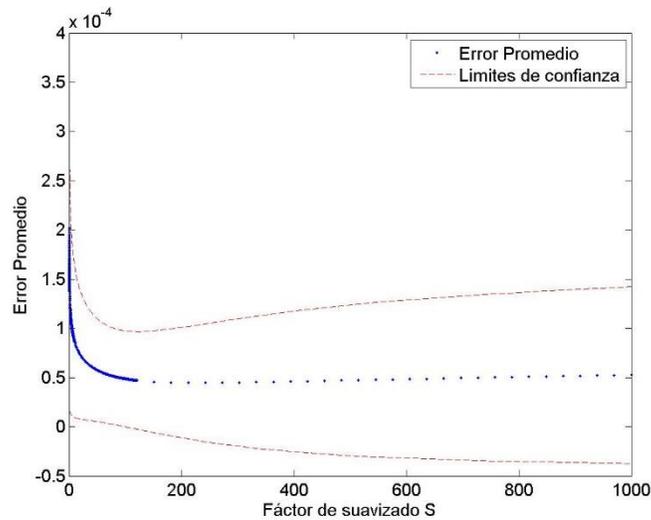


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.14. Resultados de la aplicación del gradiente direccional, sobre desplazamientos suavizados por el método `smoothn.m` y valor de suavizado  $s=1000$ .**

**Fuente: Elaboración propia a partir de MATLAB R2013b**



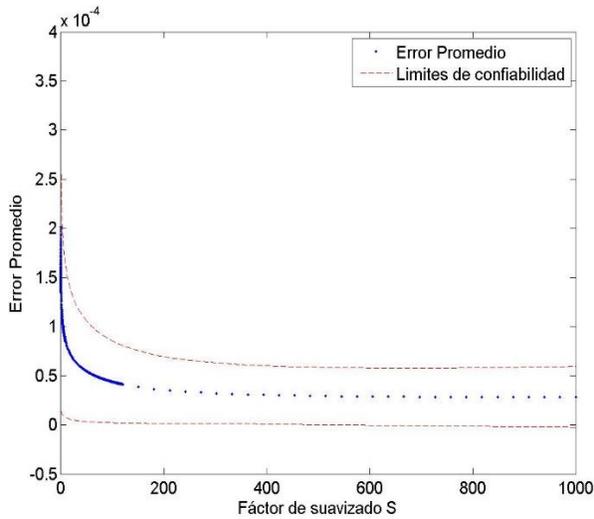
**Figura 5.15: Error promedio y desviación estándar del cálculo del campo de deformación, a partir de desplazamientos horizontales, para distintos valores de suavizado  $s$  del método “Smoothn.m”**

**Fuente: Elaboración propia a partir de MATLAB R2013b**

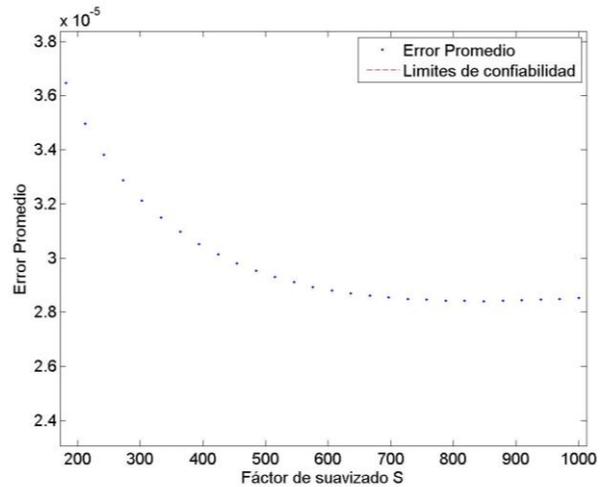
## Análisis

La Figura 5.9 refleja el efecto de suavizado de los datos y el ruido en las líneas de contorno de los desplazamientos sin suavizar, lo que hace evidente la amplificación de los errores al aplicar el gradiente sobre los datos experimentales. Comparando la Figura 5.9 b), c) y d) con la Figura 5.9 a), se aprecia cómo las líneas de contorno de desplazamientos con forma de *dientes de sierra*, se atenúan para valores de suavizado mayor. En consecuencia, los campos de deformación obtenidos a partir de los desplazamientos suavizados (Figura 5.11, Figura 5.12, Figura 5.13 y Figura 5.14), muestran mayor regularidad que el campo de deformación obtenido por diferenciación numérica directa (Figura 5.10).

Por un lado, la zona con desviación en los desplazamientos del borde inferior izquierdo de la probeta (observada en el análisis de la sección 5.3.1), genera un error en los campos de deformación obtenidos por aplicación del gradiente. Esta desviación se hace presente para todos los valores del parámetro de suavizado 's', sin embargo, su influencia se atenúa a medida que se tienen suavizados mayores. Una consecuencia análoga a la anterior, es la desviación de los desplazamientos de la zona del centro inferior de la probeta, en donde ocurre una discontinuidad en los valores máximos de la deformación. Para valores de 's' superiores a 100, esta discontinuidad comienza a homogeneizarse con los valores aledaños, teniendo como consecuencia la disminución de la escala de los valores máximos de esa zona, situación que se refleja al ver Figura 5.14 a) y la Figura 5.15 a). Por otro lado, el método al suavizar las esquinas, genera un error que se hace visible para valores de suavizado 's' superiores a 100. El efecto anterior, se aprecia en la Figura 5.9 a), b) y c) de manera más evidente en d), en donde las líneas de contorno de los desplazamientos que contienen los valores máximos y mínimos, comienzan a reducir su efecto en las esquinas, generando áreas cada vez más circulares y gradientes de valores, los cuales son interpretados como falsas concentraciones de deformación. Si no se toma en cuenta este efecto, al observar el comportamiento del gráfico de la Figura 5.15, se podría llegar a la falsa conclusión de que la variabilidad del error promedio, comienza a aumentar para valores de 's' = 91.89. Sin embargo, si se omite el efecto de este error, tomando en cuenta únicamente los datos de la probeta entre -95 y 95 mm, se tiene el comportamiento del error promedio y la variabilidad del mismo representado en el gráfico de la Figura 5.16 a). En este, se observa que el valor del error promedio, deja de variar drásticamente para el valor de  $s = 272,55$ . La Figura 5.16 b) muestra un acercamiento del caso anterior para los valores de  $s$  entre 200 y 1000, en donde para el valor de  $s = 848,44$  se alcanza el error promedio mínimo de  $2,841 \cdot 10^{-5}$  y desviación estándar de  $2,917 \cdot 10^{-5}$ .



(a) Error promedio y límites de confiabilidad para  $s$  entre 0 y 1000



(b) Error promedio y límites de confiabilidad para  $s$  entre 200 y 1000

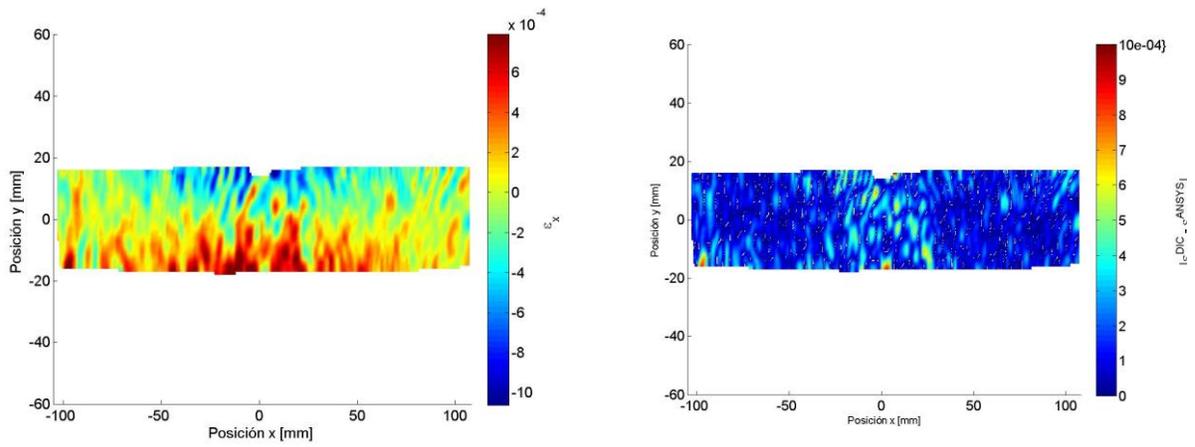
**Figura 5.16:** Error promedio y desviación estándar del cálculo del campo de deformación, a partir de desplazamientos horizontales, para distintos valores de suavizado  $s$  del método “Smoothn.m”, considerando solo los valores de la probeta entre  $x=-95$  y  $x=95$

Fuente: Elaboración propia a partir de MATLAB R2013b

## 5.4 Método PLS

Se implementó en MATLAB, el método obtención de deformaciones por aproximación de mínimos cuadrados, cuyo código se encuentra en la sección de Anexo D.3. Esta función se define por el valor escalar ‘ $M$ ’, que determina el tamaño de la ventana de cálculo de  $(2M + 1 \times 2M + 1)$  datos. Los resultados obtenidos corresponden a las deformaciones calculadas para las matrices de desplazamientos de las muestras mencionadas en la sección 5.2, con ventanas de cálculo de tamaño que van entre  $5 \times 5$  a  $33 \times 33$  datos.

Por una parte, la Figura 5.17, Figura 5.18 y Figura 5.19, muestran los resultados para las ventanas de cálculo de  $5 \times 5$ ,  $17 \times 17$  y  $33 \times 33$  datos, respectivamente. En estas figuras, la escala de colores en (a) grafica la deformación normal horizontal  $\epsilon_x$ , mientras que la escala de colores en (b), ilustra la medida del error representada como el valor absoluto de la diferencia entre el modelo de elementos finitos y la deformación normal horizontal  $\epsilon_x$  obtenida. Por otra parte, la Figura 5.20 resume la totalidad de los resultados estudiados para las ventanas de cálculo de tamaño entre  $5 \times 5$  a  $33 \times 33$  datos. En la misma figura, se muestra el gráfico del error promedio (puntos azules), en función del tamaño de sub set calculado como el promedio del valor absoluto de la diferencia entre las deformaciones y el modelo en ANSYS. Las líneas rojas representan el intervalo de confianza de los datos, calculado como la desviación estándar descrita en la ecuación 3.4.

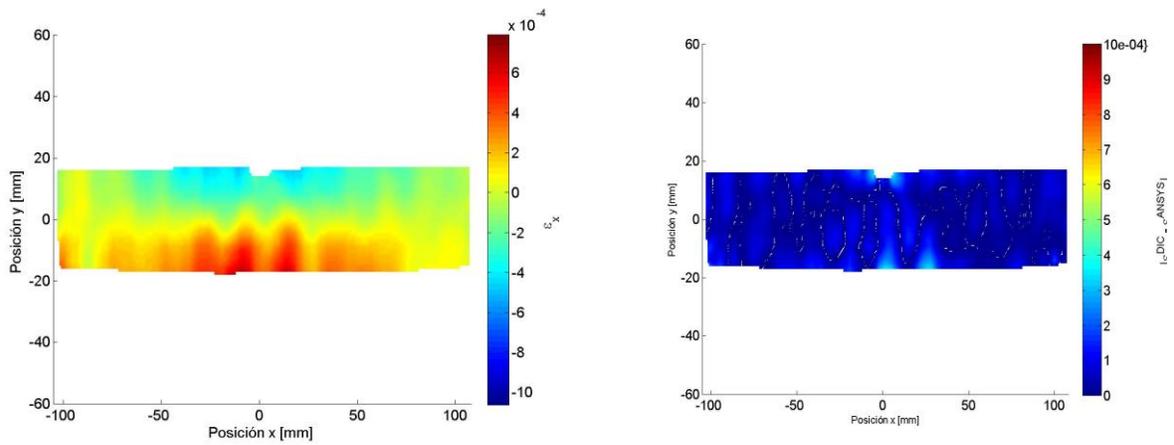


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.17: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo de 5x5 datos.**

**Fuente: Elaboración propia a partir de MATLAB R2013b**

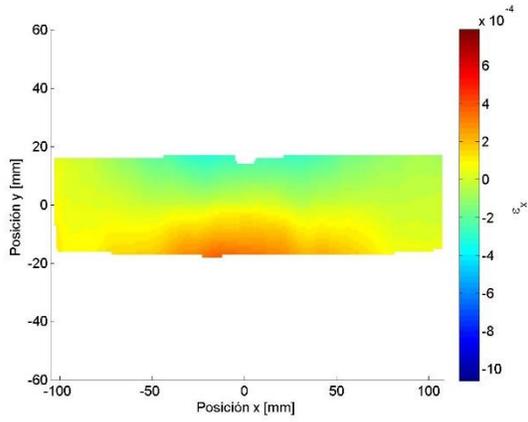


(a) Deformación normal horizontal  $\epsilon_x$

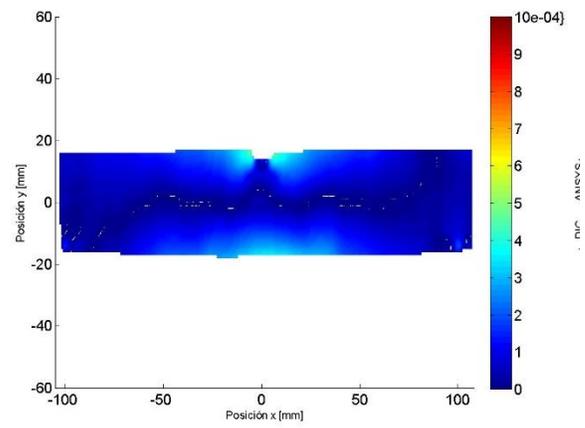
(b) Error

**Figura 5.18: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo de 17x17 datos.**

**Fuente: Elaboración propia a partir de MATLAB R2013b**



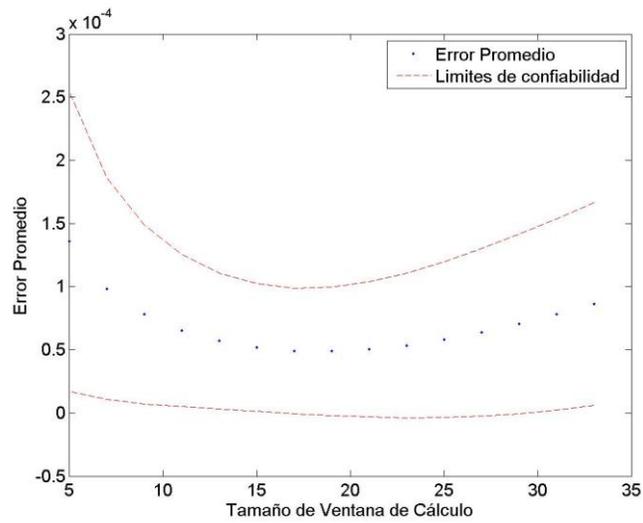
(a) Deformación normal horizontal  $\epsilon_x$



(b) Error

**Figura 5.19: Resultados de la obtención del campo de deformación a través del método PLS, para ventana de cálculo 33x33 datos.**

**Fuente: Elaboración propia a partir de MATLAB R2013b**



**Figura 5.20: Resumen de errores promedios y desviación estándar para cada campo de deformación obtenido a través del método PLS**

**Fuente: Elaboración propia a partir de MATLAB R2013b**

## Análisis

Por una parte, al analizar la Figura 5.17, Figura 5.18 y Figura 5.19, se observan las mismas perturbaciones de las zonas inferior izquierda y centro inferior de la probeta, mencionadas en la sección 5.3.1. Además de estas, en la Figura 5.18 b) y Figura 5.19 b), se aprecia una desviación de los resultados en la zona centro superior de la probeta, del orden de  $3,5 \cdot 10^{-4}$ . Este error, que es más evidente para ventanas de cálculo mayores a  $17 \times 17$  datos, se puede explicar debido a que los vacíos de información, a contiguos a esta zona en zonas centro superior, contienen menor número de datos para realizar la aproximación por ajuste de mínimos cuadrados, a diferencia de los valores más alejados de estas áreas.

Por otra parte, al analizar la Figura 5.20, se aprecia que la aproximación del campo de deformación por el método PLS reduce su error promedio y variabilidad a medida que las ventanas de cálculo son mayores, hasta el tamaño de ventana 17. Luego, a partir de este valor, vuelve aumentar el error promedio, así como los límites de confiabilidad. Lo anterior, se explica en primer lugar, debido a hecho mencionado en el marco teórico, de que ventanas muy pequeñas no logran suprimir el ruido en los desplazamientos experimentales, mientras que para tamaños muy grandes, el supuesto de la aproximación lineal de los desplazamientos deja de ser válido, especialmente en las zonas de deformaciones no homogéneas. En segundo lugar, el error en la zona centro superior de la probeta, para ventanas mayores a 17 datos explica la tendencia ascendente del error promedio y sus límites de confianza. Debido a lo anterior, se escoge como resultado óptimo para este método, la ventana de  $17 \times 17$  datos que resulta en un error promedio de  $4.97 \cdot 10^{-5}$  y desviación estándar de  $4.91 \cdot 10^{-5}$ .

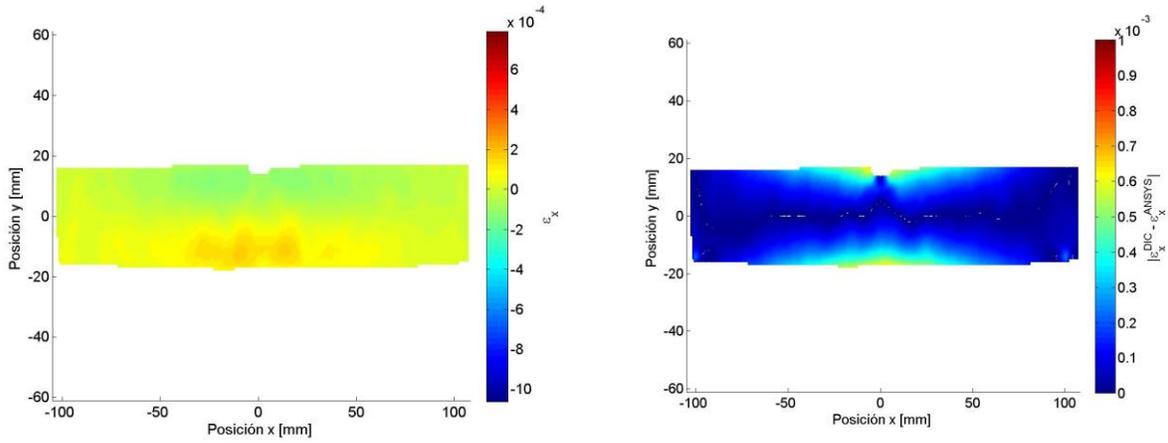
### 5.5 Reducción del Ruido en el Campo de Deformación, Utilizando una Función de Costos Regularizada

Finalmente, se implementó en MATLAB, el método de reducción de ruido para campos deformaciones obtenidos por diferenciación numérica de los desplazamientos no suavizados, por medio de la minimización de una función de costos regularizada, cuyo código se encuentra en la sección de Anexo D.4. Esta función es parametrizada por el valor escalar ' $\alpha$ ', el cual toma valores entre 0 y 1 y define el compromiso entre el suavizado absoluto de la deformación y similitud completa entre el campo original y el filtrado respectivamente.

La Figura 5.21, Figura 5.22 y la Figura 5.23 representan los campos de deformación resultantes y la distribución de error asociada, para los valores del parámetro ' $\alpha$ ' de 0.01, 0.1 y 0.3, respectivamente. En estas figuras, la escala de colores en (a) grafica la deformación normal horizontal  $\epsilon_x$ , mientras que la escala de colores en (b), ilustra la medida del error, representada como el valor absoluto de la diferencia entre el modelo de elementos finitos con la deformación normal horizontal  $\epsilon_x$  obtenida.

Los resultados de la reducción del ruido los campos de deformación, para una serie de valores ' $\alpha$ ' entre 0.0001 y 0.3, se resumen en la Figura 5.24, se muestra el gráfico del error promedio (puntos azules), en función del parámetro ' $\alpha$ ', calculado como el promedio del valor absoluto de la diferencia entre las deformaciones y el modelo en ANSYS. Las líneas rojas representan

el intervalo de confianza de los datos, calculado como la desviación estándar descrita en la ecuación 3.4.

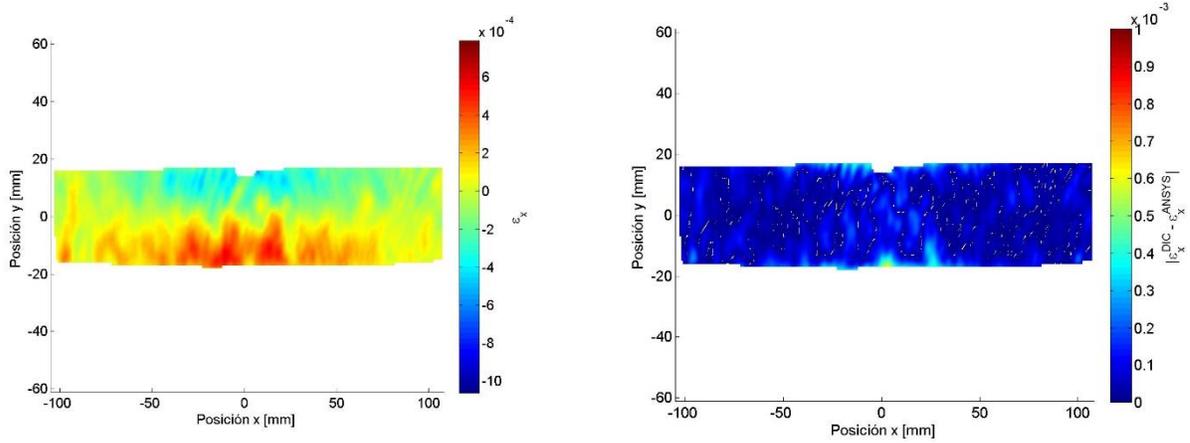


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.21: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro  $\alpha = 0.01$ .**

Fuente: Elaboración propia

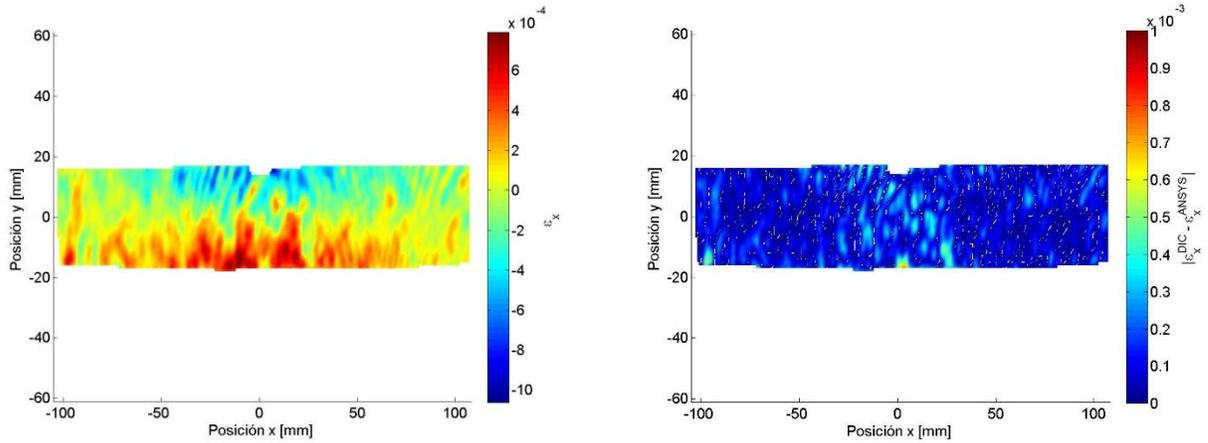


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.22: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro  $\alpha = 0.1$ .**

Fuente: Elaboración propia

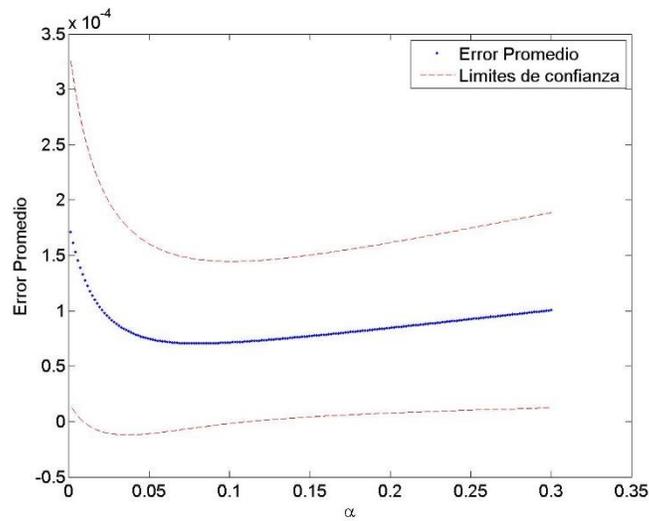


(a) Deformación normal horizontal  $\epsilon_x$

(b) Error

**Figura 5.23: Resultados de la obtención del campo de deformación a través de la minimización de la función de costos regularizada definida en el campo, para el parámetro  $\alpha = 0.3$**

Fuente: Elaboración propia



**Figura 5.24: Resumen de errores promedios y desviación estándar para cada campo de deformación obtenido a través del método de reducción de error, utilizando una función de costos regularizada.**

Fuente: Elaboración propia a partir de MATLAB R2013b

### 5.5.1 Análisis

Si se analizan los campos de deformación y distribución del error obtenidos por este método (representados en la Figura 5.21, Figura 5.22 y Figura 5.23), se observan las mismas zonas de desviación de los resultados, descritas en los análisis de las secciones 5.4.1 y 5.5.1, en las zonas inferior izquierda, centro inferior y centro superior de la probeta. Esta última, se acrecienta para valores en donde el suavizado es mayor (parámetro  $\alpha$  cercano a 0), en donde la ausencia de los datos permite el suavizado de los valores de esta área.

La Figura 5.24 muestra como el error promedio disminuye a medida que el parámetro  $\alpha$  decrece (i.e. más suavizado y menor similitud con la deformación original) hasta el valor mínimo de  $\alpha$ , en donde luego, el error aumenta exponencialmente a medida que se sobre atenúan los valores de la deformación. Este aumento se explica debido a la rápida atenuación y práctica desaparición de la zona de compresión de la zona centro superior de la probeta (ver Figura 5.23 b)), explicados por la ausencia de datos del lugar.

Se identifica el error promedio mínimo del método equivalente a  $7.066 \cdot 10^{-5}$  y desviación estándar de  $7.229 \cdot 10^{-5}$ , que se obtienen para el parámetro  $\alpha = 0.0806$ .

## 5.6 Resumen de Resultados

En lo referente al montaje experimental en el documento Guía de Consideraciones Experimentales, adjuntado en el Anexo A (que detalla el proceso de medición, dificultades y consideraciones del proceso del el método DIC), se observa que la principal dificultad, es lograr el reconocimiento completo de la región de interés por parte del software DIC, Istra 4D. Las variables que tienen mayor influencia en lo anterior son: las condiciones de iluminación; la calidad y nitidez del patrón; el enfoque de la región de interés y los ángulos de posicionamiento de las cámaras. Por otra parte, si no se corrigen los errores de distorsión y origen del sistema de coordenadas, se introducen errores adicionales que pueden ser evitados si se consideran las transformaciones de ángulo y traslación.

En la medición de desplazamientos del patrón tipo “cepillo” (Figura 5.4), se observan claros signos del efecto Aliasing (Figura 5.4). Este efecto se exagera con la aplicación de la técnica de salpicado de tinta spray con cepillo, debido a que genera manchas muy finas. Es por esta razón, que debe hacerse con cuidado, puesto que de lo contrario, aparecen manchas muy grandes que hacen que los sub sets de referencias encierren zonas completamente negras o blancas, provocando errores y desviaciones grandes de los resultados

En cuanto a la solución del problema de flexión, utilizando elementos finitos, se observa que presenta concentraciones de deformación en las zonas de los apoyos y aplicación de la carga. Este efecto, no genera grandes desviaciones de los resultados en la distribución de desplazamientos (Figura 5.6 y la Figura 5.7). Sin embargo, este efecto genera una distorsión en la distribución de los valores de deformación en estas zonas, no obstante, la región de interés considerada en la correlación de imágenes digitales, no contiene estas áreas. Todo lo anterior, sumado a la alta calidad de los elementos del modelo de malla, implica que el modelo funciona como una solución de base

apropiada para la comparación de las medidas del desplazamiento obtenidas por el algoritmo DIC y los campos de deformación obtenidos por los métodos propuestos.

Los resultados de los desplazamientos obtenidos para la probeta tipo “*timbre*”, sub set de referencia de 17x17 pixeles y distancia de paso 5 pixeles presentan el menor error promedio  $7.55 \cdot 10^{-4}$  mm y desviación estándar de  $6.25 \cdot 10^{-4}$ mm. Al comparar lo anterior, con la escala de los desplazamientos máximos medidos (del orden de  $4 \cdot 10^{-2}$ ), se tiene un error de dos órdenes de magnitud menor.

La Figura 5.25 resume los resultados obtenidos los tres métodos propuestos de reducción del ruido y errores en el campo de deformación. El eje x de la figura, indica el método en MATLAB utilizado para la obtención del campo de deformación, mientras que en el eje y se grafica el error promedio y los límites de confiabilidad, definidos por la desviación estándar (ecuación 3.4).

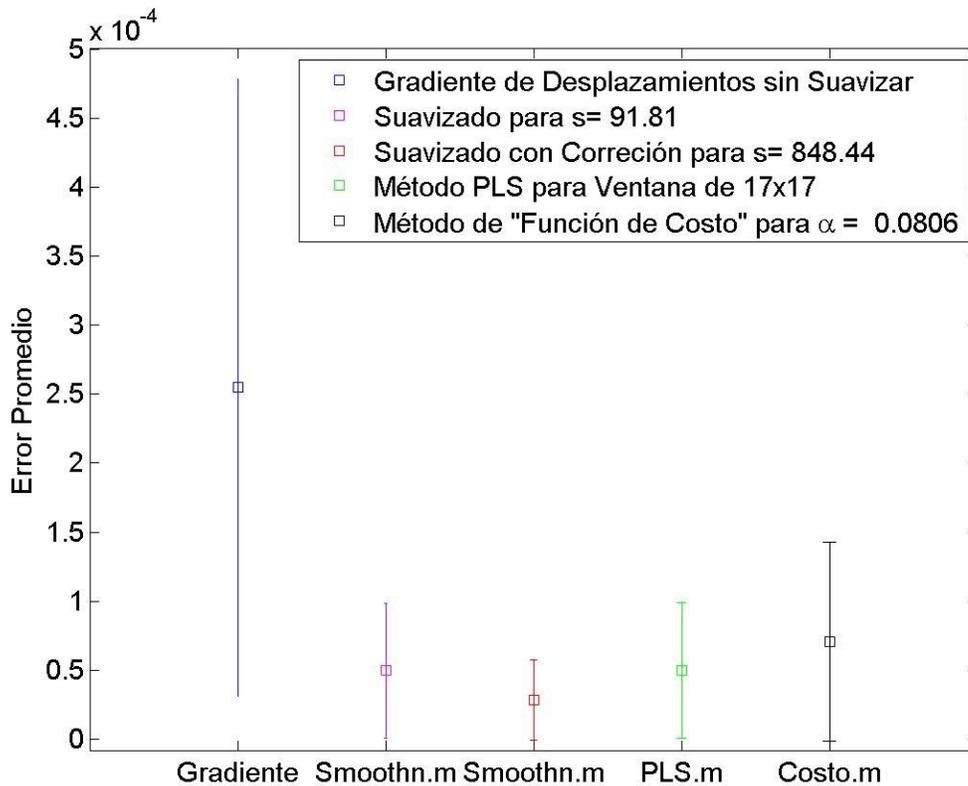


Figura 5.25: Resumen de resultados del error promedio y desviación estándar para los tres métodos propuestos de reducción del ruido y errores en el campo de deformación.

Fuente: Elaboración propia a partir de MATLAB R2013b

Se observa que el método de reducción del error más efectivo, corresponde al suavizado de los desplazamientos mediante la aproximación de estos utilizando la transformada de coseno discreta (método “*Smoothn.m*” en MATLAB) y posterior aplicación del gradiente para la obtención del campo de deformación. Para este método, se consideraron los casos en que se calcula el error para la región de interés completa y para el caso en que se corrige el error del método de suavizado en las esquinas de la probeta, considerando solo los valores de la probeta entre  $x=-95$ mm y

$x=95\text{mm}$ . Que sea el valor mínimo se explica, debido a que es el único método de los tres que no atenúa completamente la zona de compresión de la parte superior de la probeta.

Finalmente la Tabla 5.2 resume para cada método estudiado los porcentajes de reducción del error absoluto promedio, con respecto al valor del error del campo de deformación obtenido por aplicación directa del gradiente.

**Tabla 5.2. Resumen del porcentaje de reducción de error, para los métodos de obtención de campos de deformación suavizados.**

	<b>Error Promedio <math>\overline{\mu_\epsilon}</math></b>	<b>Desviación Estándar <math>s_\epsilon</math></b>
Campo de Deformación Original	$2.549 \cdot 10^{-4}$	$2.235 \cdot 10^{-4}$
<b>Métodos de Reducción de Error</b>	<b>Reducción del Error[%]</b>	<b>Reducción de la Desviación Estándar[%]</b>
Suavizado con método "Smoothn.m"	80,53	78,23
Suavizado con método "Smoothn.m" con corrección	88,85	86,95
Método PLS	80,50	78,02
Minimización de función de costos regularizada	72,28	67,76

**Fuente: Elaboración propia**

## 6 Conclusiones

Se proponen dos técnicas para la preparación de las probetas para la correlación de imágenes digitales. La primera, por medio del salpicado de tinta con un cepillo y tinta en spray negra. Esta técnica, produce manchas pequeñas y variables entre 0.9mm a 3.1mm. Para esta probeta, los resultados de los desplazamientos obtenidos a través del método DIC no fueron óptimos, debido a que presentaron efectos de Aliasing para todos los tamaños de sub set de referencia que se consideraron. Lo anterior se explica debido a que para las condiciones definidas en el montaje experimental (en particular, la distancia entre las cámaras y la probeta), la resolución de las imágenes no logra capturar con precisión manchas muy pequeñas, generando desviaciones en los valores de la escala de grises al interior del sub set de referencia, con el que se realiza la correlación.

La segunda técnica desarrollada, corresponde al estampado del patrón de manchas con tinta gráfica por medio de un timbre fabricado por cortado laser, a partir de un patrón digital optimizado. Esta técnica produce manchas de tamaño entre 3.1mm a 3.3mm e implicó los mejores resultados en la obtención de desplazamientos. De estos resultados, el sub set de 17x17 y distancia de paso 5 pixeles presentan el menor error promedio de  $7.55 \cdot 10^{-4}$ mm y desviación estándar de  $6.25 \cdot 10^{-4}$ mm, con los que se tiene un error de dos órdenes de magnitud menor, con respecto a la escala de los desplazamientos máximos medidos (de  $4 \cdot 10^{-2}$  mm). Sin embargo, la obtención de los desplazamientos finales no fue directa, ya que se iteró sobre las condiciones del montaje experimental hasta encontrar la configuración misma que produjera el reconocimiento completo de la región de interés por parte del software Istra 4D y el cálculo de desplazamientos con menor distorsión. Debido a la falta de información clara, la dispersión de esta misma, sumado al hecho de que no existen parámetros óptimos definidos para obtener una medición adecuada, se elaboró el documento “Consideraciones del Montaje Experimental” (adjunto en la sección Anexo A). Este último, establece recomendaciones para el montaje experimental y una metodología clara de cómo proceder para obtener mediciones apropiadas. En este, la principal dificultad observada, fue la definición del montaje experimental para lograr el reconocimiento completo de la región de interés por parte del software DIC. Las variables que tienen mayor influencia para lograr lo anterior son: El proceso de calibración, la calidad y nitidez del patrón, el enfoque apropiado de la región de interés, y los ángulos de posicionamiento de las cámaras y las condiciones de iluminación. De esta manera, el documento sirve como herramienta para futuros proyectos en los que se requiera la obtención de desplazamientos, por medio de correlación de imágenes digitales.

Tomando en cuenta las apreciaciones y distintos resultados obtenidos a lo largo del proceso de iteración sobre las condiciones de montaje experimental, se recomienda fabricar timbres con distintos tamaños de manchas para facilitar la realización de mediciones en escalas de tamaño diferentes a la considerada en este trabajo. Además, considerando que una calibración óptima de las cámaras implica mejores resultados en la medición de desplazamientos, se sugiere fabricar un armazón que permita sostener la placa de calibración de las cámaras, el cual posibilite giro y movimiento de esta en las posiciones requeridas para la calibración. De esta manera, se excluye el error producido por la posible inestabilidad de la persona que sostenga la placa de calibración.

En cuanto a los campos de deformación obtenidos, el método de aplicación del gradiente en los desplazamientos suavizados por medio de la aproximación sobre mínimos cuadrados pena-

lizados, utilizando la transformada de coseno discreta, atenúa las alteraciones de los desplazamientos experimentales. Lo anterior, se hace obvio al observar el efecto del aumento de los factores de suavizado sobre las líneas de contorno de los desplazamientos obtenidos, esta situación explica la disminución efectiva del error en el campo de deformación calculado. Sin embargo, el método presenta problemas en el suavizado de los bordes de la probeta, debido a que para la obtención de estos valores, el método interpola los puntos vecinos al punto de interés (valores no existentes). No obstante, esta interpolación, produjo un efecto positivo al intercalar puntos de interés dentro de la región de la probeta, los cuales no habían sido identificados por el algoritmo DIC. De esta forma, la zona de compresión del área centro superior de la probeta, no desaparece para factores de atenuación altos. Esto implica que los campos de deformación obtenidos por este método no presenten errores en la zona superior de la barra, a diferencia de los otros dos métodos de estudio. Si se omite el efecto de este error, tomando en cuenta únicamente los datos de la probeta entre -95mm y 95mm, se tiene que la reducción del error promedio del método alcanza un 88,85% y un 86,95% para la desviación estándar. Si se consideran todos los datos, la reducción corresponde al 80,53% del error promedio y en un 78,23% de la variabilidad, con respecto al campo de deformación obtenido sin suavizado previo de los desplazamientos

Por su parte, el método PLS reduce efectivamente el error promedio de las deformaciones, para valores de ventana de cálculo entre 11x11 y 23x23 datos. Para una ventana de 17x17, se alcanza un máximo de reducción del error de 80,5% y 78,02% en la desviación estándar. Este método, no considera regiones que tengan vacíos de datos (bordes y datos inválidos). Debido a lo anterior, el método produce un sobre suavizado de los valores en el borde centro superior de la probeta donde ocurre compresión. Esta última situación, se explica debido a que los valores de esa zona (que en este caso son nulos) debiesen contribuir al cálculo del valor de los puntos que finalmente se sobre atenúan.

A su vez, la reducción del ruido en el campo de deformación, utilizando una función de costos regularizada, funciona de manera óptima para valores de  $\alpha$  entre 0.05 y 0.1. Para el valor de  $\alpha = 0.0806$ , se alcanza la máxima reducción de los errores experimentales de un 72,28% y de un 67,76%. Para valores inferiores de  $\alpha = 0.08$ , el error aumenta exponencialmente a medida que se sobre atenúan los valores de la deformación. Este aumento, se debe a la rápida atenuación y práctica desaparición de la zona de compresión de la zona centro superior de la probeta, efecto análogo al explicado anteriormente para el método PLS, pues este método no toma en consideración para los cálculos los valores nulos.

Se debe tener cuidado en regiones de interés en las que hay ausencia de datos, debido a que los métodos de cálculo de deformación que se valen de la información de los puntos vecinos al punto de interés, pueden provocar el sobre suavizado de los datos no se utiliza interpolación para estos puntos vacíos (caso PLS y función de costo). Por el contrario, estos métodos de interpolación pueden dar valores erróneos en los datos, generando alteraciones en los datos al momento de aplicar el gradiente (método de suavizado por aproximación de mínimos cuadrados y transformada de coseno discreta).

Para finalizar, considerando que las deformaciones estudiadas son deformaciones pequeñas cercanas al orden de  $10^{-3}$ , un montaje experimental adecuado junto a la aplicación de los mé-

todos estudiados, logran reducir de manera óptima el efecto del ruido experimental y errores numéricos en el campo de deformación, alcanzando órdenes de  $10^{-5}$ . Además, debido a la buena reducción del error del método de suavizado y posterior aplicación del gradiente, se sugiere la evaluación de la aplicación otros algoritmos de suavizado de desplazamientos, tales como funciones de base radiales[3] y métodos de suavizado por elementos finitos [13] [14], por mencionar algunos.

## 7 Bibliografía

- [1] B. Pan, J. Yuan, and Y. Xia, “Strain field denoising for digital image correlation using a regularized cost-function,” *Opt. Lasers Eng.*, vol. 65, pp. 9–17, 2014.
- [2] B. Pan, A. Asundi, H. Xie, and J. Gao, “Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements,” *Opt. Lasers Eng.*, vol. 47, no. 7–8, pp. 865–874, 2009.
- [3] X. Dai, F. Yang, Z. Chen, X. Shao, and X. He, “Strain field estimation based on digital image correlation and radial basis function,” *Opt. Lasers Eng.*, vol. 65, pp. 64–72, 2015.
- [4] P. Lava, S. Cooreman, S. Coppieters, M. De Strycker, and D. Debruyne, “Assessment of measuring errors in DIC using deformation fields generated by plastic FEA,” *Opt. Lasers Eng.*, vol. 47, no. 7–8, pp. 747–753, 2009.
- [5] J. Zhao, Y. Song, and X. Wu, “Fast Hermite element method for smoothing and differentiating noisy displacement field in digital image correlation,” *Opt. Lasers Eng.*, vol. 68, pp. 25–34, 2015.
- [6] Correlated Solutions, “Digital Image Correlation : Overview of Principles and Software 2D Image Correlation Fundamentals,” *South Carolina Univ.*, 2009.
- [7] P. K. Thomas Becker, Karsten Splitthof, Thorsten Siebert, “Error Estimations of 3D Digital Image Correlation Measurements Introduction,” in *Proc. SPIE 6341, Speckle06: Speckles, From Grains to Flowers*, 2006, pp. 1–6.
- [8] Correlated Solutions, “Bias and Noise in DIC - Part 1 Identifying and Addressing bias,” 2011.
- [9] R. Bustamante, *Apunte de Resistencia de Materiales*. Santiago: Universidad de Chile, 2012.
- [10] M. H. Sadd, *Elasticity Theory, Applications, and Numerics*. Oxford: -Elsevier, 2005.
- [11] Correlated Solutions, “Bias and Noise in DIC - Part 2 Quantifying & Minimizing Noise,” 2011.
- [12] Correlated Solutions, “Vic-2D Testing Guide,” 2009.
- [13] P. Lava, S. Cooreman, and D. Debruyne, “Study of systematic errors in strain fields obtained via DIC using heterogeneous deformation generated by plastic FEA,” *Opt. Lasers Eng.*, vol. 48, no. 4, pp. 457–468, 2010.
- [14] L. B. Meng, G. C. Jin, and X. F. Yao, “Application of iteration and finite element smoothing technique for displacement and strain measurement of digital speckle correlation,” *Opt. Lasers Eng.*, vol. 45, no. 1, pp. 57–63, 2007.
- [15] B. Wattrisse, a Chrysochoos, J. M. Muracciole, and M. Nemoz-Gaillard, “Analysis of strain localization during tensile tests by digital image correlation,” *Exp. Mech.*, vol. 41, no. 1, pp. 29–39, 2001.

- [16] D. Garcia, “Robust smoothing of gridded data in one and higher dimensions with missing values,” *Comput. Stat. Data Anal.*, vol. 54, no. 4, pp. 1167–1178, 2010.

## ANEXOS

Anexo A: Consideraciones del Montaje Experimental

**CONSIDERACIONES PARA LA CALIBRACIÓN DEL EQUIPO DAN-  
TEC DYNAMICS Y LA UTILIZACIÓN DEL SOFTWARE ISTR4 4D  
PARA LA MEDICIÓN DE DESPLAZAMIENTOS**

VERSIÓN 1.0

EDGAR LEONARDO GEERLING GAMBOA

SANTIAGO, CHILE  
1 DE DICIEMBRE, DE 2015

## Contenido

1. Consejos del Patrón de Manchas.....	3
1.1. Técnicas de impresión del patrón.....	4
1.2. Trabajo futuro.....	6
2. Proceso de Calibración.....	6
2.1. Consideraciones previas a la calibración .....	9
2.1.1. Apertura de Cámara .....	9
2.1.2. Exposición.....	9
2.1.3. Enfoque de la muestra.....	9
2.1.4. Angulo entre cámaras .....	11
2.1.5. Resumen de consideraciones .....	11
2.2. Errores de calibración no adecuada.....	11
2.2.1. Error de distorsión de perspectiva.....	11
2.2.2. Aliasing.....	12
2.3. Calibración con Istra 4D .....	15
2.3.1. Iniciando la calibración.....	15
2.3.2. Toma de fotografías de calibración.....	17
2.4. Trabajo Futuro.....	19
3. Medición de desplazamientos .....	20
3.1. Definir Sistema de Referencias .....	25
4. Exportar datos .....	29
5. Referencias .....	29

## 1. Introducción

Este documento se crea como complemento del trabajo de título ME6909 “*REDUCCION EL ERROR EXPERIMENTAL EN LOS CAMPOS DE DEFORMACIÓN OBTENIDOS POR UN SISTEMA DE CORRELACIÓN DE IMÁGENES DIGITALES*”. En este se realizan recomendaciones para el montaje experimental y se defina una metodología clara de cómo proceder para obtener mediciones de desplazamientos utilizando la correlación de imágenes digitales, a través del equipo de Dantec Dynamics y el software Istra4D.

La idea de este documento, de características informales y explicaciones informales es que sirva como herramienta y guía para futuros proyectos en los que se requiera la obtención de desplazamientos, por medio de correlación de imágenes digitales.

## 2. Consejos del Patrón de Manchas

En primer lugar, antes de realizar cualquier medición, es necesario imprimir un patrón de manchas sobre la muestra que se desea analizar. El patrón de manchas debe ser: no repetitivo, isotrópico y de alto contraste [1]. La Figura 2.3 representa las características menos adecuadas de un patrón de manchas para los casos a), b) y c), mientras que el caso d) ilustra el caso ideal.

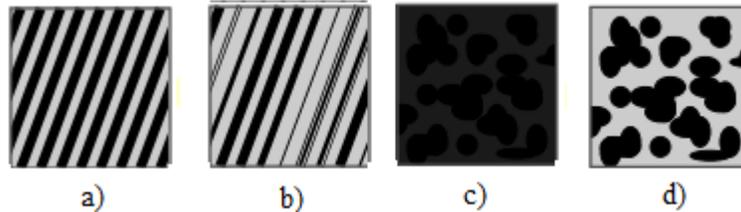


Figura 2.1 Ejemplos de patrones de tipo: a) Repetitivos; b) Anisotrópicos; c) Bajo contraste; d) Ideal.  
Fuente: Imagen adaptada de [2]

Los patrones de manchas no deben ser ni muy grandes ni muy pequeño. Si el patrón es muy grande, puede ocurrir que algunos sub sets pueden encerrar una región completamente negra o completamente blanca. Lo anterior, no permite hacer un rastreo adecuado del sub set ya que se tendrían coincidencias de estas regiones en muchas otras partes de la imagen. Se puede compensar este efecto, con la definición sub set más amplio y así asegurar mayor confiabilidad en los resultados, pero este no será capaz de describir deformaciones heterogéneas grandes con precisión. Además, por una parte, una región de sub set de pixeles más pequeña es más precisa, pero su resultado estará dominado por la granularidad<sup>6</sup> del patrón de manchas y puede llevar a errores sistemáticos [3].

### 1.1. Técnicas de impresión del patrón

Hasta la fecha, existen diversas técnicas para imprimir el patrón de manchas sobre una muestra. Entre las que se encuentran[2]:

- Polvo de tóner para manchas muy pequeñas
- Pintura en spray
- Técnica del cepillo de dientes
- Grabado electroquímico
- Impresión de pantalla
- El adhesivo de vinilo con respaldo (por ejemplo, 3M Controltac)
- Plantillas (Stencils)
- etc...

De estas técnicas, tres fueron analizadas y aplicadas en probetas de acero SAE 1020, estas son descritas a continuación:

---

<sup>6</sup> Término asociado al nivel de detalle de un modelo. Cuanto mayor es la granularidad de un modelo, más profundo es el nivel de detalle. A su vez, este concepto se utiliza generalmente para caracterizar la escala o nivel de detalle en un conjunto de datos.

- **Pintura Negra en Spray:** La técnica más utilizada en los estudios con cámaras DIC corresponde a pintar las muestras con pintura en lata spray. Esta consiste en presionar de manera sutil la boquilla del spray, haciendo un movimiento de barrido por sobre la muestra, a modo de dispersar de manera homogénea la pintura por la superficie. Esto generará manchas que en general cumplen con las 3 cualidades descritas anteriormente, con la desventaja de que las manchas serán de tamaño relativamente variable y la técnica dependerá mucho de la habilidad de quien aplique la técnica.
- **Salpicado de Pintura con Cepillo:** Como otra alternativa para conseguir un patrón de manchas adecuado se utilizó un cepillo dental común y tinta negra en spray. La pintura es aplicada en la cabeza del cepillo, desde la cual utilizando un dedo para desplazar hacia atrás las cerdas del cepillo, se salpica la probeta con la tinta negra como se muestra en la Figura 4.1. Se repite el proceso hasta que la región de interés quede aproximadamente con un 50% de cobertura con pintura negra, creando una distribución homogénea de las manchas (evitando concentraciones de manchas y manchas de tamaño desigual). El método es de fácil aplicación, pues no requiere mayor técnica y el tamaño de las manchas varía entre los 1,5 mm a 3,1 mm, siendo su distribución relativamente homogénea.



Figura 2.2: Técnica de aplicación del patrón de manchas con cepillo y tinta en spray.  
Fuente: Elaboración propia

- **Estampado del Patrón de Manchas con Timbre:** Para normalizar, sistematizar la preparación de las muestras y disminuir la posibilidad de agregar errores sistemáticos en las mediciones, se fabricó un timbre de goma de 40 mm x 80 mm que permite estampar el patrón de manchas por contacto. El patrón del timbre fue elaborado por cortado laser, a partir de un patrón de manchas digital recomendado por el proveedor del sistema DIC. Debido a que la superficie de la probeta metálica no es absorbente, el uso de tinta común para estampar no es posible, pues tiene tendencia a escurrirse y el resultado es insatisfactorio. Para sortear esta condición, se utiliza un rodillo de impresión, el cual se impregna con **tinta gráfica de off set negra**, para luego transferir de manera uniforme una capa de

pintura al timbre de goma, para finalmente estampar el patrón sobre la probeta. El tamaño de las manchas obtenido, varía entre los 3,1 mm a 3,3 mm de ancho. Los elementos anteriormente descritos se muestran en la Figura 4.2 a), mientras que el resultado del método se muestra en b).

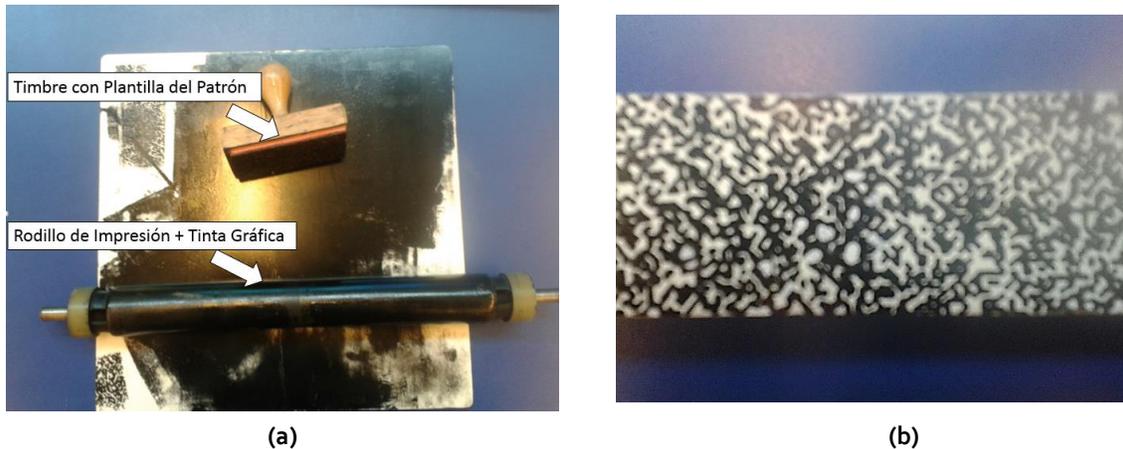


Figura 2.3: Método de Estampado del Patrón de Manchas con Timbre. En a) Elementos necesarios y (b) Resultado  
Fuente: Elaboración propia

Las probetas fueron pintadas con pintura blanca, opaca y en spray como base. De estas técnicas, la más aceptable es la que utiliza un timbre para el estampado de las manchas. Si la muestra es de un material poroso, o absorbente, la utilización de tinta gráfica puede ser reemplazada por tinta negra común.

## 1.2. Trabajo futuro

Lo ideal sería que se fabricaran timbres de tamaños de manchas distintos, para diversas aplicaciones y escalas de medición. De esta forma, se contaría con un set de timbres, que faciliten y hagan la impresión del patrón de manchas una tarea sencilla y modular.

## 2. Proceso de Calibración

El procedimiento de calibración corresponde a la determinación de los parámetros de imagen de cada una de las cámaras (parámetros intrínsecos), así como las posiciones externas y orientaciones de las cámaras con respecto a un sistema de coordenadas global. Se requiere la calibración del sistema, para la transformación de las posiciones de los puntos de la superficie de una muestra, representados por las imágenes de dos cámaras, a las coordenadas 3D correspondientes de ese punto. Los errores de calibración son una importante fuente potencial de errores sistemáticos, que limitan la resolución de los resultados [4], por lo que una calibración no adecuada puede tener como consecuencia la distorsión de las imagen adquiridas [5], [6].

El primer paso para el proceso de calibración es montar las cámaras sobre un soporte rígido y en lo posible amarrar y ordenar los cables de las cámaras. Posteriormente se debe fijar la distancia entre la muestra y las cámaras, la distancia entre ambas cámaras, así como el ángulo entre ellas (ángulo estéreo). Se recomienda que para lentes cortos, el ángulo entre las cámaras debe ser superior a  $25^\circ$  y que el área de interés a analizar se ubique cercano al centro de la imagen [6]. Es importante que ninguna de las cámaras se muevan relativas a las otras, esto se traduce en que las cámaras solo pueden girar simultáneamente en los ángulos los ejes X e Y referenciales (que se muestran en la Figura 2.1), pero en ningún caso distanciarse entre sí, o cambiar variar su ángulo de estereo " $\alpha$ " (ver Figura 2.2). Si no se cumple lo anterior, las cámaras tendrán que ser calibradas nuevamente.

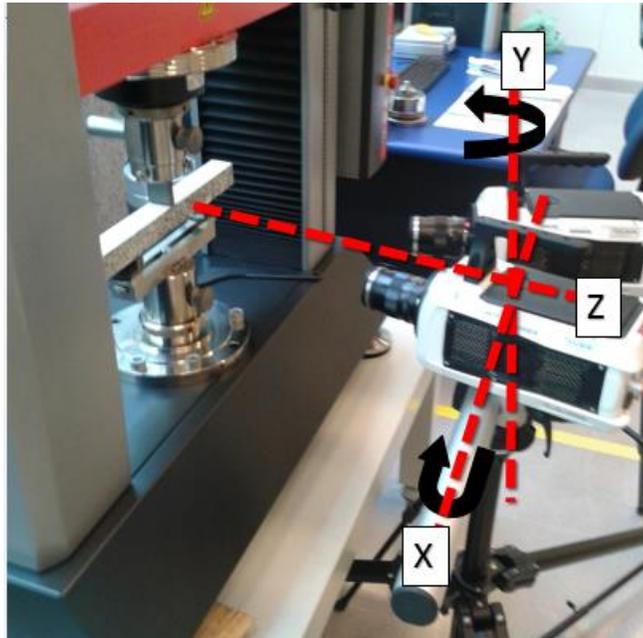


Figura 2.1 Representación de los ángulos de giro de las cámaras en los ejes referenciales X, Y, Z.  
Fuente: Elaboración propia

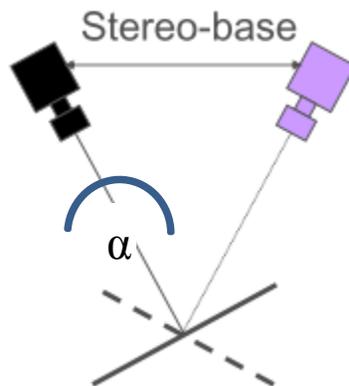


Figura 2.2 Representación del ángulo de estereo y distancia entre cámaras  
Fuente: Adaptado de[2]



## 2.1. Consideraciones previas a la calibración

Es necesario entender algunos conceptos antes de realizar una calibración apropiada, los cuales son descritos a continuación.

### 2.1.1. Apertura de Cámara

Este concepto tiene relación con la cantidad de luz que una cámara utiliza para captar una imagen. La apertura se mide con el número  $-f$ , el cual en general varía entre 1.4 a 22, en donde para números más chicos se tiene una mayor apertura, y para números de  $-f$  mayores se tiene pequeñas aperturas de cámara. Este número generalmente aparece en las cámaras en la base de los lentes (zona de unión entre el lente y las cámaras) como se puede ver en la siguiente figura. Usar aperturas más grandes, obviamente implicará tener imágenes más brillantes, sin embargo (como se explicará en detalle en la sección 2.1.3.) la profundidad de campo de la imagen se acortará.

### 2.1.2. Exposición

Así como la apertura es la cantidad física de cuanta luz la cámara admite para una foto, la exposición corresponde al tiempo a la que el sensor está expuesto a la luz, al tomar un fotograma. Tiempos de exposición altos harán la imagen más brillante, pero pueden significar que la imagen tenga zonas borrosas, si durante la toma de la fotografía ocurren movimientos significantes. Debido a esto se debe tener en cuenta la rapidez con la que se deforma la muestra en estudio, de manera que el tiempo de exposición suficiente para captar la deformación de la muestra.

Como método de estimación del tiempo de exposición, se recomienda que la velocidad de la parte que se deforma más rápido no debe deformarse más de 0.01 píxeles por sobre el tiempo de exposición

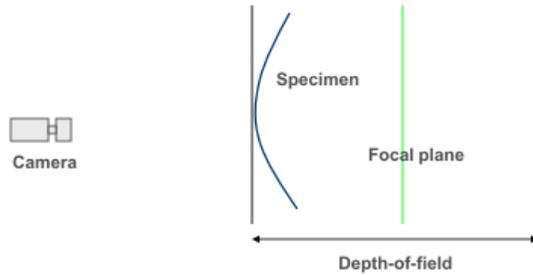
### 2.1.3. Enfoque de la muestra

Teniendo en cuenta los conceptos explicados anteriormente, se puede proceder al enfoque de la muestra. Es importante que la muestra completa esté enfocada correctamente por las cámaras durante todo el ensayo. Sin embargo, puede ser difícil enfocar las cámaras cuando la profundidad de campo<sup>7</sup> es grande. Como se muestra en la siguiente figura

---

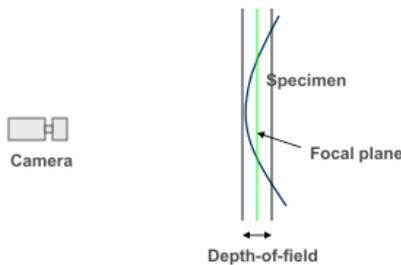
<sup>7</sup> Este concepto corresponde al margen de distancia que aparece nítida en una imagen

### Small aperture (high f-number):

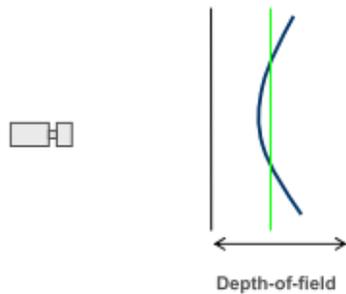


Si se abre la apertura de los lentes de las cámaras (representados por bajos números “-f”), provocará que la profundidad de campo sea muy corta, como se ve en la figura:

### Large aperture (small f-number):



Este efecto causará que todas las zonas de la muestra que estén fuera del plano focal de desenfocuen drásticamente. De esta forma, se puede regular de mejor forma el enfoque de tal manera en que toda la muestra quede enfocada. Posterior al enfoque de la muestra con apertura de lente amplio, se puede volver a cerrar la apertura, causando el efecto retratado en la siguiente imagen:



Para el experimento realizado en el trabajo de título, se realizó este procedimiento, utilizando una apertura de lente equivalente a  $f 18$ . Lo anterior, debido a que la superficie que estaba siendo analizada era plana. Para aplicaciones en que se tenga un espécimen no plano (como un cilindro por ejemplo, o la situación reflejada en las figuras anteriores), se recomienda definir con precisión la profundidad de campo.

#### **2.1.4. Angulo entre cámaras**

Se recomienda que para lentes cortos, el ángulo entre las cámaras debe ser superior a  $25^\circ$  y que el área de interés a analizar se ubique cercano al centro de la imagen [6]. Como consecuencia de lo anterior y sumado a la cercanía de las cámaras con las muestras, se produce que el enfoque de la imagen completa sea complejo, debido a que el ángulo de las cámaras incide en que el plano de enfoque no sea el mismo a lo largo de la barra, generando una zona enfocada y otra parcialmente desenfocada.

#### **2.1.5. Resumen de consideraciones**

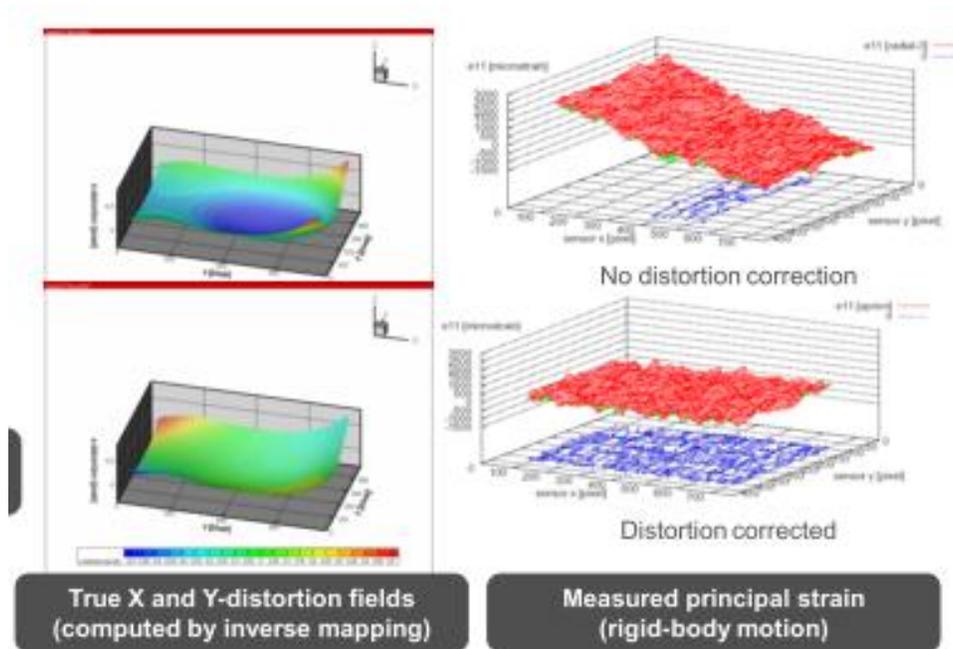
A continuación se hace una lista de las consideraciones a tener en cuenta al realizar el enfoque y calibración:

- Generalmente el proceso de calibración es el cuello de botella para lograr desplazamientos
- Usar tiempos de exposición suficientemente cortos, de manera que no se generen zonas borrosas producto del movimiento(deformación) de la muestra
- Si se tienen muestras planas, la influencia de la profundidad de campo grandes es mínima
- Las aperturas medias (entre  $-f 12$  y  $-f 16$ ) en general son las más recomendadas.
- Para la definición de un sistema de coordenadas adecuado, es necesario marcar el origen del sistema, además de otro punto de referencia, con los cuales se define la dirección del eje x.

### **2.2. Errores de calibración no adecuada**

#### **2.2.1. Error de distorsión de perspectiva.**

Se pueden generar distorsiones, debido a la calibración no adecuada del sistema DIC, ya sea por una calibración deficiente o porque el sistema de coordenadas no fue definido apropiadamente. La siguiente imagen, es un ejemplo de un error de distorsión de persepectiva, para una placa plana, sometida a deformación. Las imágenes superiores muestran la placa sin correcciones geométricas, mientras que las inferiores muestran los resultados al corregir este efecto.



Estos errores ocurrieron en la obtención de los desplazamientos de la superficie de la probeta sometida a flexión. Estos se corrigieron, rotando los datos en los planos YZ y XZ, de manera tal de dejar todos los valores de las coordenadas Z de los puntos de interés en cero. Este efecto puede ser fácilmente corregido utilizando rotaciones de los ejes de coordenadas como se muestra en el siguiente script.

```
%Girar sistema de coordenadas en torno a Y
catop = abs(cx(1,1)-cx(1,aux2));
catady = abs(cz(1,1)-cz(1,aux2));
beta1= (pi/2) -atan(catop/catady);
CX=coord_x.*cos(beta1)+coord_z.*sin(beta1);
CZ=-coord_x.*sin(beta1)+coord_z.*cos(beta1);
%Girar sistema de coordenadas en torno a X
catady = abs(cy(aux1,1)-cy(1,1));
catop = abs(cz(aux1,1)-cz(1,1));
beta=- atan(catop/catady);
CY=coord_y.*cos(beta)-CZ.*sin(beta);
CZ=-coord_y.*sin(beta)+CZ.*cos(beta);
```

En donde se toman los valores de los bordes de la figura y se calcula el ángulo entre este y los ejes de coordenadas, de manera de dejar todos los datos con  $z = 0$ .

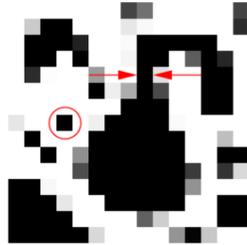
### 2.2.2. Aliasing

Aliasing es el fenómeno que ocurre cuando una señal (en este caso, una imagen) contiene componentes de altas frecuencias, las cuales no pueden ser representadas por la frecuencia de muestreo, (en este caso, la resolución en pixeles de la imagen). Si el tamaño de manchas es muy pequeño, la

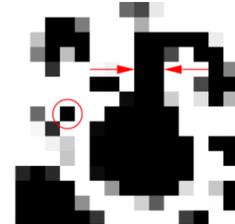
resolución de la cámara puede no ser suficiente para representar la muestra con precisión, ocurriendo el efecto de Aliasing. Este efecto significa que, a medida que se hace la correlación, el sub set de referencia contendrá alteraciones oscilantes, producto de las pequeñas manchas y la incapacidad de las cámaras para captarlas en detalle, debido a una baja resolución de imagen (ver Figura 2.4). Este efecto, producirá una desviación del valor en la escala de grises para la correlación, por lo que el rastreo de los sub set no será preciso.



(a) Patrón original

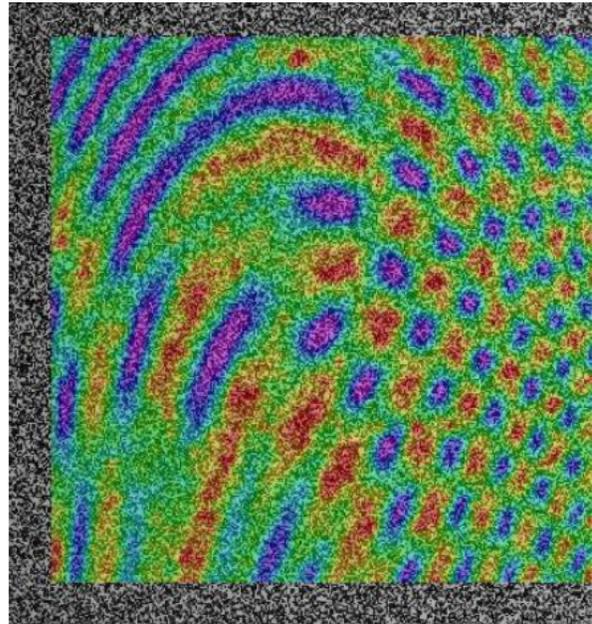


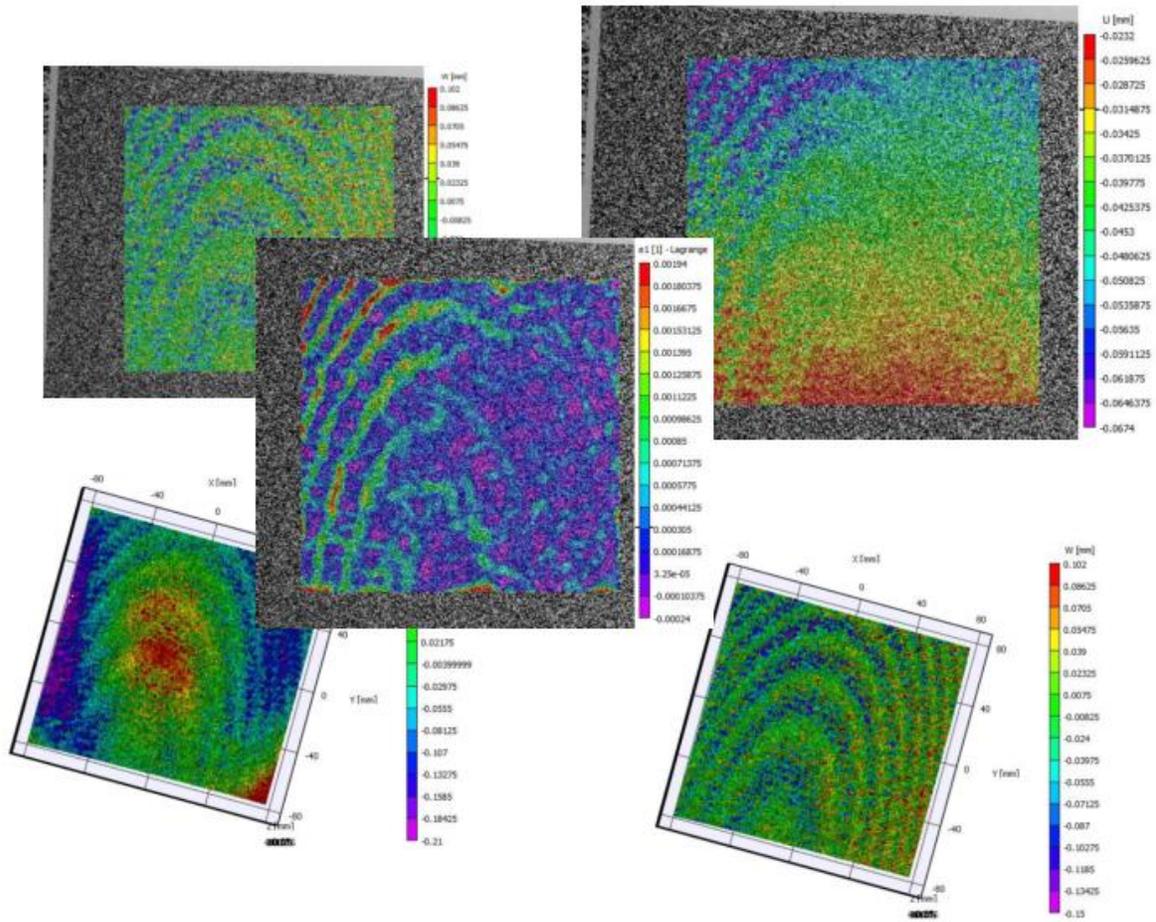
(b) Imagen adquirida



(c) Imagen deformada adquirida

El Aliasing, generalmente se expresa en los resultados como patrones del tipo *moiré* [1], como se muestra en las siguientes Figuras.





## 2.3. Calibración con Istra 4D

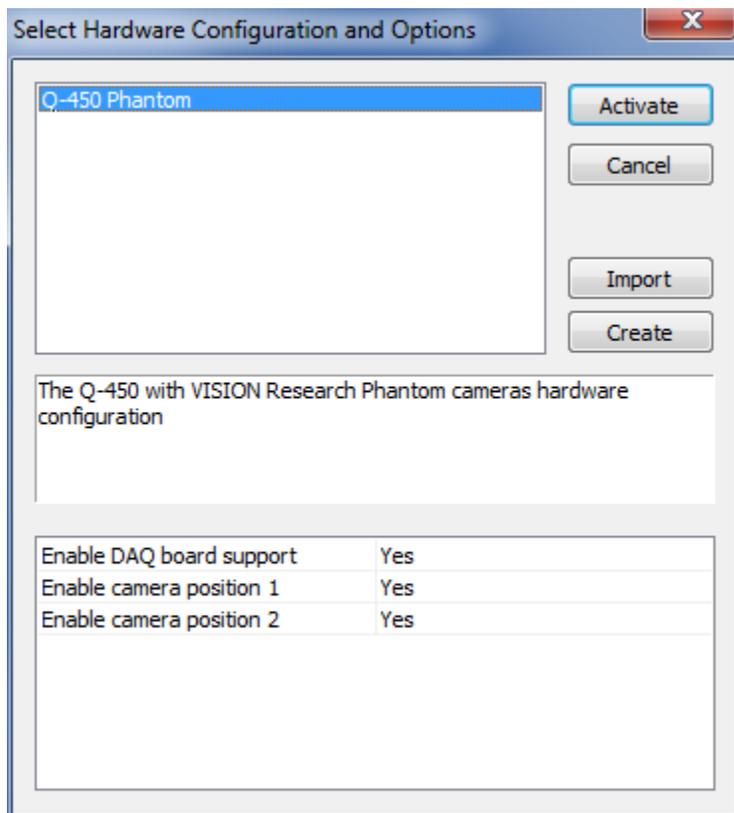
### 2.3.1. Iniciando la calibración

Antes de medir o calibrar, es necesario activar el hardware de las cámaras con el botón “Hardware Activate”, descrito en la siguiente figura:



**Figura 2.3 Botón activar Hardware**  
Fuente: Adaptado del Manual Istra 4D- Q400

Al hacer click, se abrirá un cuadro de dialogo como el de la siguiente figura. Es importante destacar, que las cámaras deben estar conectadas y encendidas para este paso, si no el programa envía un mensaje de error



**Figura 2.4 Botón activar Hardware**  
Fuente: Adaptado del Manual Istra 4D- Q400

Una vez activadas las cámaras, se puede iniciar la calibración con el botón que se muestra en la Figura 2.5. La calibración solo empieza cuando se hace click en el botón circular verde, mostrado en la Figura 2.6

Start a new calibration setup

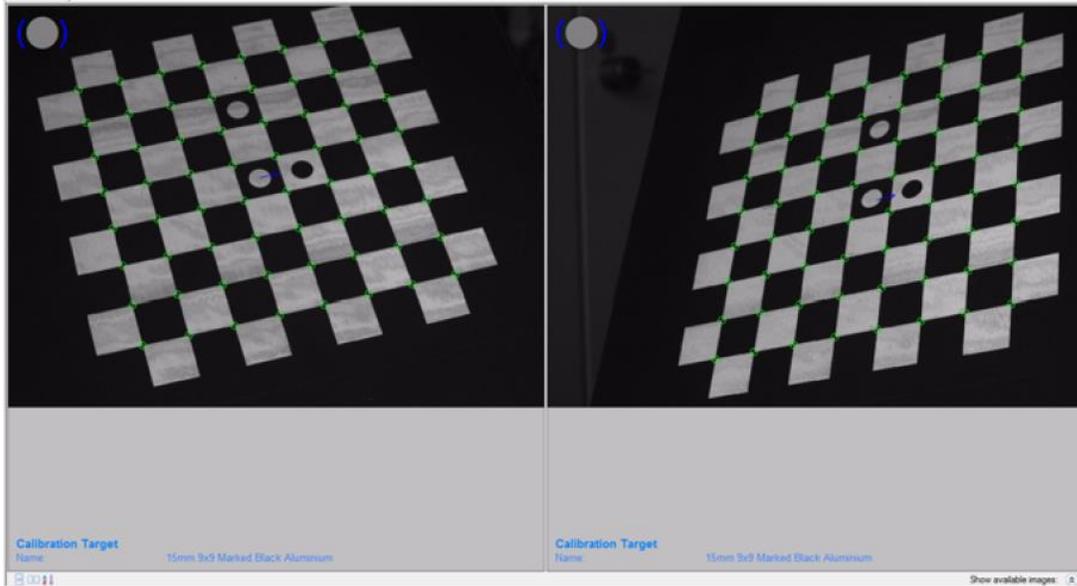


**Figura 2.5 Botón activar calibración**  
Fuente: Adaptado del Manual Istra 4D- Q400

Description	Icon
Starts the automatic calibration process	
Continues the calibration	
Pause the calibration process.	
Stops the calibration process	

**Figura 2.6 Botones de control de calibración**  
Fuente: Adaptado del Manual Istra 4D- Q400

Es necesario contar con una pizarra de calibración de tipo “chessboard”, la cual debiese estar adjunta con el resto del equipo de Dantec Dynamics. Al comenzar la calibración, se abre una interfaz como la mostrada en la Figura 2.7. En esta interfaz se muestran las imágenes adquiridas por ambas cámaras, hay que posicionar el tablero de tal manera en que ambas cámaras lo capturen en su totalidad, procurando abarcar la totalidad del espacio disponible. Antes de comenzar, es necesario enfocar con ambas cámaras el tablero (para recomendaciones de enfoque, ver sección 2.2.).



**Figura 2.7** Interfaz de Calibración, a la izquierda se muestra la imagen adquirida por la cámara 1, mientras que a la derecha se muestra la imagen adquirida por la cámara 2.  
**Fuente:** Adaptado del Manual Istra 4D- Q400

Inmediatamente en la interfaz, aparecerán círculos de colores, cuyo significado se detalla en la figura que se muestra a continuación.

Blue:		
The coordinate system can not be detected		
Green:		
Sufficient number of corners are detected		
Red:		
The coordinate system is detected but not enough corners detected		

**Figura 2.8** Significados de círculos de reconocimiento de la tabla tipo “chessboard”, en la interfaz de calibración  
**Fuente:** Adaptado del Manual Istra 4D- Q400

Una vez encuadrada la imagen, se da click a “Start automatic calibration process”.

### 2.3.2. Toma de fotografías de calibración

Para realizar la calibración, es necesario que el sistema tome al menos 9 fotos. Las primeras tres con el patrón de la tabla tipo “chessboard” de frente a las dos cámaras. De estas tres fotos la

primera debe ser con el tablero completamente recto, para la segunda y tercera foto el tablero debe inclinarse hacia arriba y hacia abajo respectivamente. Para la segunda serie de fotos, se debe apuntar el tablero a una sola cámara, de manera tal que quede completamente encuadrado por una cámara, y la otra cámara pueda aun captar la mayor cantidad de puntos del tablero. La última serie de 3 fotos, es análoga a la anterior, pero apuntando el tablero a la otra cámara.

Es importante que mientras el sistema esté tomando la foto (círculo en la pantalla superior derecha de color amarillo), el tablero debe permanecer completamente inmóvil, de lo contrario, se introducirán errores sistemáticos en el proceso de calibración. A continuación se adjunta la simbología de los círculos de la pantalla superior derecha de la interfaz, que indican el estado de las cámaras.

<p>Grey:</p> <p>Calibration setup started, but the calibration is not active.</p>	
<p>Green:</p> <p>The calibration is active.</p>	
<p>Yellow:</p> <p>Attention, don't move the calibration target because the system will try to acquire an image within the next second.</p>	
<p>Red:</p> <p>The software can't detect the necessary number of markers. Please change the position of the target in a way that more markers are detected.</p>	
<p>Pink:</p> <p>The software is still busy processing the acquired images and calculating the calibration parameters.</p>	

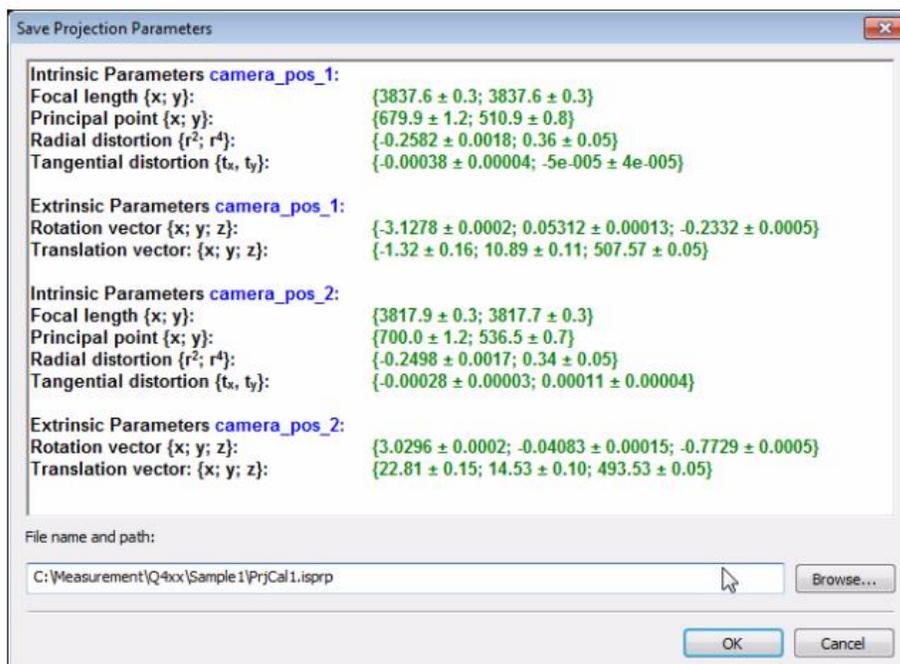
Es importante que durante todo el proceso de toma de fotografías, la interfaz reconozca los puntos del tablero “chessboard” con círculos verdes (es decir, que se reconocen suficientes esquinas). Si aparecen rojos o azules, y se toma una fotografía, la calibración puede ser corrompida completamente.

La calidad de la calibración se ve reflejada en el parámetro Residuo, indicado en la parte inferior de la interfaz del software. El valor para el residuo debe ser menor de 0.3 a 0,5 (dependiendo del destino de calibración se utiliza). Si este número es más de 1,0 se ha producido un error durante el cálculo y la calibración debe repetirse. Un pequeño residuo no es una garantía para una buena calibración, pero un alto residuo es una indicación de una mala calibración.

Optimizing	
Residuum [pixel]:	0.243
Intrinsic parameters	
Focal length {x, y}:	{3837.6; 3837.6}
Principal point {x, y}:	{679.9; 510.9}
Radial distortion {r <sup>2</sup> ; r <sup>4</sup> }:	{-0.2582; 0.36}
Tangential distortion {t <sub>x</sub> , t <sub>y</sub> }:	{-0.00038; -5e-005}
Extrinsic parameters	
Translation {x, y, z}:	{-1.32; 10.89; 507.57}
Rotation {x, y, z}:	{-3.1278; 0.05312; -0.2332}
Calibration Target	
Name:	15mm 9x9 Marked Black Aluminium

Para el tablero de 35x35 mm utilizado en las mediciones, se obtuvo por lo general calibraciones entre 0.28 a 0.33.

Al hacer click en “save” aparecerá el siguiente dialogo que resume los valores obtenidos para calibración



## 2.4. Trabajo Futuro

Tomando en cuenta las apreciaciones y distintos resultados obtenidos a lo largo del proceso de iteración sobre las condiciones de montaje experimental, se recomienda fabricar timbres con distintos tamaños de manchas, para facilitar la realización de mediciones en escalas de tamaño diferentes a la considerada en este trabajo. Además, considerando que una calibración óptima de las cámaras implica mejores resultados en la medición de desplazamientos, se sugiere fabricar un ar-

mazón que permita sostener la placa de calibración de las cámaras, el cual posibilite giro y movimiento de esta en las posiciones requeridas para la calibración. De esta manera se excluiría el error producido por la dependencia de la habilidad de la persona que sostiene la placa de calibración

### 3. Medición de desplazamientos

#### 3.1. Configuración de Mediciones

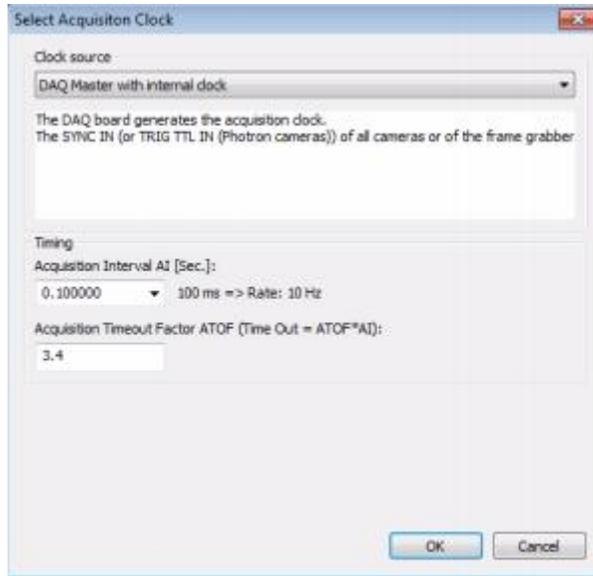
Una vez calibrado el sistema, al pulsar el botón , se abre el módulo de Live Image View, en donde se muestran las imágenes capturadas de ambas cámaras. Lo siguiente es configurar los parámetros de adquisición en el menú



En este, se pueden configurar los aspectos de frecuencia de adquisición y tiempo de exposición de las cámaras.

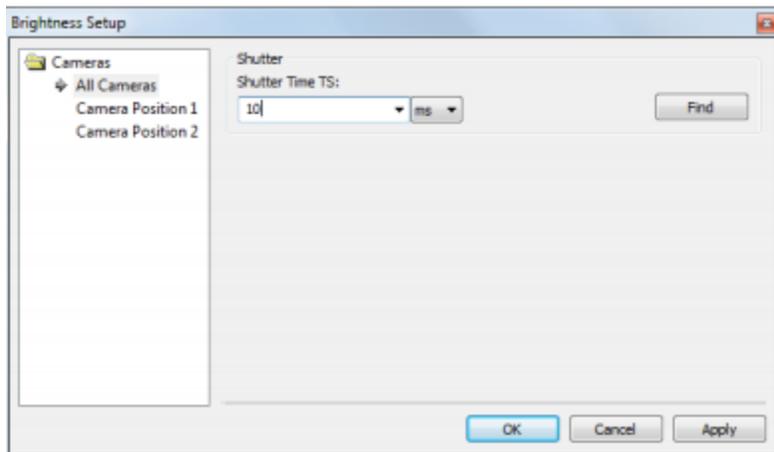
Description	Icon
Acquisition Clock Setup	
Analog Input Setup	
Brightness Setup	

Si se configura la frecuencia de adquisición se abre una interfaz que se muestre a continuación:



En Acquisition Interval se fija el tiempo entre cada fotograma adquirido. En general para pruebas de deformación este parámetro dependerá del fenómeno estudiado, por un lado si se desea estudiar propagación de grietas, el tiempo debe ser suficiente para captar el avance de grieta, por lo que el tiempo de adquisición dependerá de la velocidad de esta última. Por el contrario, si solo importa el estado inicial y final, se pueden utilizar tiempos de adquisición mayores. Para el caso del estudio de flexión se utilizó un tiempo de 250 ms.

Si se regula el tiempo de exposición se abre una ventana como la siguiente

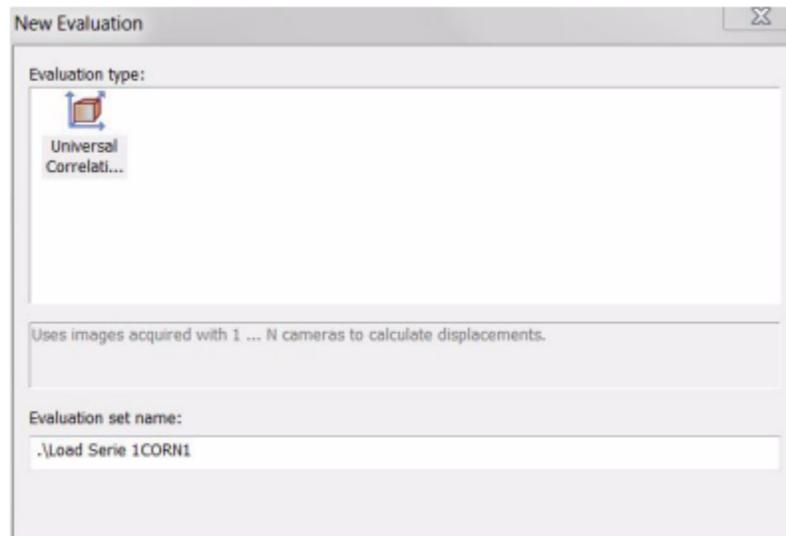


Para el ensayo de flexión se utilizó un valor de exposición de 40 ms.

### 3.2.Reconocimiento de puntos de interés

Para comprobar si los puntos de interés son válidos, se sugiere tomar una sola imagen de la muestra en su estado original. Para esto se debe pulsar el botón  "Start Recording", luego se toma solo una snapshot de la imagen con el botón  y detenemos la grabación .

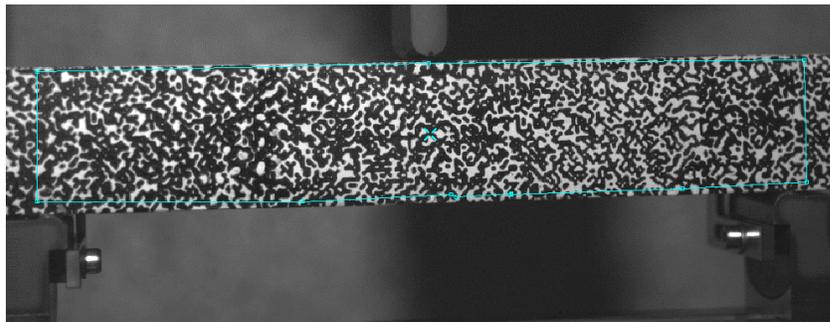
Una vez hecho esto, se procede a iniciar una configuración de calibración con el botón , con la que se abre una configuración:



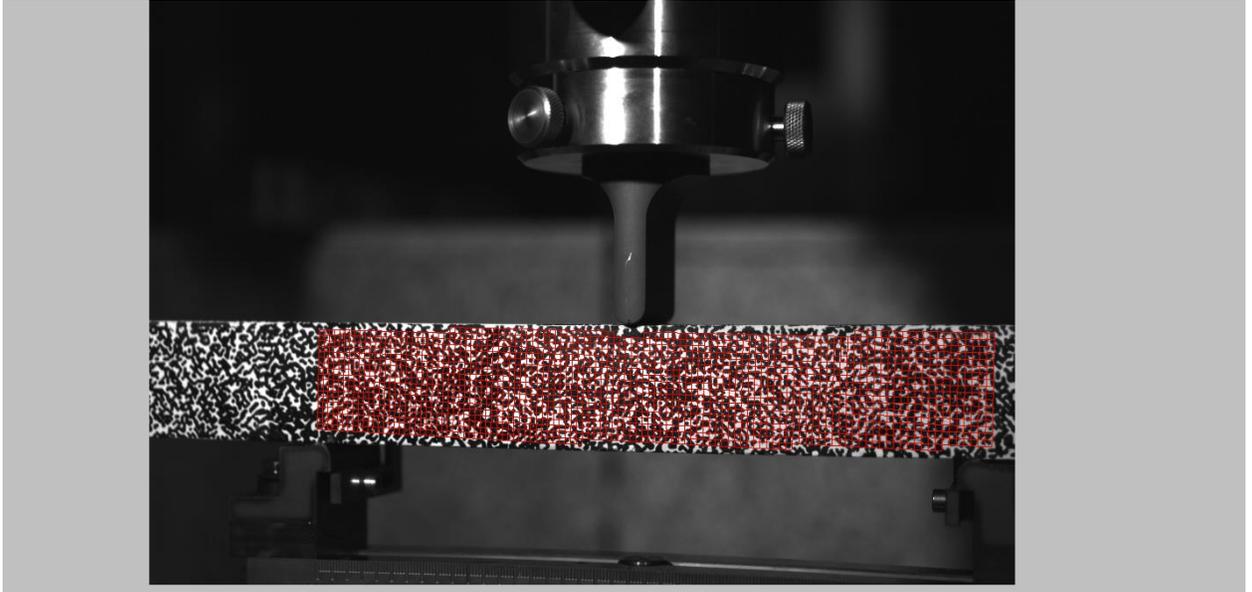
Se selecciona un nombre y damos aceptar. Posteriormente se eligen los parámetros de correlación en el sector derecho de la pantalla en Evaluation Settings, en esta pantalla se selecciona el tamaño de subconjunto de referencia ( Facet Size), y la distancia de paso entre píxeles ( Grid Spacing). Para mayores distancias de paso entre píxeles se tendrá mayor resolución de datos, a costa de mayor costo computacional. Mientras que los subconjuntos de referencia tendrán relación con el tamaño de manchas que se analiza, como se explicó anteriormente. Se sugiere un step de 5 píxeles.



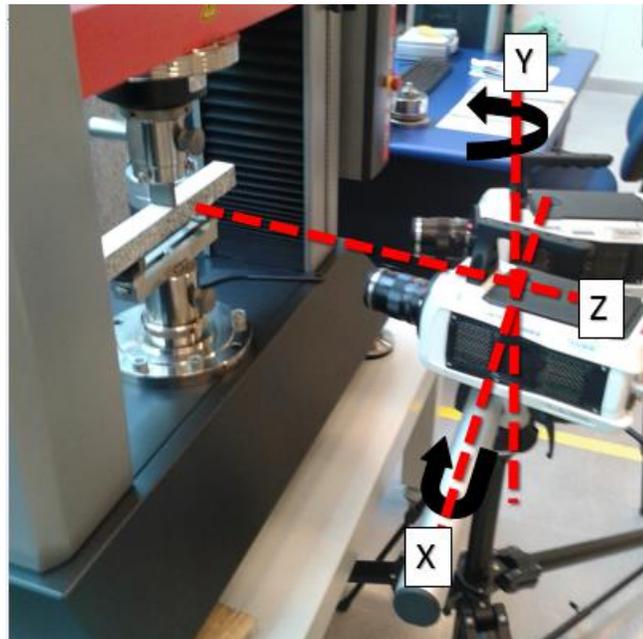
Luego es necesario definir un polígono sobre el cuál se calcularán los desplazamientos. En la sección Mask Definition. Luego damos click en  y seleccionamos la región de interés a medir. Luego con , seleccionamos un punto dentro de la región de interés denominado valor semilla, a partir de este se definirán las ubicaciones de todos los puntos de interés.



Posteriormente damos click a Control y Start Correlation. Luego el programa indicará la cantidad de puntos válidos registrados, para visualizarlos, se debe seleccionar la pestaña “Visualization Settings” y luego en Graphics, seleccionamos “GRID” . y se mostrará una imagen como la siguiente



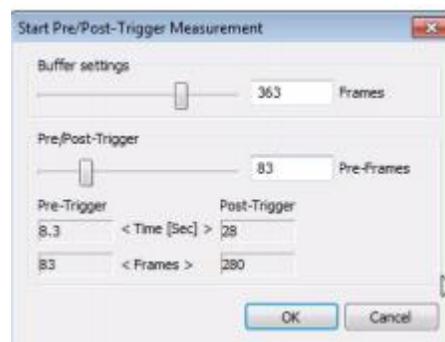
En esta imagen se puede ver si se lograron obtener los puntos requeridos, de lo contrario se tendrá que iterar sobre las condiciones de iluminación, enfoque y ángulos con respecto a los ejes x e y, que se muestran en la figura (con las manivelas propias del soporte de las camaras) y repetir el proceso anterior



### 3.3.Toma de Mediciones

Una vez que se logra el reconocimiento completo de la región de interés, se sugiere tomar una sola una imagen de la muestra en su estado original. Para esto de debe pulsar el boton  "Start Recording", luego se toma solo una snapshot de la imagen con el boton  y detenemos la grabación . Con esto tendrá una imagen de referencia para el cálculo de los desplazamientos.

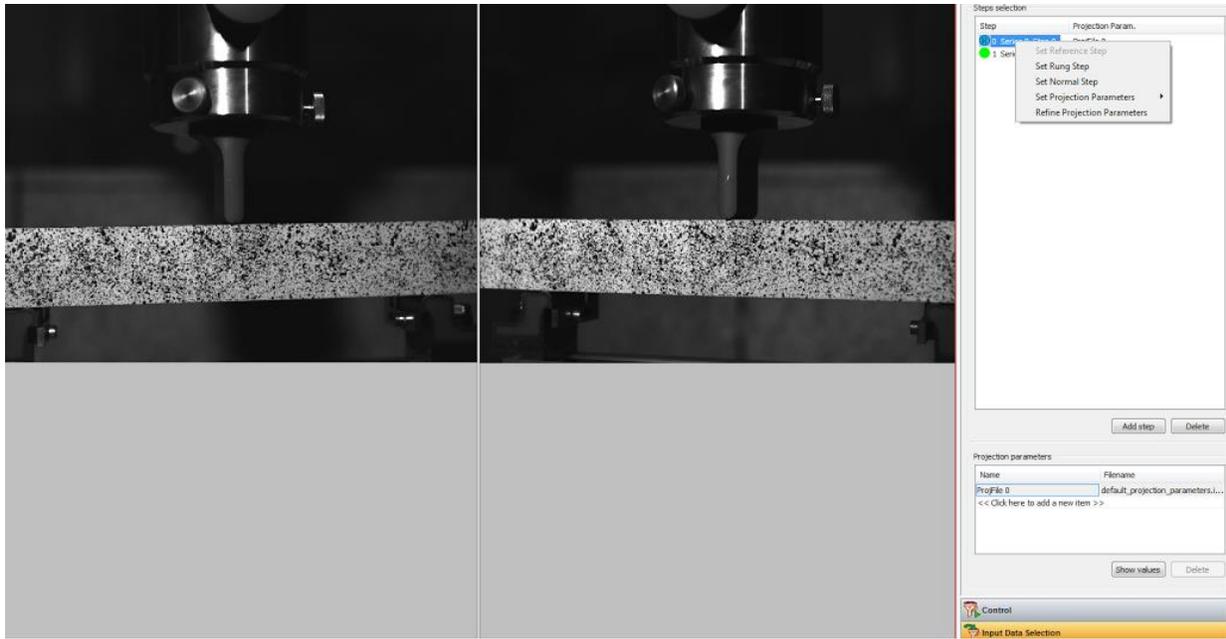
Una vez hecho esto se puede tomar imágenes de forma continua para obtener las mediciones de la experiencia durante su desarrollo. Para esto se debe hacer click en  "Continuos recording" con lo que se abre



En donde se selecciona la cantidad de frames ( con los que se determina la duración de la grabación), el tiempo dependerá de la frecuencia de adquisición de las cámaras y la cantidad de frames finales.

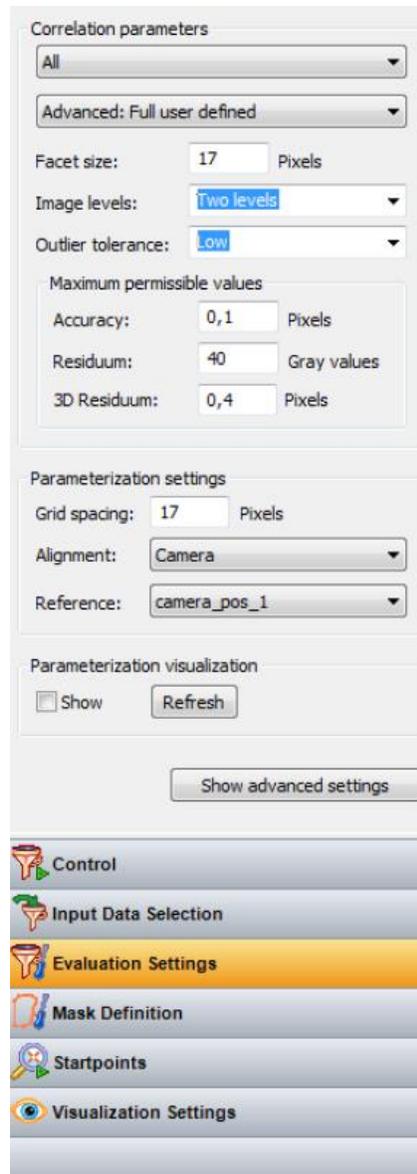
Posteriormente se hace el procedimiento análogo de Start New Evaluation, de la sección anterior.

Si se desea insertar la imagen de referencia, se selecciona la pestaña "Input Data Selection" en el menú de la derecha y posteriormente Add step. Se debe buscar el nombre de la medición de referencia y se agrega a los datos. Para seleccionarla como medida de referencia se debe hacer click derecho en este dato recién ingresado y seleccionar: Set Reference como se muestra en la figura:

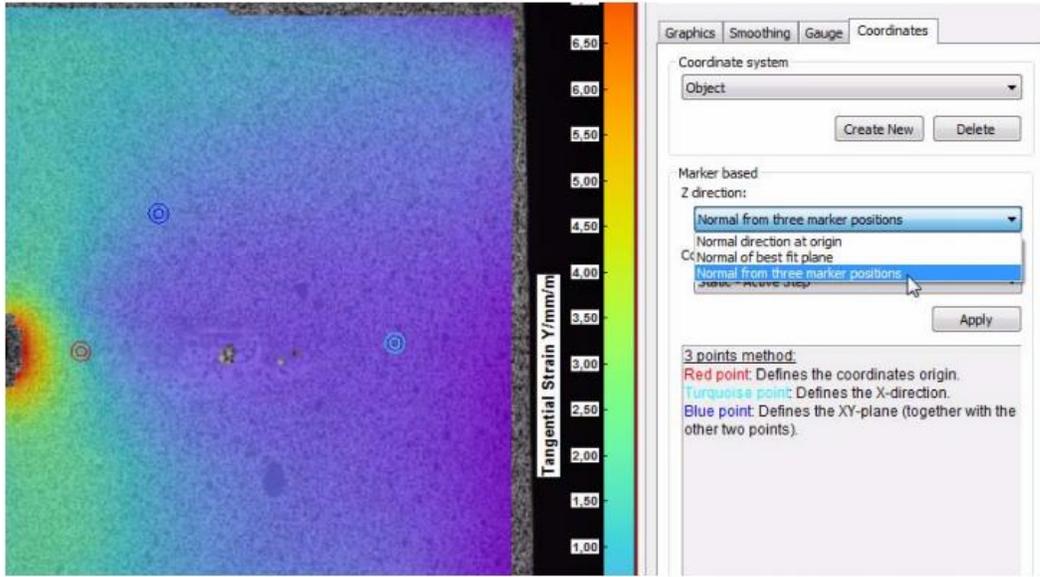


### 3.4. Definir Sistema de Referencias

Para definir el sistema de referencias, es necesario ya haber realizado la correlación de imágenes luego en las opciones de post-procesamiento del software, en la pestaña Visualization Settings



Luego, en este módulo se selecciona opción coordinates, donde se abre la siguiente interfaz en la que podemos crear un nuevo sistema de coordenadas según 3 opciones.



**Normal direction at the origin:** Define el origen del sistema con el círculo rojo, y junto con el círculo celeste, definen la línea del eje x. El eje y se define entonces como perpendicular hacia arriba de este eje x. El eje z a su vez, se define por el vector normal a la superficie cercana al círculo rojo.

**Normal of best fit plane:** el eje x e y se definen igual que en el método anterior, pero el eje z se define como el vector normal a la superficie plana reconstruida de mejor ajuste de datos

**Normal from three marker points:** en este caso el eje x e y se definen igual que en el método anterior, pero el eje z se define a partir del vector perpendicular al plano definido por los 3 puntos.

Es importante notar que para arrastrar los círculos es necesario apretar CTRL y luego arrastrarlos con el click del mouse apretado. **(ESTO NO SALE EN EL MANUAL)**

## 4. Exportar datos

Finalmente los datos son exportados con el botón  , en donde se debe elegir la opción hdf5. Cuya estructura de datos se encuentra explicada en detalle en la referencia[7] y se refleja en los siguientes esquemas

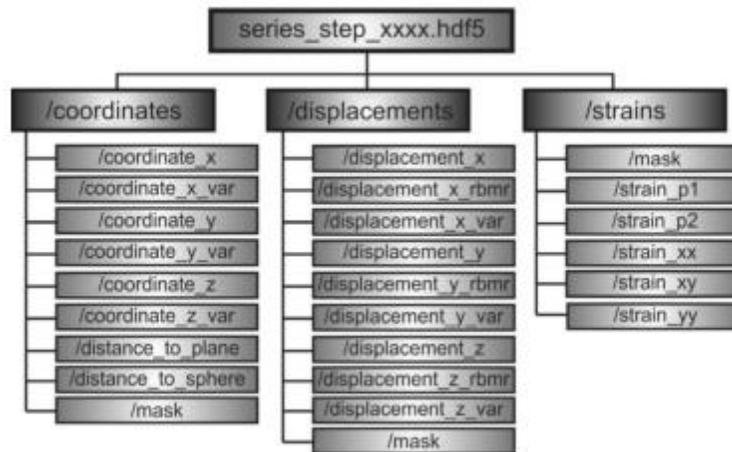
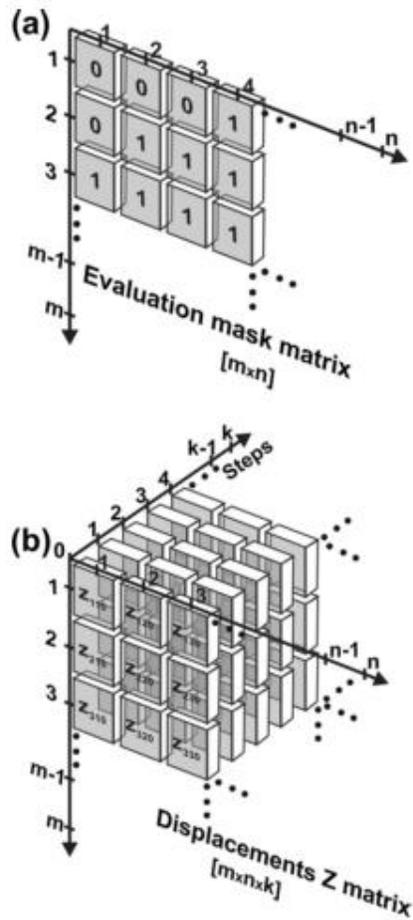


Fig. 2. HDF5 file exported by the program Istra4D.

Ejemplos de como las matrices se ordenan en el formato hdf5 ->



## 5. Referencias

- [1] Correlated Solutions, “Vic-2D Testing Guide,” 2009.
- [2] Correlated Solutions, “Digital Image Correlation : Overview of Principles and Software 2D Image Correlation Fundamentals,” *South Carolina Univ.*, 2009.
- [3] P. Lava, S. Cooreman, and D. Debruyne, “Study of systematic errors in strain fields obtained via DIC using heterogeneous deformation generated by plastic FEA,” *Opt. Lasers Eng.*, vol. 48, no. 4, pp. 457–468, 2010.
- [4] P. K. Thomas Becker, Karsten Splitthof, Thorsten Siebert, “Error Estimations of 3D Digital Image Correlation Measurements Introduction,” in *Proc. SPIE 6341, Speckle06: Speckles, From Grains to Flowers*, 2006, pp. 1–6.
- [5] Correlated Solutions, “Bias and Noise in DIC - Part 1 Identifying and Addressing bias,” 2011.

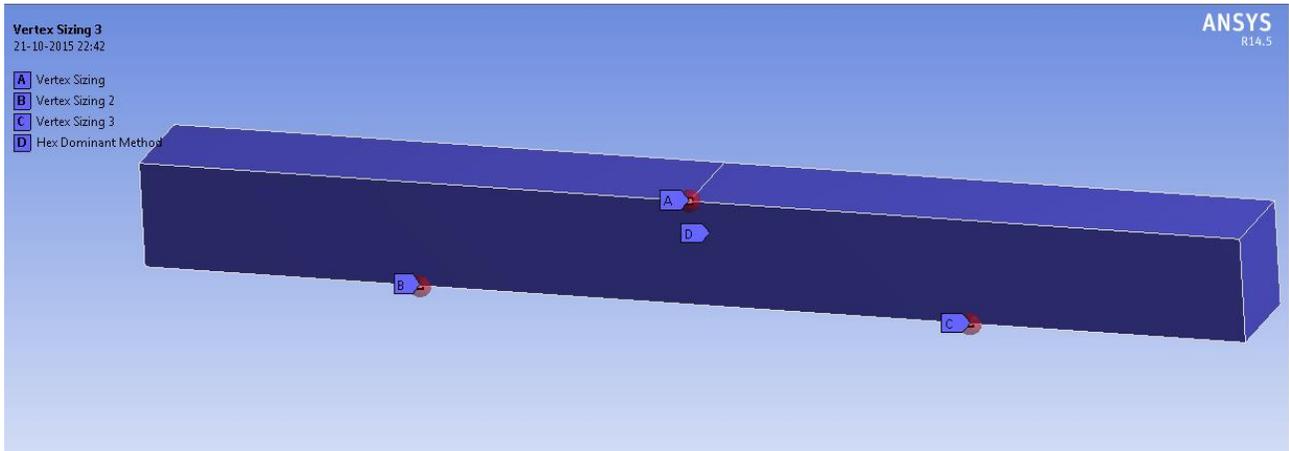
- [6] Correlated Solutions, “Bias and Noise in DIC - Part 2 Quantifying & Minimizing Noise,” 2011.
- [7] F. Trebuña and M. Hagara, “Experimental modal analysis performed by high-speed digital image correlation system,” *Measurement*, vol. 50, pp. 78–85, 2014.

## Anexo B: Definición de malla de Elementos Finitos

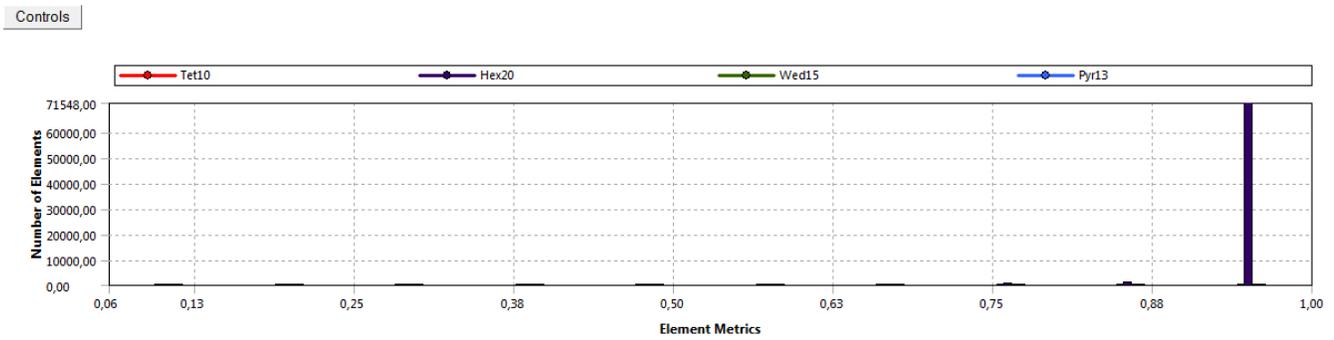
Tabla B.1. Parámetros de mallado global, local principales y estadísticas de malla.

Parámetros	Valor
<b>Mallado Global</b>	
Relevancia	25
Centro de Relevancia	Fino
Tamaño de Elementos	2 [mm]
Suavizado	Medio
Transición	Lenta
Centro de Ángulo "Span"	Medio
Tamaño Mínimo de Borde	38 [mm]
<b>Parámetros de Refinamiento Local</b>	
Tipo: Esfera de Influencia	
Tamaño de Esfera	4 [mm]
Tamaño de Elementos	1 [mm]
<b>Estadísticas</b>	
Nodos	318164
Elementos	74173
Calidad de Elemento Mínima	5,72E-02
Calidad de Elemento Máxima	0,999998
Calidad de Elemento Promedio	0,9869
Desviación Estándar de Calidad	7,41E-02

Fuente: Elaboración propia.



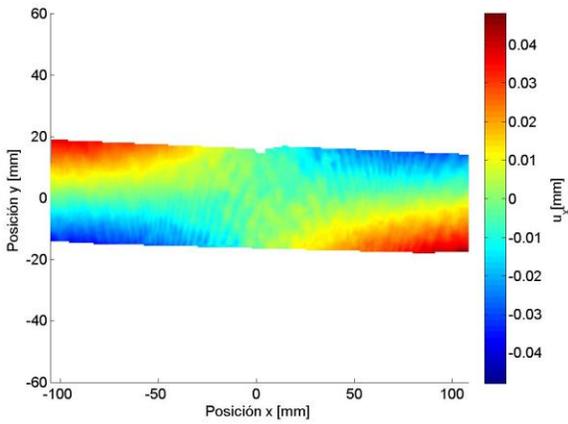
**Figura B.1** Mallado Hexa-dominante en todo el volumen de la geometría D. Refinado por medio de esfera de influencia en el punto de aplicación de fuerza A y en apoyos B y C.  
**Fuente:** Elaboración propia a partir de ANSYS v.14.5.



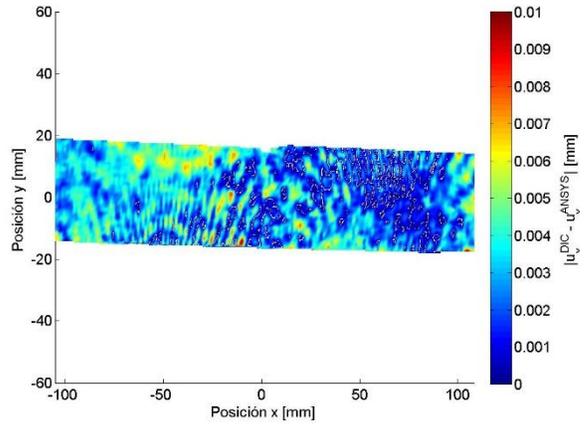
**Figura B.2:** Estadísticas de la calidad de elementos del modelo en Elementos Finitos  
**Fuente:** Elaboración propia a partir de ANSYS v.14.5.

## Anexo C: Muestras No Incluidas En Resultados

### C.1 Distribución de Desplazamientos Horizontales y Desviación Estándar Probeta tipo Cepillo



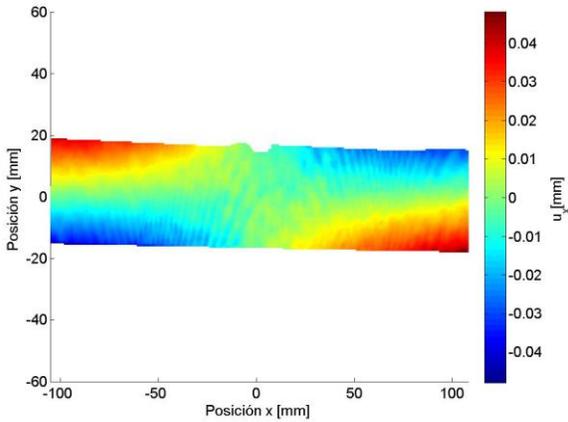
(a) Desplazamientos horizontales  $u_x$



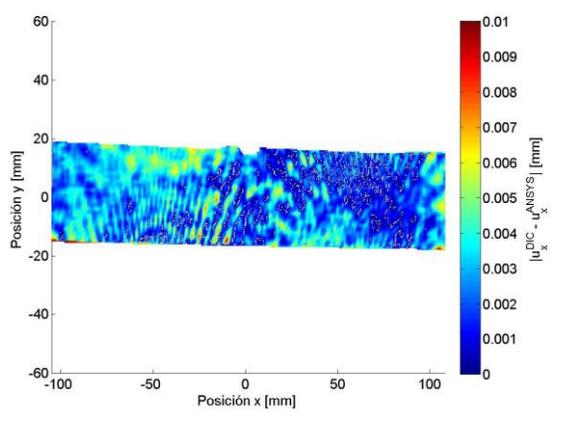
(b) Desviación estándar de los datos

**Figura C.1. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre”, sub set de referencia 17x17, y distancia de paso 5 píxeles.**

**Fuente: Elaboración propia**



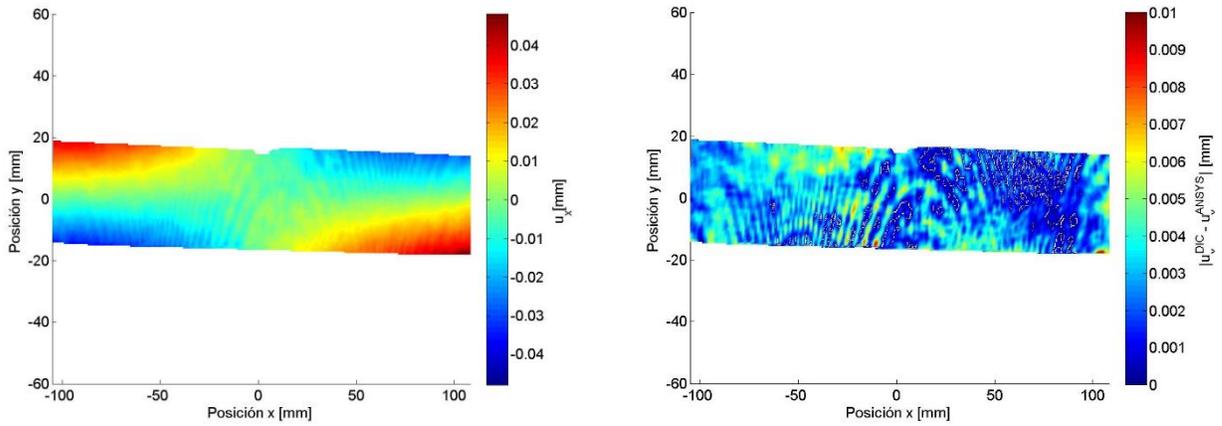
(a) Desplazamientos horizontales  $u_x$



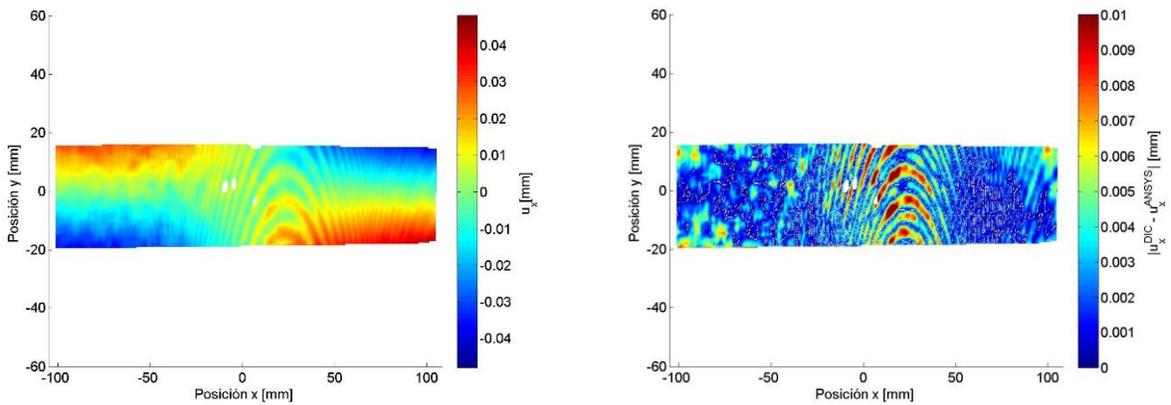
(b) Desviación estándar de los datos (MATLAB)

**Figura C.2. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**

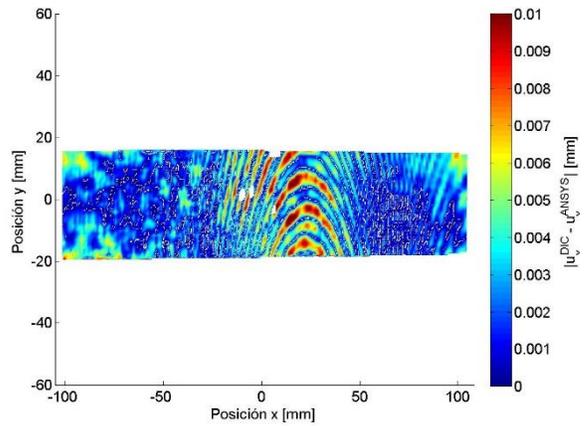
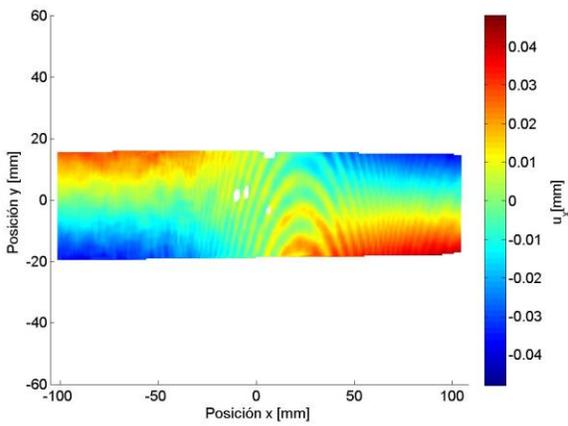
**Fuente: Elaboración propia**



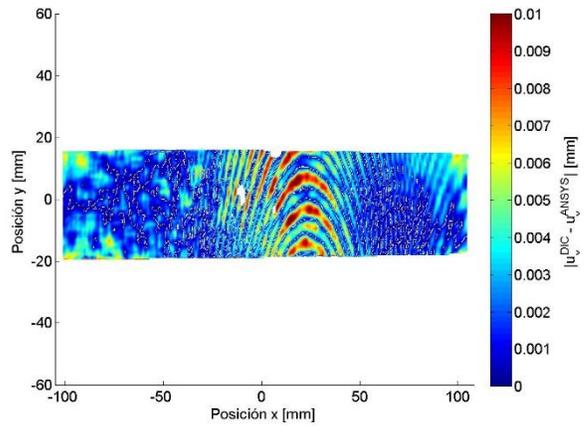
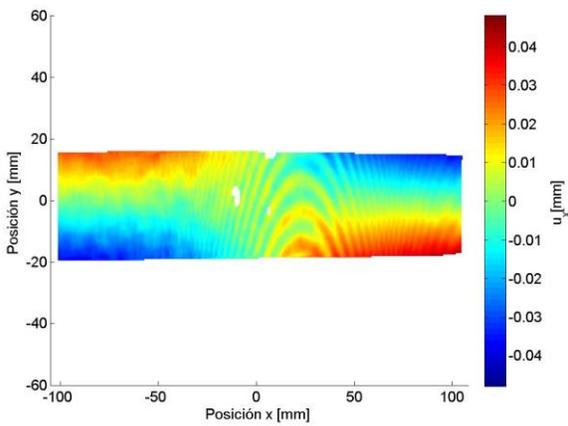
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.3. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



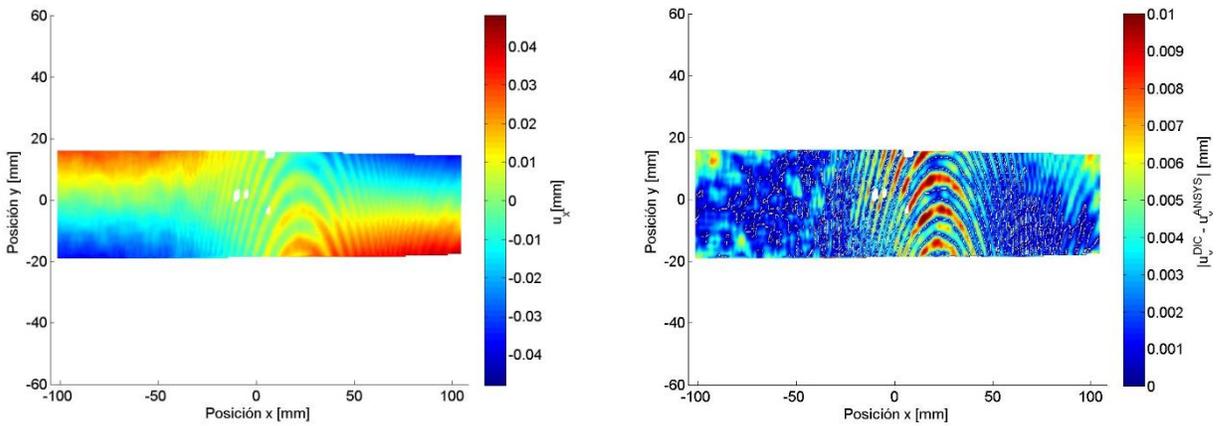
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.4. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.5. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**

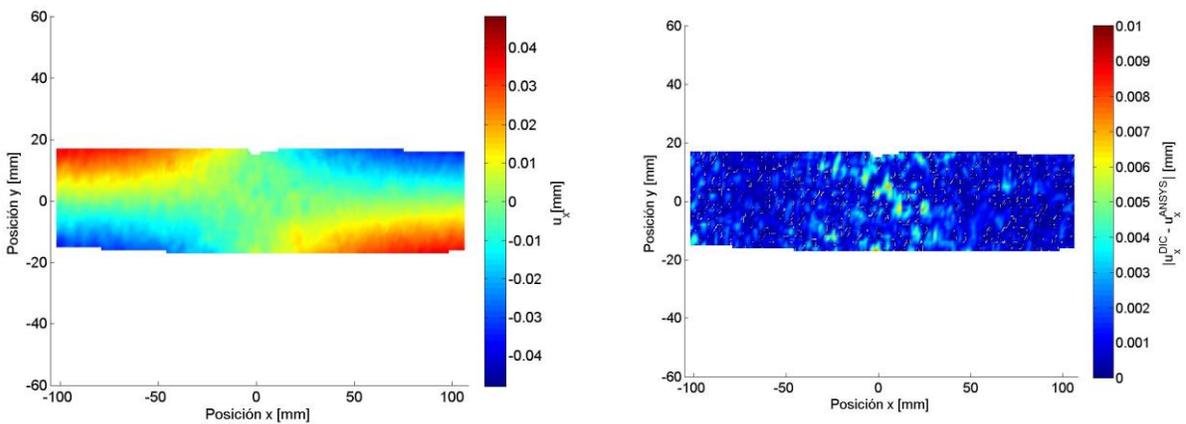


(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.6. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**

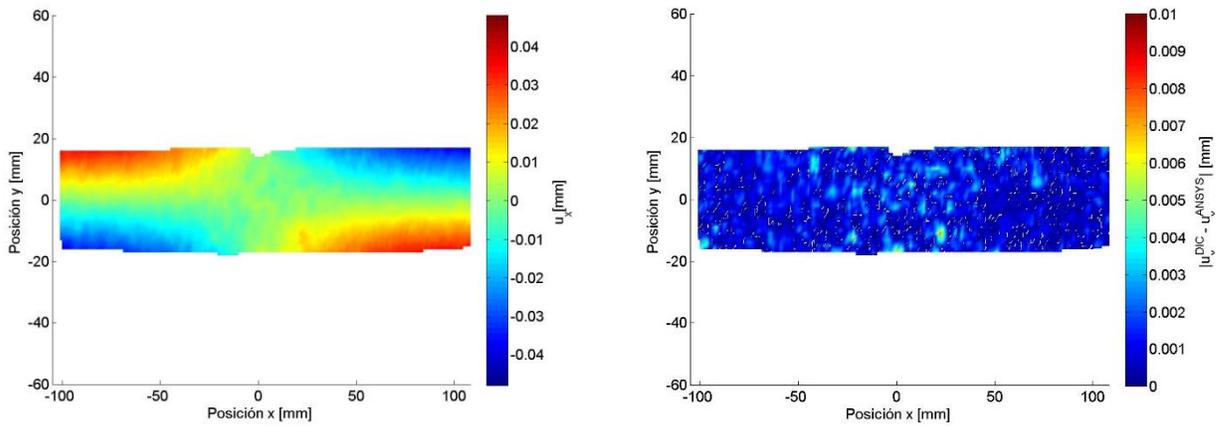


(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.7. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**

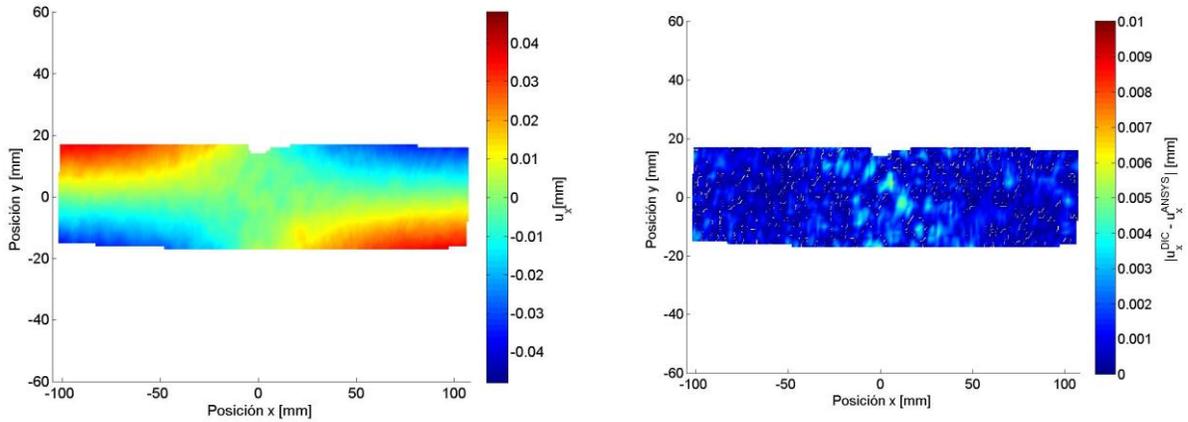
### C.2 Distribución de Desplazamientos Horizontales y Desviación Estándar Probeta con estampado timbre



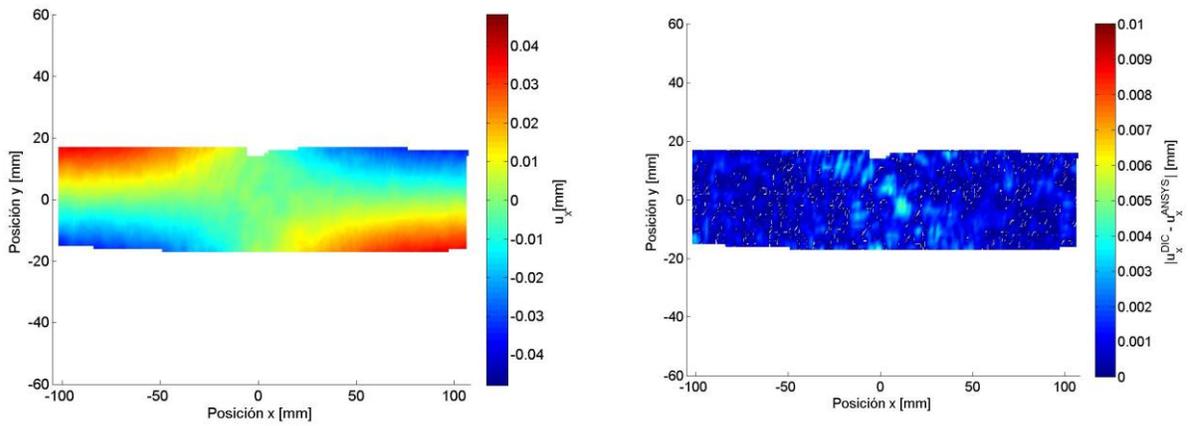
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.8. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



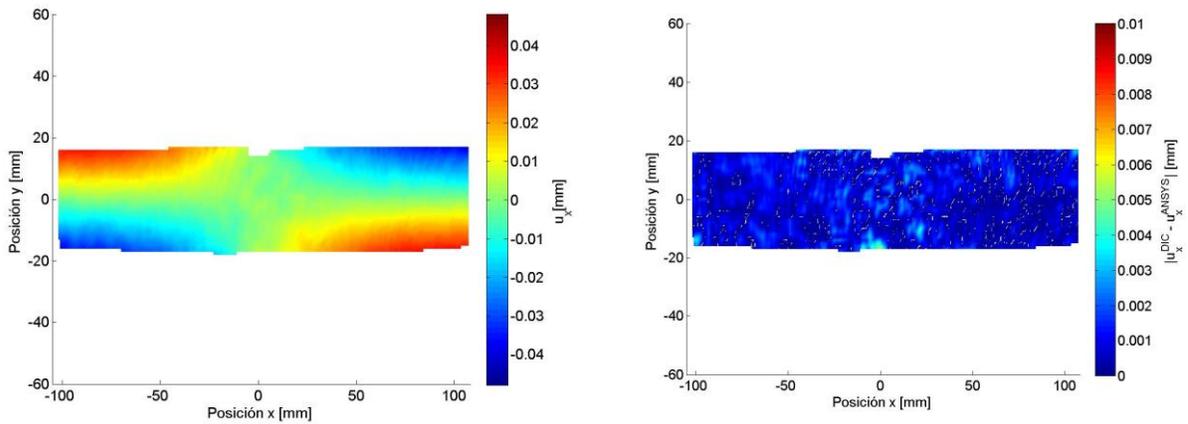
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.9. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



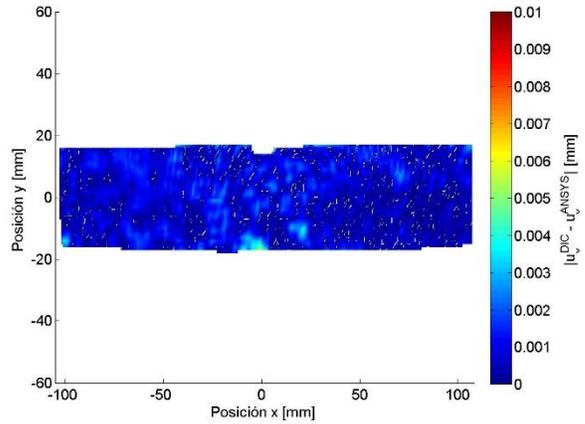
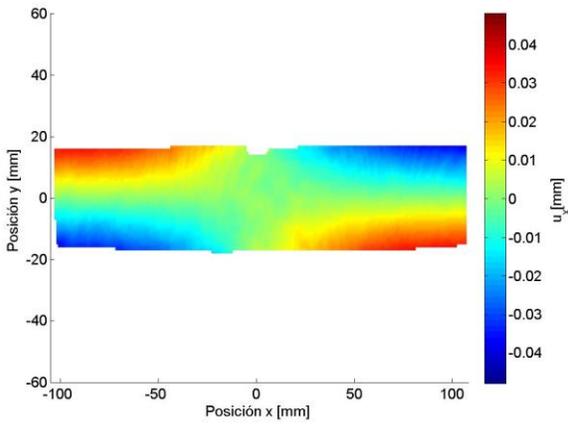
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.10. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.11. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



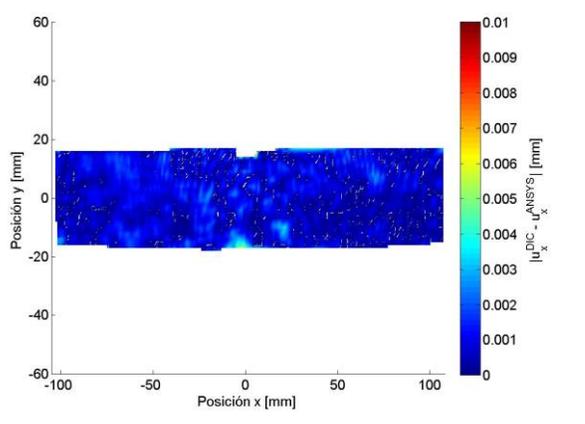
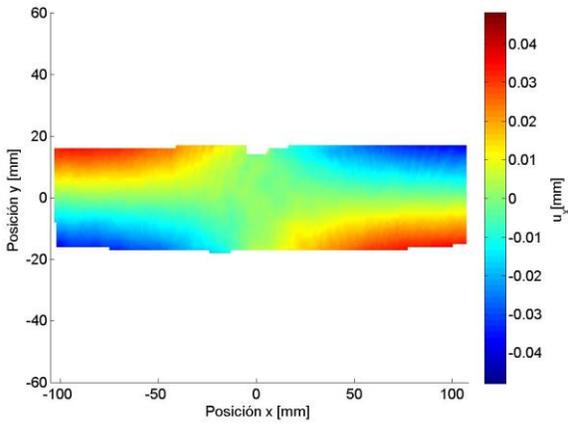
(a) Desplazamientos horizontales  $u_x$  (b) Desviación estándar de los datos (MATLAB)  
**Figura C.12. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.**  
**Fuente: Elaboración propia**



(a) Desplazamientos horizontales  $u_x$

(b) Desviación estándar de los datos (MATLAB)

**Figura C.13. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.  
Fuente: Elaboración propia**



(a) Desplazamientos horizontales  $u_x$

(b) Desviación estándar de los datos (MATLAB)

**Figura C.14. Resultados de la medición DIC, para la probeta con patrón tipo “Timbre” sub set de referencia 29x29 y distancia de paso 5 píxeles.  
Fuente: Elaboración propia**

## Anexo D: Códigos MATLAB

### D.1 Función Importar\_datos(), get\_disp\_x(),get\_disp\_y(),get\_disp\_z()

```
function varargout = Importar_datos(varargin)
% FFT_GUI MATLAB code for fft_GUI.fig
%   FFT_GUI, by itself, creates a new FFT_GUI or raises the existing
%   singleton*.
%
%   H = FFT_GUI returns the handle to a new FFT_GUI or the handle to
%   the existing singleton*.
%
%   FFT_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FFT_GUI.M with the given input arguments.
%
%   FFT_GUI('Property','Value',...) creates a new FFT_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before fft_GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to fft_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help fft_GUI

% Last Modified by GUIDE v2.5 10-May-2014 18:11:17

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @fft_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @fft_GUI_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before fft_GUI is made visible.
function fft_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to fft_GUI (see VARARGIN)

% Choose default command line output for fft_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes fft_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = fft_GUI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in load_files.
function load_files_Callback(hObject, eventdata, handles)
% hObject handle to load_files (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global stat_txtStatus;
[files_names, files_dir] = uigetfile('*.hdf5', 'Select the files', 'MultiSe-
lect', 'on');
setappdata(0, 'files_names', files_names)

try
    old_folder = cd(files_dir);
catch
    return;
end

set(handles.listbox1, 'String', files_names)

set(stat_txtStatus, 'String', 'Start Load x y coordinates');
% Load coordinates x,y
filename = files_names{1,1};
coord_x = hdf5read(filename, 'coordinates/coordinate_x');

coord_y = hdf5read(filename, 'coordinates/coordinate_y');

mask = logical(hdf5read(filename, 'coordinates/mask'));

coord_z = hdf5read(filename, 'coordinates/coordinate_z');

```

```

coord_x(~mask)=0;
coord_y(~mask)=0;
coord_z(~mask)=0;

save coord_x coord_x
save coord_y coord_y
save coord_z coord_z

setappdata(0, 'coord_x', coord_x);
setappdata(0, 'coord_y', coord_y);

set(stat_txtStatus, 'String', 'Start Load z coordinates');
drawnow;
% refreshdata(handles.txtStatus);
guidata(handles.txtStatus, handles);
%Load displacements z
try
cd(old_folder);
catch
return;
end
displ_z_all = get_disp_z(files_names, files_dir);
try
cd(old_folder);
catch
return;
end
displ_x_all = get_disp_x(files_names, files_dir);
try
cd(old_folder);
catch
return;
end
displ_y_all = get_disp_y(files_names, files_dir);
set(stat_txtStatus, 'String', 'Coordinates loaded -> Select Points');
save displ_x_all displ_x_all;
save displ_y_all displ_y_all;
save displ_z_all displ_z_all;
setappdata(0, 'displ_z_all', displ_z_all);

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject handle to listbox1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns listbox1 contents
as cell array
% contents{get(hObject, 'Value')} returns selected item from listbox1

% --- Executes during object creation, after setting all properties.

```

```

function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function choose_freq_CreateFcn(hObject, eventdata, handles)
% hObject    handle to choose_freq (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function txtStatus_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txtStatus (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global stat_txtStatus
stat_txtStatus=hObject;
set(stat_txtStatus,'String',' -> Load Files ');
function [displ_x_all] = get_disp_x(files_names,files_dir)
try
    old_folder = cd(files_dir);
catch
    return;
end
n = size(files_names);

    for i=1:n(2)

        file = H5F.open(files_names{1,i}, 'H5F_ACC_RDONLY', 'H5P_DEFAULT');
        displ_x = hdf5read(files_names{1,i}, 'displacements/displacement_x_rbmnr');
        displ_x_var = hdf5read(files_names{1,i}, 'coordinates/coordinate_x_var');
        mask = logical(hdf5read(files_names{1,i}, 'displacements/mask'));
        displ_x(~mask) = NaN;
        displ_x_var(~mask) = NaN;
        % mask_all(:, :, i) = mask; % => unuseful if mask_all is not an output

```

```

    displ_x_all(:, :, i) = displ_x;
    displ_x_var(:, :, i) = displ_x_var;
    close(file);

end
var = flipud(displ_x_var(:, :, 1));
%   maxi = nanmax(nanmax(var));
%   mini = nanmin(nanmin(var));
%   prom = nanmean(nanmean(var));
%
%   display(maxi);
%   display(mini);
%   display(prom);

save displ_x_all displ_x_all;
save mask
save var;
figure()
%   M = displ_x_all(:, :, 2);
%   dmaxi = nanmax(nanmax(M));
%   dmini = nanmin(nanmin(M));
%
%   display(dmaxi);
%   display(dmini);
%
%
s=files_dir;
lbslash = find(s == '\\', 2, 'last');
name = s(lbslash+1:end);
title(name);
try
    cd(old_folder);
catch
    return;
end
end

function [displ_y_all] = get_disp_y(files_names, files_dir)
try
    old_folder = cd(files_dir);
catch
    return;
end
n = size(files_names);

for i=1:n(2)

    file = H5F.open(files_names{1,i}, 'H5F_ACC_RDONLY', 'H5P_DEFAULT');
    displ_y = hdf5read(files_names{1,i}, 'displacements/displacement_y');
    mask = logical(hdf5read(files_names{1,i}, 'displacements/mask'));
    displ_y(~mask) = NaN;

    % mask_all(:, :, i) = mask; % => unuseful if mask_all is not an output
    displ_y_all(:, :, i) = displ_y;

```

```

        close(file);
    end
    save displ_y_all displ_y_all;

try
    cd(old_folder);
catch
    return;
end
end
function [displ_z_all] = get_disp_z(files_names,files_dir)
try
    old_folder = cd(files_dir);
catch
    return;
end
n = size(files_names);

    for i=1:n(2)

        file = HDF5.open(files_names{1,i}, 'H5F_ACC_RDONLY', 'H5P_DEFAULT');
        displ_z = hdf5read(files_names{1,i}, 'displacements/displacement_z');
        mask = logical(hdf5read(files_names{1,i}, 'displacements/mask'));
        displ_z(~mask) = NaN;

        % mask_all(:, :, i) = mask; % => unuseful if mask_all is not an output
        displ_z_all(:, :, i) = displ_z;

        close(file);
    end
    save displ_z_all displ_z_all;
    try
        cd(old_folder);
    catch
        return;
    end
    save displ_z_all displ_z_all;
    save mask
end

```

## D.2 Método de suavizado de desplazamientos “smoothn.m”

```
function [z,s,exitflag] = smoothn(varargin)

%SMOOTHN Robust spline smoothing for 1-D to N-D data.
% SMOOTHN provides a fast, automatized and robust discretized spline
% smoothing for data of arbitrary dimension.
%
% Z = SMOOTHN(Y) automatically smoothes the uniformly-sampled array Y. Y
% can be any N-D noisy array (time series, images, 3D data,...). Non
% finite data (NaN or Inf) are treated as missing values.
%
% Z = SMOOTHN(Y,S) smoothes the array Y using the smoothing parameter S.
% S must be a real positive scalar. The larger S is, the smoother the
% output will be. If the smoothing parameter S is omitted (see previous
% option) or empty (i.e. S = []), it is automatically determined by
% minimizing the generalized cross-validation (GCV) score.
%
% Z = SMOOTHN(Y,W) or Z = SMOOTHN(Y,W,S) smoothes Y using a weighting
% array W of positive values, that must have the same size as Y. Note
% that a nil weight corresponds to a missing value.
%
% If you want to smooth a vector field or multicomponent data, Y must be
% a cell array. For example, if you need to smooth a 3-D vectorial flow
% (Vx,Vy,Vz), use Y = {Vx,Vy,Vz}. The output Z is also a cell array which
% contains the smoothed components. See examples 5 to 8 below.
%
% [Z,S] = SMOOTHN(...) also returns the calculated value for the
% smoothness parameter S so that you can fine-tune the smoothing
% subsequently if needed.
%
%
% 1) ROBUST smoothing
% -----
% Z = SMOOTHN(...,'robust') carries out a robust smoothing that minimizes
% the influence of outlying data.
%
% An iteration process is used with the 'ROBUST' option, or in the
% presence of weighted and/or missing values. Z = SMOOTHN(...,OPTIONS)
% smoothes with the termination parameters specified in the structure
% OPTIONS. OPTIONS is a structure of optional parameters that change the
% default smoothing properties. It must be last input argument.
% ---
% The structure OPTIONS can contain the following fields:
% -----
% OPTIONS.TolZ:      Termination tolerance on Z (default = 1e-3),
%                   OPTIONS.TolZ must be in ]0,1[
% OPTIONS.MaxIter:  Maximum number of iterations allowed
%                   (default = 100)
% OPTIONS.Initial:  Initial value for the iterative process
%                   (default = original data, Y)
% OPTIONS.Weight:   Weight function for robust smoothing:
%                   'bisure' (default), 'talworth' or 'cauchy'
% -----
%
% [Z,S,EXITFLAG] = SMOOTHN(...) returns a boolean value EXITFLAG that
% describes the exit condition of SMOOTHN:
```

```

%      1      SMOOTHN converged.
%      0      Maximum number of iterations was reached.
%
%
% 2) Different spacing increments
% -----
% SMOOTHN, by default, assumes that the spacing increments are constant
% and equal in all the directions (i.e. dx = dy = dz = ...). This means
% that the smoothness parameter is also similar for each direction. If
% the increments differ from one direction to the other, it can be useful
% to adapt these smoothness parameters. You can thus use the following
% field in OPTIONS:
%     OPTIONS.Spacing' = [d1 d2 d3...],
% where dI represents the spacing between points in the Ith dimension.
%
% Important note: d1 is the spacing increment for the first
% non-singleton dimension (i.e. the vertical direction for matrices).
%
%
% 3) REFERENCES (please refer to the two following papers)
% -----
% 1) Garcia D, Robust smoothing of gridded data in one and higher
% dimensions with missing values. Computational Statistics & Data
% Analysis, 2010;54:1167-1178.
% <a
% href="matlab:web('http://www.biome-
cardio.com/pageshtm/publi/csda10.pdf')">PDF download</a>
% 2) Garcia D, A fast all-in-one method for automated post-processing of
% PIV data. Exp Fluids, 2011;50:1247-1259.
% <a
% href="matlab:web('http://www.biomecardio.com/pageshtm/publi/me-
dial11.pdf')">PDF download</a>
%
%
% EXAMPLES:
% -----
% --- Example #1: smooth a curve ---
% x = linspace(0,100,2^8);
% y = cos(x/10)+(x/50).^2 + randn(size(x))/10;
% y([70 75 80]) = [5.5 5 6];
% z = smoothn(y); % Regular smoothing
% zr = smoothn(y,'robust'); % Robust smoothing
% subplot(121), plot(x,y,'r.',x,z,'k','LineWidth',2)
% axis square, title('Regular smoothing')
% subplot(122), plot(x,y,'r.',x,zr,'k','LineWidth',2)
% axis square, title('Robust smoothing')
%
% --- Example #2: smooth a surface ---
% xp = 0:.02:1;
% [x,y] = meshgrid(xp);
% f = exp(x+y) + sin((x-2*y)*3);
% fn = f + randn(size(f))*0.5;
% fs = smoothn(fn);
% subplot(121), surf(xp,xp,fn), zlim([0 8]), axis square
% subplot(122), surf(xp,xp,fs), zlim([0 8]), axis square
%
% --- Example #3: smooth an image with missing data ---

```

```

% n = 256;
% y0 = peaks(n);
% y = y0 + randn(size(y0))*2;
% I = randperm(n^2);
% y(I(1:n^2*0.5)) = NaN; % lose 1/2 of data
% y(40:90,140:190) = NaN; % create a hole
% z = smoothn(y); % smooth data
% subplot(2,2,1:2), imagesc(y), axis equal off
% title('Noisy corrupt data')
% subplot(223), imagesc(z), axis equal off
% title('Recovered data ...')
% subplot(224), imagesc(y0), axis equal off
% title('... compared with the actual data')
%
%
% --- Example #4: smooth volumetric data ---
% [x,y,z] = meshgrid(-2:.2:2);
% xslice = [-0.8,1]; yslice = 2; zslice = [-2,0];
% vn = x.*exp(-x.^2-y.^2-z.^2) + randn(size(x))*0.06;
% subplot(121), slice(x,y,z,vn,xslice,yslice,zslice,'cubic')
% title('Noisy data')
% v = smoothn(vn);
% subplot(122), slice(x,y,z,v,xslice,yslice,zslice,'cubic')
% title('Smoothed data')
%
%
% --- Example #5: smooth a cardioid ---
% t = linspace(0,2*pi,1000);
% x = 2*cos(t).*(1-cos(t)) + randn(size(t))*0.1;
% y = 2*sin(t).*(1-cos(t)) + randn(size(t))*0.1;
% z = smoothn({x,y});
% plot(x,y,'r.',z{1},z{2},'k','linewidth',2)
% axis equal tight
%
%
% --- Example #6: smooth a 3-D parametric curve ---
% t = linspace(0,6*pi,1000);
% x = sin(t) + 0.1*randn(size(t));
% y = cos(t) + 0.1*randn(size(t));
% z = t + 0.1*randn(size(t));
% u = smoothn({x,y,z});
% plot3(x,y,z,'r.',u{1},u{2},u{3},'k','linewidth',2)
% axis tight square
%
%
% --- Example #7: smooth a 2-D vector field ---
% [x,y] = meshgrid(linspace(0,1,24));
% Vx = cos(2*pi*x+pi/2).*cos(2*pi*y);
% Vy = sin(2*pi*x+pi/2).*sin(2*pi*y);
% Vx = Vx + sqrt(0.05)*randn(24,24); % adding Gaussian noise
% Vy = Vy + sqrt(0.05)*randn(24,24); % adding Gaussian noise
% I = randperm(numel(Vx));
% Vx(I(1:30)) = (rand(30,1)-0.5)*5; % adding outliers
% Vy(I(1:30)) = (rand(30,1)-0.5)*5; % adding outliers
% Vx(I(31:60)) = NaN; % missing values
% Vy(I(31:60)) = NaN; % missing values
% Vs = smoothn({Vx,Vy},'robust'); % automatic smoothing
% subplot(121), quiver(x,y,Vx,Vy,2.5), axis square
% title('Noisy velocity field')
% subplot(122), quiver(x,y,Vs{1},Vs{2}), axis square
% title('Smoothed velocity field')

```

```

%
% --- Example #8: smooth a 3-D vector field ---
% load wind % original 3-D flow
% -- Create noisy data
% Gaussian noise
un = u + randn(size(u))*8;
vn = v + randn(size(v))*8;
wn = w + randn(size(w))*8;
% Add some outliers
I = randperm(numel(u)) < numel(u)*0.025;
un(I) = (rand(1,nnz(I))-0.5)*200;
vn(I) = (rand(1,nnz(I))-0.5)*200;
wn(I) = (rand(1,nnz(I))-0.5)*200;
% -- Visualize the noisy flow (see CONEPLLOT documentation)
figure, title('Noisy 3-D vectorial flow')
xmin = min(x(:)); xmax = max(x(:));
ymin = min(y(:)); ymax = max(y(:));
zmin = min(z(:));
daspect([2,2,1])
xrange = linspace(xmin,xmax,8);
yrange = linspace(ymin,ymax,8);
zrange = 3:4:15;
[cx cy cz] = meshgrid(xrange,yrange,zrange);
k = 0.1;
hcones = coneplot(x,y,z,un*k,vn*k,wn*k,cx,cy,cz,0);
set(hcones,'FaceColor','red','EdgeColor','none')
hold on
wind_speed = sqrt(un.^2 + vn.^2 + wn.^2);
hsurfaces = slice(x,y,z,wind_speed,[xmin,xmax],ymax,zmin);
set(hsurfaces,'FaceColor','interp','EdgeColor','none')
hold off
axis tight; view(30,40); axis off
camproj perspective; camzoom(1.5)
camlight right; lighting phong
set(hsurfaces,'AmbientStrength',.6)
set(hcones,'DiffuseStrength',.8)
% -- Smooth the noisy flow
Vs = smoothn({un,vn,wn},'robust');
% -- Display the result
figure, title('3-D flow smoothed by SMOOTHN')
daspect([2,2,1])
hcones = coneplot(x,y,z,Vs{1}*k,Vs{2}*k,Vs{3}*k,cx,cy,cz,0);
set(hcones,'FaceColor','red','EdgeColor','none')
hold on
wind_speed = sqrt(Vs{1}.^2 + Vs{2}.^2 + Vs{3}.^2);
hsurfaces = slice(x,y,z,wind_speed,[xmin,xmax],ymax,zmin);
set(hsurfaces,'FaceColor','interp','EdgeColor','none')
hold off
axis tight; view(30,40); axis off
camproj perspective; camzoom(1.5)
camlight right; lighting phong
set(hsurfaces,'AmbientStrength',.6)
set(hcones,'DiffuseStrength',.8)
%
% See also SMOOTH1Q, DCTN, IDCTN.
%
% -- Damien Garcia -- 2009/03, revised 2014/10

```

```

% website: <a
% href="matlab:web('http://www.biomecardio.com')">www.BiomeCardio.com</a>

%% Check input arguments
error(nargchk(1,5,nargin));
OPTIONS = struct;
NArgIn = nargin;
if isstruct(varargin{end}) % SMOOTHN(...,OPTIONS)
    OPTIONS = varargin{end};
    NArgIn = NArgIn-1;
end
idx = cellfun(@ischar,varargin);
if any(idx) % SMOOTHN(...,'robust',...)
    assert(sum(idx)==1 && strcmpi(varargin{idx},'robust'),...
        'Wrong syntax. Read the help text for instructions.')
    isrobust = true;
    assert(find(idx)==NArgIn,...
        'Wrong syntax. Read the help text for instructions.')
    NArgIn = NArgIn-1;
else
    isrobust = false;
end

%% Test & prepare the variables
%---
% y = array to be smoothed
y = varargin{1};
if ~iscell(y), y = {y}; end
sizy = size(y{1});
ny = numel(y); % number of y components
for i = 1:ny
    assert(isequal(sizy,size(y{i})),...
        'Data arrays in Y must have the same size.')
    y{i} = double(y{i});
end
noe = prod(sizy); % number of elements
if noe==1, z = y{1}; s = []; exitflag = true; return, end
%---
% Smoothness parameter and weights
W = ones(sizy);
s = [];
if NArgIn==2
    if isempty(varargin{2}) || isscalar(varargin{2}) % smoothn(y,s)
        s = varargin{2}; % smoothness parameter
    else % smoothn(y,W)
        W = varargin{2}; % weight array
    end
elseif NArgIn==3 % smoothn(y,W,s)
    W = varargin{2}; % weight array
    s = varargin{3}; % smoothness parameter
end
assert(isnumeric(W), 'W must be a numeric array')
assert(isnumeric(s), 'S must be a numeric scalar')
assert(isequal(size(W),sizy),...
    'Arrays for data and weights (Y and W) must have same size.')
assert(isempty(s) || (isscalar(s) && s>0),...

```

```

    'The smoothing parameter S must be a scalar >0')
%---
% Field names in the structure OPTIONS
OptionNames = fieldnames(OPTIONS);
idx = ismember(OptionNames,...
    {'TolZ','MaxIter','Initial','Weight','Spacing','Order'});
assert(all(idx),...
    ['' OptionNames{find(~idx,1)} '' is not a valid field name for OP-
    TIONS.'])
%---
% "Maximal number of iterations" criterion
if ~ismember('MaxIter',OptionNames)
    OPTIONS.MaxIter = 100; % default value for MaxIter
end
assert(isnumeric(OPTIONS.MaxIter) && isscalar(OPTIONS.MaxIter) &&...
    OPTIONS.MaxIter>=1 && OPTIONS.MaxIter==round(OPTIONS.MaxIter),...
    'OPTIONS.MaxIter must be an integer >=1')
%---
% "Tolerance on smoothed output" criterion
if ~ismember('TolZ',OptionNames)
    OPTIONS.TolZ = 1e-3; % default value for TolZ
end
assert(isnumeric(OPTIONS.TolZ) && isscalar(OPTIONS.TolZ) &&...
    OPTIONS.TolZ>0 && OPTIONS.TolZ<1, 'OPTIONS.TolZ must be in ]0,1[')
%---
% "Initial Guess" criterion
if ~ismember('Initial',OptionNames)
    isinitial = false;
else
    isinitial = true;
    z0 = OPTIONS.Initial;
    if ~iscell(z0), z0 = {z0}; end
    nz0 = numel(z0); % number of y components
    assert(nz0==ny,...
        'OPTIONS.Initial must contain a valid initial guess for Z')
    for i = 1:nz0
        assert(isequal(sizy,size(z0{i})),...
            'OPTIONS.Initial must contain a valid initial guess for Z')
        z0{i} = double(z0{i});
    end
end
end
%---
% "Weight function" criterion (for robust smoothing)
if ~ismember('Weight',OptionNames)
    OPTIONS.Weight = 'bisure'; % default weighting function
else
    assert(ischar(OPTIONS.Weight),...
        'A valid weight function (OPTIONS.Weight) must be chosen')
    assert(any(ismember(lower(OPTIONS.Weight),...
        {'bisure','talworth','cauchy'})),...
        'The weight function must be ''bisure'', ''cauchy'' or '' tal-
        worth''.')
end
end
%---
% "Order" criterion (by default m = 2)
% Note: m = 0 is of course not recommended!
if ~ismember('Order',OptionNames)

```

```

    m = 2; % order
else
    m = OPTIONS.Order;
    if ~ismember(m,0:2);
        error('MATLAB:smoothn:IncorrectOrder',...
            'The order (OPTIONS.order) must be 0, 1 or 2.')
    end
end
end
%---
% "Spacing" criterion
d = ndims(y{1});
if ~ismember('Spacing',OptionNames)
    dI = ones(1,d); % spacing increment
else
    dI = OPTIONS.Spacing;
    assert(isnumeric(dI) && length(dI)==d,...
        'A valid spacing (OPTIONS.Spacing) must be chosen')
end
dI = dI/max(dI);
%---
% Weights. Zero weights are assigned to not finite values (Inf or NaN),
% (Inf/NaN values = missing data).
IsFinite = isfinite(y{1});
for i = 2:ny, IsFinite = IsFinite & isfinite(y{i}); end
nof = nnz(IsFinite); % number of finite elements
W = W.*IsFinite;
assert(all(W(:)>=0), 'Weights must all be >=0')
W = W/max(W(:));
%---
% Weighted or missing data?
isweighted = any(W(:)<1);
%---
% Automatic smoothing?
isauto = isempty(s);

%% Create the Lambda tensor
%---
% Lambda contains the eigenvalues of the difference matrix used in this
% penalized least squares process (see CSDA paper for details)
d = ndims(y{1});
Lambda = zeros(sizy);
for i = 1:d
    siz0 = ones(1,d);
    siz0(i) = sizy(i);
    Lambda = bsxfun(@plus,Lambda,...
        (2-2*cos(pi*(reshape(1:sizy(i),siz0)-1)/sizy(i)))/dI(i)^2);
end
if ~isauto, Gamma = 1./(1+s*Lambda.^m); end

%% Upper and lower bound for the smoothness parameter
% The average leverage (h) is by definition in [0 1]. Weak smoothing occurs
% if h is close to 1, while over-smoothing appears when h is near 0. Upper
% and lower bounds for h are given to avoid under- or over-smoothing. See
% equation relating h to the smoothness parameter for m = 2 (Equation #12
% in the referenced CSDA paper).

```

```

N = sum(sizy~=1); % tensor rank of the y-array
hMin = 1e-6; hMax = 0.99;
if m==0 % Not recommended. For mathematical purpose only.
    sMinBnd = 1/hMax^(1/N)-1;
    sMaxBnd = 1/hMin^(1/N)-1;
elseif m==1
    sMinBnd = (1/hMax^(2/N)-1)/4;
    sMaxBnd = (1/hMin^(2/N)-1)/4;
elseif m==2
    sMinBnd = (((1+sqrt(1+8*hMax^(2/N)))/4/hMax^(2/N))^2-1)/16;
    sMaxBnd = (((1+sqrt(1+8*hMin^(2/N)))/4/hMin^(2/N))^2-1)/16;
end

%% Initialize before iterating
%---
Wtot = W;
%--- Initial conditions for z
if isweighted
    %--- With weighted/missing data
    % An initial guess is provided to ensure faster convergence. For that
    % purpose, a nearest neighbor interpolation followed by a coarse
    % smoothing are performed.
    %---
    if isinitial % an initial guess (z0) has been already given
        z = z0;
    else
        z = InitialGuess(y,IsFinite);
    end
else
    z = cell(size(y));
    for i = 1:ny, z{i} = zeros(sizy); end
end
%---
z0 = z;
for i = 1:ny
    y{i}(~IsFinite) = 0; % arbitrary values for missing y-data
end
%---
tol = 1;
RobustIterativeProcess = true;
RobustStep = 1;
nit = 0;
DCTy = cell(1,ny);
vec = @(x) x(:);
%--- Error on p. Smoothness parameter s = 10^p
errp = 0.1;
opt = optimset('TolX',errp);
%--- Relaxation factor RF: to speedup convergence
RF = 1 + 0.75*isweighted;

%% Main iterative process
%---
while RobustIterativeProcess
    %--- "amount" of weights (see the function GCVscore)
    aow = sum(Wtot(:))/noe; % 0 < aow <= 1
    %---

```

```

while tol>OPTIONS.TolZ && nit<OPTIONS.MaxIter
    nit = nit+1;
    for i = 1:ny
        DCTy{i} = dctn(Wtot.*(y{i}-z{i})+z{i});
    end
    if isauto && ~rem(log2(nit),1)
        %---
        % The generalized cross-validation (GCV) method is used.
        % We seek the smoothing parameter S that minimizes the GCV
        % score i.e. S = Argmin(GCVscore).
        % Because this process is time-consuming, it is performed from
        % time to time (when the step number - nit - is a power of 2)
        %---
        fminbnd(@gcv,log10(sMinBnd),log10(sMaxBnd),opt);
    end
    for i = 1:ny
        z{i} = RF*idctn(Gamma.*DCTy{i}) + (1-RF)*z{i};
    end

    % if no weighted/missing data => tol=0 (no iteration)
    tol = isweighted*norm(vec([z0{:}]-[z{:}]))/norm(vec([z{:}]));

    z0 = z; % re-initialization
end
exitflag = nit<OPTIONS.MaxIter;

if isrobust %-- Robust Smoothing: iteratively re-weighted process
    %--- average leverage
    h = 1;
    for k = 1:N
        if m==0 % not recommended - only for numerical purpose
            h0 = 1/(1+s/dI(k)^(2^m));
        elseif m==1
            h0 = 1/sqrt(1+4*s/dI(k)^(2^m));
        elseif m==2
            h0 = sqrt(1+16*s/dI(k)^(2^m));
            h0 = sqrt(1+h0)/sqrt(2)/h0;
        else
            error('m must be 0, 1 or 2.')
        end
        h = h*h0;
    end
    %--- take robust weights into account
    Wtot = W.*RobustWeights(y,z,IsFinite,h,OPTIONS.Weight);
    %--- re-initialize for another iterative weighted process
    isweighted = true; tol = 1; nit = 0;
    %---
    RobustStep = RobustStep+1;
    RobustIterativeProcess = RobustStep<4; % 3 robust steps are enough.
else
    RobustIterativeProcess = false; % stop the whole process
end
end

%% Warning messages
%---

```

```

if isauto
    if abs(log10(s)-log10(sMinBnd))<errp
        warning('MATLAB:smoothn:SLowerBound',...
            ['S = ' num2str(s,'%3e') ': the lower bound for S ',...
            'has been reached. Put S as an input variable if required.'])
    elseif abs(log10(s)-log10(sMaxBnd))<errp
        warning('MATLAB:smoothn:SUpperBound',...
            ['S = ' num2str(s,'%3e') ': the upper bound for S ',...
            'has been reached. Put S as an input variable if required.'])
    end
end
if nargout<3 && ~exitflag
    warning('MATLAB:smoothn:MaxIter',...
        ['Maximum number of iterations (' int2str(OPTIONS.MaxIter) ') has
        been exceeded. ',...
        'Increase MaxIter option (OPTIONS.MaxIter) or decrease TolZ (OP-
        TIONS.TolZ) value.'])
end

if numel(z)==1, z = z{:}; end

%% GCV score
%---
function GCVscore = gcv(p)
    % Search the smoothing parameter s that minimizes the GCV score
    %---
    s = 10^p;
    Gamma = 1./(1+s*Lambda.^m);
    %--- RSS = Residual sum-of-squares
    RSS = 0;
    if aow>0.9 % aow = 1 means that all of the data are equally weighted
        % very much faster: does not require any inverse DCT
        for kk = 1:ny
            RSS = RSS + norm(DCTy{kk}(:).*(Gamma(:)-1))^2;
        end
    else
        % take account of the weights to calculate RSS:
        for kk = 1:ny
            yhat = idctn(Gamma.*DCTy{kk});
            RSS = RSS + norm(sqrt(Wtot(IsFinite)).*...
                (y{kk}(IsFinite)-yhat(IsFinite)))^2;
        end
    end
end
%---
TrH = sum(Gamma(:));
GCVscore = RSS/nof/(1-TrH/noe)^2;
end

end

function W = RobustWeights(y,z,I,h,wstr)
    % One seeks the weights for robust smoothing...
    ABS = @(x) sqrt(sum(abs(x).^2,2));
    r = cellfun(@minus,y,z,'UniformOutput',0); % residuals
    r = cellfun(@(x) x(:),r,'UniformOutput',0);

```

```

rI = cell2mat(cellfun(@(x) x(I),r,'UniformOutput',0));
MMED = median(rI); % marginal median
AD = ABS(bsxfun(@minus,rI,MMED)); % absolute deviation
MAD = median(AD); % median absolute deviation

%-- Studentized residuals
u = ABS(cell2mat(r))/(1.4826*MAD)/sqrt(1-h);
u = reshape(u,size(I));

switch lower(wstr)
case 'cauchy'
    c = 2.385; W = 1./(1+(u/c).^2); % Cauchy weights
case 'talworth'
    c = 2.795; W = u<c; % Talworth weights
case 'bisquare'
    c = 4.685; W = (1-(u/c).^2).^2.*((u/c)<1); % bisquare weights
otherwise
    error('MATLAB:smoothn:IncorrectWeights',...
        'A valid weighting function must be chosen')
end
W(isnan(W)) = 0;

% NOTE:
% ----
% The RobustWeights subfunction looks complicated since we work with cell
% arrays. For better clarity, here is how it would look like without the
% use of cells. Much more readable, isn't it?
%
% function W = RobustWeights(y,z,I,h)
%     % weights for robust smoothing.
%     r = y-z; % residuals
%     MAD = median(abs(r(I)-median(r(I)))); % median absolute deviation
%     u = abs(r/(1.4826*MAD)/sqrt(1-h)); % studentized residuals
%     c = 4.685; W = (1-(u/c).^2).^2.*((u/c)<1); % bisquare weights
%     W(isnan(W)) = 0;
% end

end

%% Initial Guess with weighted/missing data
function z = InitialGuess(y,I)
ny = numel(y);
%-- nearest neighbor interpolation (in case of missing values)
if any(~I(:))
    z = cell(size(y));
    if license('test','image_toolbox')
        for i = 1:ny
            [z{i},L] = bwdist(I);
            z{i} = y{i};
            z{i}(~I) = y{i}(L(~I));
        end
    else
        % If BWDIST does not exist, NaN values are all replaced with the
        % same scalar. The initial guess is not optimal and a warning
        % message thus appears.
        z = y;
    end
end

```

```

        for i = 1:ny
            z{i}(~I) = mean(y{i}(I));
        end
        warning('MATLAB:smoothn:InitialGuess',...
            ['BWDIST (Image Processing Toolbox) does not exist. ',...
            'The initial guess may not be optimal; additional',...
            ' iterations can thus be required to ensure complete',...
            ' convergence. Increase ''MaxIter'' criterion if neces-
sary.'])
    end
else
    z = y;
end
%-- coarse fast smoothing using one-tenth of the DCT coefficients
siz = size(z{1});
z = cellfun(@(x) dctn(x),z, 'UniformOutput',0);
for k = 1:ndims(z{1})
    for i = 1:ny
        z{i}(ceil(siz(k)/10)+1:end,:) = 0;
        z{i} = reshape(z{i},circshift(siz,[0 1-k]));
        z{i} = shiftdim(z{i},1);
    end
end
z = cellfun(@(x) idctn(x),z, 'UniformOutput',0);
end

%% DCTN
function y = dctn(y)

%DCTN N-D discrete cosine transform.
% Y = DCTN(X) returns the discrete cosine transform of X. The array Y is
% the same size as X and contains the discrete cosine transform
% coefficients. This transform can be inverted using IDCTN.
%
% Reference
% -----
% Narasimha M. et al, On the computation of the discrete cosine
% transform, IEEE Trans Comm, 26, 6, 1978, pp 934-936.
%
% Example
% -----
%     RGB = imread('autumn.tif');
%     I = rgb2gray(RGB);
%     J = dctn(I);
%     imshow(log(abs(J)),[]), colormap(jet), colorbar
%
% The commands below set values less than magnitude 10 in the DCT matrix
% to zero, then reconstruct the image using the inverse DCT.
%
%     J(abs(J)<10) = 0;
%     K = idctn(J);
%     figure, imshow(I)
%     figure, imshow(K,[0 255])
%
% -- Damien Garcia -- 2008/06, revised 2011/11
% -- www.BiomeCardio.com --

```

```

y = double(y);
sizy = size(y);
y = squeeze(y);
dimy = ndims(y);

% Some modifications are required if Y is a vector
if isvector(y)
    dimy = 1;
    if size(y,1)==1, y = y.'; end
end

% Weighting vectors
w = cell(1,dimy);
for dim = 1:dimy
    n = (dimy==1)*numel(y) + (dimy>1)*sizy(dim);
    w{dim} = exp(1i*(0:n-1)'*pi/2/n);
end

% --- DCT algorithm ---
if ~isreal(y)
    y = complex(dctn(real(y)),dctn(imag(y)));
else
    for dim = 1:dimy
        siz = size(y);
        n = siz(1);
        y = y([1:2:n 2*floor(n/2):-2:2],:);
        y = reshape(y,n,[]);
        y = y*sqrt(2*n);
        y = ifft(y,[],1);
        y = bsxfun(@times,y,w{dim});
        y = real(y);
        y(1,:) = y(1,+)/sqrt(2);
        y = reshape(y,siz);
        y = shiftdim(y,1);
    end
end

y = reshape(y,sizy);

end

%% IDCTN
function y = idctn(y)

%IDCTN N-D inverse discrete cosine transform.
% X = IDCTN(Y) inverts the N-D DCT transform, returning the original
% array if Y was obtained using Y = DCTN(X).
%
% Reference
% -----
% Narasimha M. et al, On the computation of the discrete cosine
% transform, IEEE Trans Comm, 26, 6, 1978, pp 934-936.
%
% Example

```

```

% -----
%     RGB = imread('autumn.tif');
%     I = rgb2gray(RGB);
%     J = dctn(I);
%     imshow(log(abs(J)),[], colormap(jet), colorbar
%
% The commands below set values less than magnitude 10 in the DCT matrix
% to zero, then reconstruct the image using the inverse DCT.
%
%     J(abs(J)<10) = 0;
%     K = idctn(J);
%     figure, imshow(I)
%     figure, imshow(K,[0 255])
%
% See also DCTN, IDSTN, IDCT, IDCT2, IDCT3.
%
% -- Damien Garcia -- 2009/04, revised 2011/11
% -- www.BiomeCardio.com --

y = double(y);
sizy = size(y);
y = squeeze(y);
dimy = ndims(y);

% Some modifications are required if Y is a vector
if isvector(y)
    dimy = 1;
    if size(y,1)==1
        y = y.';
    end
end

% Weighing vectors
w = cell(1,dimy);
for dim = 1:dimy
    n = (dimy==1)*numel(y) + (dimy>1)*sizy(dim);
    w{dim} = exp(1i*(0:n-1)'*pi/2/n);
end

% --- IDCT algorithm ---
if ~isreal(y)
    y = complex(idctn(real(y)),idctn(imag(y)));
else
    for dim = 1:dimy
        siz = size(y);
        n = siz(1);
        y = reshape(y,n,[]);
        y = bsxfun(@times,y,w{dim});
        y(1,:) = y(1,)/sqrt(2);
        y = ifft(y,[],1);
        y = real(y*sqrt(2*n));
        I = (1:n)*0.5+0.5;
        I(2:2:end) = n-I(1:2:end-1)+1;
        y = y(I,:);
        y = reshape(y,siz);
        y = shiftdim(y,1);
    end
end

```

```

end
end

y = reshape(y,sizy);

end

%% -----
% Simplified SMOOTHN function for better clarity.
% -----
% The following function works with complex N-D arrays.

%{
function z = ezsmoothn(y)

sizy = size(y);
n = prod(sizy);
N = sum(sizy~=1);

Lambda = zeros(sizy);
d = ndims(y);
for i = 1:d
    siz0 = ones(1,d);
    siz0(i) = sizy(i);
    Lambda = bsxfun(@plus,Lambda,...
        2-2*cos(pi*(reshape(1:sizy(i),siz0)-1)/sizy(i)));
end

W = ones(sizy);
zz = y;
for k = 1:4
    tol = Inf;
    while tol>1e-3
        DCTy = dctn(W.*(y-zz)+zz);
        fminbnd(@GCVscore,-10,30);
        tol = norm(zz(:)-z(:))/norm(z(:));
        zz = z;
    end
    tmp = sqrt(1+16*s);
    h = (sqrt(1+tmp)/sqrt(2)/tmp)^N;
    W = bisquare(y-z,h);
end
function GCVs = GCVscore(p)
    s = 10^p;
    Gamma = 1./(1+s*Lambda.^2);
    z = idctn(Gamma.*DCTy);
    RSS = norm(sqrt(W(:)).*(y(:)-z(:)))^2;
    TrH = sum(Gamma(:));
    GCVs = RSS/n/(1-TrH/n)^2;
end
end

function W = bisquare(r,h)
MAD = median(abs(r(:)-median(r(:)))));

```

```
u = abs(r/(1.4826*MAD)/sqrt(1-h));  
W = (1-(u/4.685).^2).^2.*(u/4.685<1);  
end  
  
%}
```

### D.3 Método PLS

```
function deformacion = PLS9(m,displ)
%funcion recibe matriz de desplazamientos original y el valor escalar de m
%para definir ventana de cálculo de (2m+1 x 2m+1) datos

%CREAR COPIA DE MATRIZ ORIGINAL
copydispl= displ;

%OBTENER TAMAÑO MATRIZ ORIGINAL
[filas, columnas] = size(displ);

%AUMENTAR TAMAÑO DE MATRIZ ORIGINAL CON VALORES NAN
%PARA QUE VENTANA DE CÁLCULO NO EXCEDA NUNCA LIMITES DE LA MATRIZ
a= repmat(NaN(columnas,1)', [m+1,1]);
displ=cat(1,a,displ);
displ=cat(1,displ,a);
[filas, columnas] = size(displ);
a= repmat(NaN(filas,1), [1,m+1]);
displ=cat(2,a,displ);
displ=cat(2,displ,a);
[filas, columnas] = size(displ);

%CREAR MATRIZ DE CEROS QUE CONTENDRÁ LAS DEFORMACIONES FINALES
deformacion=zeros(filas,columnas);

%IDENTIFICAR VALORES DE PUNTOS DE INTERÉS VALIDOS
A=isnan(displ);

%CREAR MATRIZ DE POSICIONES M QUE DEFINAN LA POSICION DE LOS
%DESPLAZAMIENTOS DENTRO DE LA VENTANA DE CÁLCULO
count=1;

for x= 1:(2*m+1)
    for y=1:(2*m+1)
        M(count,:)=[-m-1+x,y-m-1]; % if index out of bounds =>
        count=count + 1;
    end
end

CW = zeros(2*m+1,2*m+1);

%CALCULO DE DESPLAZAMIENTO PARA CADA PUNTO DE INTERÉS EN LA MATRIZ DE
%DESPLAZAMIENTOS
for j= 1:columnas
    for i=1:filas

        Mreg = [];
        D = [];
        clearD=[];
        clearM=[];
        count=1;
```

```

if A(i,j)==0
    CW = zeros(2*m+1,2*m+1);
    CW(CW==0)=NaN;
    D = displ(i-m:i+m,j-m:j+m);
    CW = D;
    cw = CW(:);
    cwd = cat(2,M,cw);
    cwd=cwd(~any(isnan(cwd),2),:);
    %AJUSTE DE MINIMOS CUADRADOS PARA LA VENTANA DE CALCULO
    sf = fit(cwd(:,1:2),cwd(:,3),'poly11');
    C = coeffvalues(sf);
    a1 = C(2);
    deformacion(i,j)=a1;
end
end
end
%SE ELIMINAN LAS DIMENSIONES AGREGADAS EN UN PRINCIPIO PARA EVITAR ERRORES
%DE INDICES FUERA DE LOS BORDES
deformacion(1:m+1,:)=[];
deformacion(:,1:m+1)=[];
deformacion(end-(m):end,:)=[];
deformacion(:,end-(m):end)=[];
deformacion(isnan(copydispl)==1)=NaN;

end

```

## D.4 Reducción del Ruido en el Campo de Deformación, a través de Minimización de Función de Costos Regularizada

```
function [xs,ys,strain] = costoregularizada(x,y,displ,lambda)
inan    = find(isnan(displ));
%DISTANCIA EN MM ENTRE DATOS, DEBE COINCIDIR CON LA DISTANCIA ENTRE DATOS
%UTILIZADA PARA AJUSTAR LOS DESPLAZAMIENTOS EXPERIMENTALES A MATRICES
%COMPARABLES
deltax = 1;
deltay = 1;

xs=x;
ys=y;
%CALCULO DEL CAMPO DE DEFORMACIÓN POR DIFERENCIACIÓN DEL GRADIENTE
[eijn_x, eijn_y] = gradient(displ,deltax,deltay);
[rows, columns]=size(displ);
inan    = find(isnan(eijn_x));
eijn_x(inan) = 0;
%SOLUCIÓN INICIAL
eij_x = eijn_x

%INICIALIZAR VALORES PARA ITERACIÓN
dUdej=zeros(rows,columns);
[n,m]=size(eij_x);
up=zeros(n,m);
left=zeros(n,m);
down=zeros(n,m);
right=zeros(n,m);

%MÉTODO DE NEWTON RHAPSON DE 1000 ITERACIONES
for k= 1:1000
%VALORES DE LOS PUNTOS VECINOS (ARRIBA, ABAJO, IZQUIERDA, DERECHA)
up(1:n-1,1:m)    = eij_x(2:rows,:);
down(2:n,1:m)    = eij_x(1:rows-1,:);
left(1:n,1:m-1)  = eij_x(:,2:columns);
right(1:n,2:m)   = eij_x(:,1:columns-1);

%GRADIENTE DE LA FUNCIÓN DE COSTOS REGULARIZADA
dUdej = 2*lambda.*(eij_x -eijn_x) + 2*(1-lambda).*(4.*eij_x -up-down-left -
right );
%GRADIENTE DEL GRADIENTE DE LA FUNCIÓN DE COSTOS REGULARIZADA
ddUdej = 8-6*lambda;
%GUARDAR VALOR PARA PROXIMA ITERACIÓN
eij_xk = eij_x;
%CÁLCULO DE NUEVO CAMPO DE DESPLAZAMIENTO
eij_x = eij_x - (dUdej./ddUdej);
end

strain = eij_x;
xs=x;
ys=y;
end
```

## D.5 Script para cargar datos en Matlab

Luego de Importar los datos con Importar\_datos, las matrices quedan guardadas en la carpeta del archivo hdf5. Éstas, necesitan ser cargadas al ambiente matlab para poder procesarlas.

```
load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_17s5_displ = displ_x_all(:,:,2);
c1_17s5_cx= coord_x;
c1_17s5_cy= coord_y;
c1_17s5_cz= coord_z;
c1_17s5_var= var;
```

```
load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_19s5_displ = displ_x_all(:,:,2);
c1_19s5_cx= coord_x;
c1_19s5_cy= coord_y;
c1_19s5_cz= coord_z;
c1_19s5_var= var;
```

```
load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_23s5_displ = displ_x_all(:,:,2);
c1_23s5_cx= coord_x;
c1_23s5_cy= coord_y;
c1_23s5_cz= coord_z;
c1_23s5_var= var;
```

```
load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
```

```

load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_25s5_displ = displ_x_all(:, :, 2);
c1_25s5_cx= coord_x;
c1_25s5_cy= coord_y;
c1_25s5_cz= coord_z;
c1_25s5_var= var;

```

```

load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_27s5_displ = displ_x_all(:, :, 2);
c1_27s5_cx= coord_x;
c1_27s5_cy= coord_y;
c1_27s5_cz= coord_z;
c1_27s5_var= var;

```

```

load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_29s5_displ = displ_x_all(:, :, 2);
c1_29s5_cx= coord_x;
c1_29s5_cy= coord_y;
c1_29s5_cz= coord_z;
c1_29s5_var= var;

```

```

load('coord_x.mat')
load('coord_y.mat')
load('coord_z.mat')
load('displ_x_all.mat')
load('displ_y_all.mat')
load('displ_z_all.mat')
load('mask.mat')
load('var.mat')
c1_31s5_displ = displ_x_all(:, :, 2);
c1_31s5_cx= coord_x;
c1_31s5_cy= coord_y;

```

```
c1_31s5_cz= coord_z;  
c1_31s5_var= var;
```

```
load('coord_x.mat')  
load('coord_y.mat')  
load('coord_z.mat')  
load('displ_x_all.mat')  
load('displ_y_all.mat')  
load('displ_z_all.mat')  
load('mask.mat')  
load('var.mat')  
c1_33s5_displ = displ_x_all(:, :, 2);  
c1_33s5_cx= coord_x;  
c1_33s5_cy= coord_y;  
c1_33s5_cz= coord_z;  
c1_33s5_var= var;
```

```
step    = 1;  
a       = gridfit(xdisp,ydisp,ansys_displacement,-200:step:200,-  
20:step:20);  
ANSYSX  = -200:step:200;  
ANSYSY  = -20:step:20;
```

```
%timbre window 17  
[tx,ty,alf,er] = minimizar_er-  
ror(t_17s5_cx,t_17s5_cy,t_17s5_cz,t_17s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);  
[t_17s5_e,t_17s5_m,t_17s5_exp] = geter-  
ror4(t_17s5_cx,t_17s5_cy,t_17s5_cz,t_17s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-  
ror(t_17s6_cx,t_17s6_cy,t_17s6_cz,t_17s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);  
[t_17s6_e,t_17s6_m,t_17s6_exp] = geter-  
ror4(t_17s6_cx,t_17s6_cy,t_17s6_cz,t_17s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-  
ror(t_17s7_cx,t_17s7_cy,t_17s7_cz,t_17s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);  
[t_17s7_e,t_17s7_m,t_17s7_exp] = geter-  
ror4(t_17s7_cx,t_17s7_cy,t_17s7_cz,t_17s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
%timbre window 19  
[tx,ty,alf,er] = minimizar_er-  
ror(t_19s5_cx,t_19s5_cy,t_19s5_cz,t_19s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);  
[t_19s5_e,t_19s5_m,t_19s5_exp] = geter-  
ror4(t_19s5_cx,t_19s5_cy,t_19s5_cz,t_19s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_19s6_cx,t_19s6_cy,t_19s6_cz,t_19s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_19s6_e,t_19s6_m,t_19s6_exp] = geter-
ror4(t_19s6_cx,t_19s6_cy,t_19s6_cz,t_19s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_19s7_cx,t_19s7_cy,t_19s7_cz,t_19s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_19s7_e,t_19s7_m,t_19s7_exp] = geter-
ror4(t_19s7_cx,t_19s7_cy,t_19s7_cz,t_19s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
%timbre window 21
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_21s5_cx,t_21s5_cy,t_21s5_cz,t_21s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_21s5_e,t_21s5_m,t_21s5_exp] = geter-
ror4(t_21s5_cx,t_21s5_cy,t_21s5_cz,t_21s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_21s6_cx,t_21s6_cy,t_21s6_cz,t_21s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_21s6_e,t_21s6_m,t_21s6_exp] = geter-
ror4(t_21s6_cx,t_21s6_cy,t_21s6_cz,t_21s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_21s7_cx,t_21s7_cy,t_21s7_cz,t_21s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_21s7_e,t_21s7_m,t_21s7_exp] = geter-
ror4(t_21s7_cx,t_21s7_cy,t_21s7_cz,t_21s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
%timbre window 23
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_23s5_cx,t_23s5_cy,t_23s5_cz,t_23s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_23s5_e,t_23s5_m,t_23s5_exp] = geter-
ror4(t_23s5_cx,t_23s5_cy,t_23s5_cz,t_23s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_23s6_cx,t_23s6_cy,t_23s6_cz,t_23s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_23s6_e,t_23s6_m,t_23s6_exp] = geter-
ror4(t_23s6_cx,t_23s6_cy,t_23s6_cz,t_23s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_23s7_cx,t_23s7_cy,t_23s7_cz,t_23s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_23s7_e,t_23s7_m,t_23s7_exp] = geter-
ror4(t_23s7_cx,t_23s7_cy,t_23s7_cz,t_23s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
%timbre window 25
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_25s5_cx,t_25s5_cy,t_25s5_cz,t_25s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_25s5_e,t_25s5_m,t_25s5_exp] = geter-
ror4(t_25s5_cx,t_25s5_cy,t_25s5_cz,t_25s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_25s6_cx,t_25s6_cy,t_25s6_cz,t_25s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_25s6_e,t_25s6_m,t_25s6_exp] = geter-
ror4(t_25s6_cx,t_25s6_cy,t_25s6_cz,t_25s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```

[tx,ty,alf,er] = minimizar_er-
ror(t_25s7_cx,t_25s7_cy,t_25s7_cz,t_25s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_25s7_e,t_25s7_m,t_25s7_exp] = geter-
ror4(t_25s7_cx,t_25s7_cy,t_25s7_cz,t_25s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

%timbre window 27
[tx,ty,alf,er] = minimizar_er-
ror(t_27s5_cx,t_27s5_cy,t_27s5_cz,t_27s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_27s5_e,t_27s5_m,t_27s5_exp] = geter-
ror4(t_27s5_cx,t_27s5_cy,t_27s5_cz,t_27s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_27s6_cx,t_27s6_cy,t_27s6_cz,t_27s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_27s6_e,t_27s6_m,t_27s6_exp] = geter-
ror4(t_27s6_cx,t_27s6_cy,t_27s6_cz,t_27s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_27s7_cx,t_27s7_cy,t_27s7_cz,t_27s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_27s7_e,t_27s7_m,t_27s7_exp] = geter-
ror4(t_27s7_cx,t_27s7_cy,t_27s7_cz,t_27s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
%timbre window 29
[tx,ty,alf,er] = minimizar_er-
ror(t_29s5_cx,t_29s5_cy,t_29s5_cz,t_29s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_29s5_e,t_29s5_m,t_29s5_exp] = geter-
ror4(t_29s5_cx,t_29s5_cy,t_29s5_cz,t_29s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_29s6_cx,t_29s6_cy,t_29s6_cz,t_29s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_29s6_e,t_29s6_m,t_29s6_exp] = geter-
ror4(t_29s6_cx,t_29s6_cy,t_29s6_cz,t_29s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_29s7_cx,t_29s7_cy,t_29s7_cz,t_29s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_29s7_e,t_29s7_m,t_29s7_exp] = geter-
ror4(t_29s7_cx,t_29s7_cy,t_29s7_cz,t_29s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

%timbre window 31
[tx,ty,alf,er] = minimizar_er-
ror(t_31s5_cx,t_31s5_cy,t_31s5_cz,t_31s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_31s5_e,t_31s5_m,t_31s5_exp] = geter-
ror4(t_31s5_cx,t_31s5_cy,t_31s5_cz,t_31s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_31s6_cx,t_31s6_cy,t_31s6_cz,t_31s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_31s6_e,t_31s6_m,t_31s6_exp] = geter-
ror4(t_31s6_cx,t_31s6_cy,t_31s6_cz,t_31s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

[tx,ty,alf,er] = minimizar_er-
ror(t_31s7_cx,t_31s7_cy,t_31s7_cz,t_31s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_31s7_e,t_31s7_m,t_31s7_exp] = geter-
ror4(t_31s7_cx,t_31s7_cy,t_31s7_cz,t_31s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);

%timbre window 33

```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_33s5_cx,t_33s5_cy,t_33s5_cz,t_33s5_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_33s5_e,t_33s5_m,t_33s5_exp] = geter-
ror4(t_33s5_cx,t_33s5_cy,t_33s5_cz,t_33s5_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_33s6_cx,t_33s6_cy,t_33s6_cz,t_33s6_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_33s6_e,t_33s6_m,t_33s6_exp] = geter-
ror4(t_33s6_cx,t_33s6_cy,t_33s6_cz,t_33s6_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

```
[tx,ty,alf,er] = minimizar_er-
ror(t_33s7_cx,t_33s7_cy,t_33s7_cz,t_33s7_displ,ANSYSX,ANSYSY,a,[0,0,0]);
[t_33s7_e,t_33s7_m,t_33s7_exp] = geter-
ror4(t_33s7_cx,t_33s7_cy,t_33s7_cz,t_33s7_displ,tx,ty,alf,ANSYSX,ANSYSY,a);
```

## D.6 Función ajuste del origen sistema de coordenadas minimizando desviación estandar para vectores de traslación $tx,ty$ y de rotación $alf$ .

```
function [tx,ty,alf,FVAL] = minimizar_error(coord_x,coord_y,coord_z,M,AN-
SYSX,ANSYSY,ANSYS,x0)
%M corresponde a la matriz de desplazamientos que entrega el software DIC

cx = clearzero(coord_x);
cy = clearzero(coord_y);
cz = clearzero(coord_z);
N = clearnan(M); % Datos de ISTRa vienen reflejados en eje x
[aux1 aux2] = size(cx);
%Girar sistema de coordenadas en torno a Y
catop = abs(cx(1,1)-cx(1,aux2));
catady = abs(cz(1,1)-cz(1,aux2));
beta2= (pi/2) -atan(catop/catady);
CX=coord_x.*cos(beta2)+coord_z.*sin(beta2);
CZ=-coord_x.*sin(beta2)+coord_z.*cos(beta2);
cx = clearzero(CX);
cz = clearzero(CZ);
%Girar sistema de coordenadas en torno a X
catady = abs(cy(aux1,1)-cy(1,1));
catop = abs(cz(aux1,1)-cz(1,1));
beta=- atan(catop/catady);
CY=coord_y.*cos(beta)-CZ.*sin(beta);
CZ=-coord_y.*sin(beta)+CZ.*cos(beta);

x0=[x0(1),x0(2),x0(3)];
options = optimset('Display','iter','Algorithm','sqp');
[X,FVAL] = fminsearchbnd(@nestedfun,x0,[-400,-100,-pi/6],[400,100,pi/6],op-
tions);
% Nested function that computes the objective function
function y = nestedfun(X)
```

```
xx      =      (CX-X(1)).*cos(X(3))-(CY-X(2)).*sin(X(3));
yy      =      (CX-X(1)).*sin(X(3))+(CY-X(2)).*cos(X(3));
experimental = griddata(xx,yy,M,ANSYSX',ANSYSY);
A= abs(ANSYS-experimental);
B = reshape( A ,numel(A),1);
y = nanstd(B)

end
tx=X(1);
ty=X(2);
alf=X(3);
end
```

## D.7 Función de obtención de error promedio de desplazamientos

```
function [errormatrix,devstd,experimental] = geterror4(coord_x,coord_y,coord_z,M,tx,ty,alf,ANSYSX,ANSYSY,ANSYS)
cx = clearzero(coord_x);
cy = clearzero(coord_y);
cz = clearzero(coord_z);
N = clearnan(M); %
[aux1 aux2] = size(cx);
%Girar sistema de coordenadas en torno a Y
catop = abs(cx(1,1)-cx(1,aux2));
catady = abs(cz(1,1)-cz(1,aux2));
beta1= (pi/2) -atan(catop/catady);
CX=coord_x.*cos(beta1)+coord_z.*sin(beta1);
CZ=-coord_x.*sin(beta1)+coord_z.*cos(beta1);
cx = clearzero(CX);
cz = clearzero(CZ);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%CORRECCIÓN DE LAS DISTORSIONES DE PERSPECTIVA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Girar sistema de coordenadas en torno a X
catady = abs(cy(aux1,1)-cy(1,1));
catop = abs(cz(aux1,1)-cz(1,1));
beta=- atan(catop/catady);
CY=coord_y.*cos(beta)-CZ.*sin(beta);
CZ=-coord_y.*sin(beta)+CZ.*cos(beta);
%Girar sistema de coordenadas en plano XY según Alfa

CX = (CX).*cos(alf)-(CY).*sin(alf);
CY = (CX).*sin(alf)+(CY).*cos(alf);
%Centrar datos segun vector de traslación tx,ty
CX = CX-tx;
CY = CY-ty;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ajustar matrices a datos ANSYS
experimental = griddata(CX,CY,M,ANSYSX',ANSYSY,'linear');
errormatrix = abs(ANSYS-experimental);
B = reshape(errormatrix,numel(errormatrix),1);
devstd = nanstd(B,1);
```

## D.8 Scripts para obtención de gráficos

```
% GRAFICOS FINALES INFORMES
%DESPLAZAMIENTOS

ventanas2 =[17,19,21,23,25,27,29,31,33];
step      = 0.5;
a         = gridfit(xdisp,ydisp,ansys_displacement,-200:step:200,-
20:step:20);
ANSYSX   = -200:step:200;
ANSYSY   = -20:step:20;
%%Probets Cepillo
for k = 1:length(ventanas)
%     for k = 1:2
f2 = figure('position',[900 250 900 250]);
hold on
surf(ANSYSX,ANSYSY,Cexpstep5(:, :, k));
ylabel('Posición y [mm]', 'FontSize',14)
xlabel('Posición x [mm]', 'FontSize',14)
caxis([-0.048 0.048])
c=colorbar;
ylabel(c, 'u_x[mm]', 'FontSize',14)
set(gca, 'FontSize', 14)
xlim([-105,108])
ylim([-60,60])
view(2)
shading interp
shading interp
%datos error abs

graphtitle=['\1_C1_', num2str(ventanas(k)), '_s5_e'];
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes',graphtitle, '.jpg');
saveas(f2,filename, 'jpg');

f3 = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,Cerrstep5(:, :, k));
ylabel('Posición y [mm]', 'FontSize',14)
xlabel('Posición x [mm]', 'FontSize',14)
caxis([0 0.01])
c=colorbar;
ylabel(c, '|u_x^{DIC} - u_x^{ANSYS}| [mm]', 'FontSize',14)

grid
xlim([-105,108])
ylim([-60,60])
view(2)
shading interp
set(gca, 'FontSize', 14)

% % Save Results
% % .fig Format
```

```

% graphtitle=['SLD2 ',test(Test,:), '_ ',mate-
rial(mi,:), 'Amp', num2str(Amp*1e6), 'mum'];
% filename=strcat(path,graphtitle, '.fig');
% saveas(f2,filename, 'fig');
% Jpeg format

graphtitle=['\1_C1_', num2str(ventanas(k)), '_s5_error'];
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', graphtitle, '.jpg');
saveas(f3,filename, 'jpg');
end
ventanas2 =[17,19,21,23,25,27,29,31,33];
%errorbar
fe = figure('position', [900 250 900 250]);
plot(ventanas2,meancep5, '-s')
hold on
plot(ventanas2,meancep5+stdcep5, '--r')
plot(ventanas2,meancep5-stdcep5, '--r')
set(gca, 'XTick', ventanas2)
xlabel('Tamaño de Subset [pixeles]', 'FontSize', 14)
ylabel('Error Promedio [mm]', 'FontSize', 14)
legend('Error Promedio [mm]', 'Limites de confiabilidad')
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\errorbardisplcepbre2', '.jpg');
saveas(fe,filename, 'jpg');

%%Probets Timbre
step = 1;
a = gridfit(xdisp, ydisp, ansys_displacement, -200:step:200, -20:step:20);
ANSYSX = -200:step:200;
ANSYSY = -20:step:20;

for k = 1:length(ventanas)
% for k = 1:2
f2 = figure('position', [900 250 900 250]);
hold on
surf(ANSYSX, ANSYSY, Texpstep5(:, :, k));
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([-0.048 0.048])
c=colorbar;
ylabel(c, 'u_x[mm]', 'FontSize', 14)
set(gca, 'FontSize', 14)
xlim([-105, 108])
ylim([-60, 60])
view(2)
shading interp
shading interp
%datos error abs

graphtitle=['\1_T_', num2str(ventanas(k)), '_s5_e'];
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', graphtitle, '.jpg');
saveas(f2,filename, 'jpg');

```

```

f3 = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,Terrstep5(:,:,k));
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([0 0.01])
c=colorbar;
ylabel(c,'|u_x^{DIC} - u_x^{ANSYS}| [mm]','FontSize',14)

grid
xlim([-105,108])
ylim([-60,60])
view(2)
shading interp
set(gca,'FontSize', 14)

%% Save Results
%% .fig Format
% graphtitle=['SLD2 ',test(Test,:),'_',mate-
% rial(mi,:), 'Amp',num2str(Amp*1e6), 'mum'];
% filename=strcat(path,graphtitle,'.fig');
% saveas(f2,filename,'fig');
% Jpeg format

graphtitle=['\1_T ',num2str(ventanas(k)),'_s5_error'];
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes',graphtitle,'.jpg');
saveas(f3,filename,'jpg');
end
ventanas2 =[17,19,21,23,25,27,29,31,33];
%errorbar
fe = figure('position',[900 250 900 250]);
plot(ventanas2,meantim5,'-s')
hold on
plot(ventanas2,meantim5+stdtim5,'--r')
plot(ventanas2,meantim5-stdtim5,'--r')
set(gca,'XTick',ventanas2)
xlabel('Tamaño de Subset [pixeles]','FontSize',14)
ylabel('Error Promedio [mm]','FontSize',14)
legend('Error Promedio [mm]','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\errorbardispltim-
bre2','.jpg');
saveas(fe,filename,'jpg');

%%
%DEFORMACIONES
%DEFORMACION ORIGINAL
f4 = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,FX)

```

```

view(2)
shading interp
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
xlim([-105, 108])
ylim([-60, 60])
caxis([-0.0011 0.0008])
c=colorbar;
ylabel(c, '\epsilon_x')
grid
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\grad-despl-sin-
suav', '.jpg');
saveas(f4, filename, 'jpg');

[FX, FY] = gradient(t_27s5_exp, ANSYSX, ANSYSY);
error_original_promedio = nanmean(nanmean(abs(FX-ANSYS_strain)));

B= reshape(abs(FX-ANSYS_strain), numel(abs(FX-ANSYS_strain)), 1)
std_original = nanstd(B, 1)

%def sin smooth
fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, abs(FX-ANSYS_strain));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([0, 0.001])
c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_error_origi-
nal', '.jpg');
saveas(fc, filename, 'jpg');

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%CALCULO GRÁFICOS DEFORMACIONES SMOOTH
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GRAFICOS DE CONTORNO DE DESPLAZAMIENTOS%
fc = figure('position', [900 250 900 250]);
contour(ANSYSX, ANSYSY, t_27s5_exp, 18)
xlim([-105, 108])
view(2)
ylim([-21, 21])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis('auto')

```

```

c=colorbar
ylabel(c,'u_x [mm]','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\contour_dis-
pl','.jpg');
saveas(fc,filename,'jpg');

```

```

%GRAFICOS DE CONTORNO DE DESPLAZAMIENTOS con S=1%

```

```

fc = figure('position',[900 250 900 250]);
contour(ANSYSX,ANSYSY,smoothn(t_27s5_exp,1),18)
xlim([-105,108])
view(2)
ylim([-21,21])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis('auto')
c=colorbar
ylabel(c,'u_x [mm]','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\con-
tour_displ_s1','.jpg');
saveas(fc,filename,'jpg');

```

```

%GRAFICOS DE CONTORNO DE DESPLAZAMIENTOSS=10%

```

```

fc = figure('position',[900 250 900 250]);
contour(ANSYSX,ANSYSY,smoothn(t_27s5_exp,1000),18)
xlim([-105,108])
view(2)
ylim([-21,21])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis('auto')
c=colorbar
ylabel(c,'u_x [mm]','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\con-
tour_displ_s1000','.jpg');
saveas(fc,filename,'jpg');

```

```

%GRAFICOS DE CONTORNO DE DESPLAZAMIENTOSS=100%

```

```

fc = figure('position',[900 250 900 250]);
contour(ANSYSX,ANSYSY,smoothn(t_27s5_exp,100),18)
xlim([-105,108])
view(2)
ylim([-21,21])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis('auto')
c=colorbar
ylabel(c,'u_x [mm]','FontSize',14)
set(gca,'FontSize', 14)

```

```

filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\con-
tour_displ_s100','.jpg');
saveas(fc,filename,'jpg');

%GRAFICOS DE surf y error DE DESPLAZAMIENTOS con S=1%
aux = find(S>=1);
pos = aux(1);
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,DefSmooth(:, :, pos));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([-0.001064 0.0007887])
c=colorbar
grid
ylabel(c,'\epsilon_x','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes','\t27s5_smooth_s1','.jpg');
saveas(fc,filename,'jpg');

fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,abs(DefSmooth(:, :, pos)-ANSYS_strain));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([0,0.001])
c=colorbar
grid
ylabel(c,'\|\epsilon_x^{\{DIC\}} - \epsilon_x^{\{ANSYS\}} | ','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes','\t27s5_errsmooth_s1','.jpg');
saveas(fc,filename,'jpg');

%GRAFICOS DE surf y error DE DESPLAZAMIENTOS con S=10%
aux = find(S>=10);
pos = aux(1);
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,smoothn(DefSmooth(:, :, pos)));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([-0.001064 0.0007887])
c=colorbar

```

```

grid
ylabel(c, '\epsilon_x', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_smooth_s10', '.jpg');
saveas(fc, filename, 'jpg');

fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, abs(DefSmooth(:, :, pos) - ANSYS_strain));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([0, 0.001])
c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} ', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_errsmooth_s10', '.jpg');
saveas(fc, filename, 'jpg');

```

```

%GRAFICOS DE surf y error DE DESPLAZAMIENTOS con S=100%
aux = find(S>=100);
pos = aux(1);
fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, DefSmooth(:, :, pos));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([-0.001064 0.0007887])
c=colorbar
grid
ylabel(c, '\epsilon_x ', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_smooth_s100', '.jpg');
saveas(fc, filename, 'jpg');
%SURF ERROR S=100
fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, abs(DefSmooth(:, :, pos) - ANSYS_strain));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([0, 0.001])

```

```

c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize',14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes', '\t27s5_errsmooth_s100', '.jpg');
saveas(fc,filename, 'jpg');

%GRAFICOS DE surf y error DE DESPLAZAMIENTOS con S=1000%
aux = find(S>=1000);
pos = aux(1);
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,DefSmooth(:, :, pos));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]', 'FontSize',14)
xlabel('Posición x [mm]', 'FontSize',14)
caxis([-0.001064 0.0007887])
c=colorbar
grid
ylabel(c, '\epsilon_x ', 'FontSize',14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes', '\t27s5_smooth_s1000', '.jpg');
saveas(fc,filename, 'jpg');
%SURF ERROR S=100
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,abs(DefSmooth(:, :, pos)-ANSYS_strain));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]', 'FontSize',14)
xlabel('Posición x [mm]', 'FontSize',14)
caxis([0,0.001])
c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize',14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes', '\t27s5_errsmooth_s1000', '.jpg');
saveas(fc,filename, 'jpg');

%errorbar
f6 = figure('position',[900 250 900 250]);
plot(S, Smoothmean, '.')
hold on
plot(S, Smoothmean + SmoothDevstd, '--r')
plot(S, Smoothmean - SmoothDevstd, '--r')

```

```

xlabel('Factor de suavizado S','FontSize',14)
ylabel('Error Promedio','FontSize',14)
legend('Error Promedio','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\er-
rorbarsmooth2','.jpg');
saveas(f6,filename,'jpg');

%ERRORBAR corregido

f6 = figure('position',[900 250 900 250]);
plot(S,Smoothmean_f, '.')
hold on
plot(S,Smoothmean_f + SmoothDevstd_f, '--r')
plot(S,Smoothmean_f - SmoothDevstd_f, '--r')

xlabel('Factor de suavizado S','FontSize',14)
ylabel('Error Promedio','FontSize',14)
legend('Error Promedio','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\er-
rorbarsmooth_crregido','.jpg');
saveas(f6,filename,'jpg')

%%
%PLS

%%GRAFICOS PARA DISTINTAS VENTANAS

for k = 1:length(Windows)
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,DefPLS(:, :, k))
view(2)
shading interp
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
xlim([-105,108])
ylim([-60,60])
caxis([-0.001064 0.0007887])
view(2)
shading interp
c=colorbar;
ylabel(c, '\epsilon_x ', 'FontSize',14)
set(gca,'FontSize', 14)
grid
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes','\DefSmooth_M_', num2str(2.*Windows(k)+1), '.jpg');
saveas(fc,filename,'jpg');

fe = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,abs(DefPLS(:, :, k)-ANSYS_strain));
view(2)
shading interp

```

```

ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
xlim([-105, 108])
ylim([-60, 60])
view(2)
shading interp
caxis([0, 0.001])

c=colorbar;
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize', 14)

set(gca, 'FontSize', 14)
grid
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imágenes', '\DefSmoother-
ror_M_', num2str(2.*windows(k)+1), '.jpg');
saveas(fe, filename, 'jpg');
end

%errorbar
f6 = figure('position', [900 250 900 250]);
plot(windows.*2+1, PLSmean, '.');
hold on
plot(windows.*2+1, PLSmean + PLSDevstd, '--r')
plot(windows.*2+1, PLSmean - PLSDevstd, '--r')

xlabel('Tamaño de Ventana de Cálculo', 'FontSize', 14)
ylabel('Error Promedio', 'FontSize', 14)
legend('Error Promedio', 'Limites de confiabilidad')
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imágenes', '\er-
rorbarsPLSh', '.jpg');
saveas(f6, filename, 'jpg');

%%
%COSTO
%GRAFICOS DE surf y error DE alpha = 0.01
aux = find(alf>=0.01);
pos = aux(1);
fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, StrainCOST(:, :, pos));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([-0.001064 0.0007887])
c=colorbar
grid

```

```

ylabel(c, '\epsilon_x', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_cost_a001', '.jpg');
saveas(fc, filename, 'jpg');

fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, abs(StrainCOST(:, :, pos) - ANSYS_strain));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([0, 0.001])
c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_err-
cost_a001', '.jpg');
saveas(fc, filename, 'jpg');

%GRAFICOS DE surf y error DE alpha = 0.1
aux = find(alf >= 0.1);
pos = aux(1);
fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, StrainCOST(:, :, pos));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([-0.001064 0.0007887])
c=colorbar
grid
ylabel(c, '\epsilon_x', 'FontSize', 14)
set(gca, 'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes', '\t27s5_cost_a01', '.jpg');
saveas(fc, filename, 'jpg');

fc = figure('position', [900 250 900 250]);
surf(ANSYSX, ANSYSY, abs(StrainCOST(:, :, pos) - ANSYS_strain));
xlim([-105, 108])
view(2)
shading interp
ylim([-61, 61])
ylabel('Posición y [mm]', 'FontSize', 14)
xlabel('Posición x [mm]', 'FontSize', 14)
caxis([0, 0.001])
c=colorbar
grid
ylabel(c, '\epsilon_x^{DIC} - \epsilon_x^{ANSYS} |', 'FontSize', 14)
set(gca, 'FontSize', 14)

```

```
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\t27s5_err-
cost_a01','.jpg');
saveas(fc,filename,'jpg');
```

```
%GRAFICOS DE surf y error DE DESPLAZAMIENTOS conlpha = 0.3
```

```
aux = find(alf>=0.3);
pos = aux(1);
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,StrainCOST(:, :, pos));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([-0.001064 0.0007887])
c=colorbar
grid
ylabel(c,'\epsilon_x ','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desk-
top\ME6909\Imagenes','\t27s5_cost_a03','.jpg');
saveas(fc,filename,'jpg');
%SURF ERROR S=1lpha = 0.3
fc = figure('position',[900 250 900 250]);
surf(ANSYSX,ANSYSY,abs(StrainCOST(:, :, pos)-ANSYS_strain));
xlim([-105,108])
view(2)
shading interp
ylim([-61,61])
ylabel('Posición y [mm]','FontSize',14)
xlabel('Posición x [mm]','FontSize',14)
caxis([0,0.001])
c=colorbar
grid
ylabel(c,'|\epsilon_x^{\{DIC\}} - \epsilon_x^{\{ANSYS\}} |','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\t27s5_err-
cost_a03','.jpg');
saveas(fc,filename,'jpg');
```

```
%errorbar smooth
```

```
f6 = figure('position',[900 250 900 250]);
plot(alf,costmean, '.')
hold on
plot(alf,costmean + devstd, '--r')
plot(alf,costmean- devstd, '--r')
```

```

xlabel('Factor de suavizado \alpha','FontSize',14)
ylabel('Error Promedio','FontSize',14)
legend('Error Promedio','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\er-
rorbarsmooth2','.jpg');
saveas(f6,filename,'jpg');

figure
valor=0.3;
Faind = find(alf>=valor);
pos = Faind(1);
surf(ANSYSX,ANSYSY,StrainCOST(:, :, pos))
view(2)
shading interp
xlabel('Posición x [mm]')
ylabel('Posición y [mm]')
xlim([-105,108])
ylim([-60,60])
caxis([-0.001064 0.0007887])
view(2)
shading interp
title({'REDUCCIÓN DEL RUIDO POR FUNCIÓN DE COSTOS REGULARIZADA';'Distribución
de deformación normal horizontal \epsilon_x'})
c=colorbar;
figure
surf(ANSYSX,ANSYSY,abs(StrainCOST(:, :, pos)-ANSYS_strain))
view(2)
xlabel('Posición x [mm]')
ylabel('Posición y [mm]')
xlim([-105,108])
ylim([-60,60])

view(2)
shading interp
caxis([0,0.001])
title({'REDUCCIÓN DEL RUIDO POR FUNCIÓN DE COSTOS REGULARIZADA';'Error |\ep-
silon_E_X_P - \epsilon_P_L_S|'})
c=colorbar;

%errorbar
f6 = figure('position',[900 250 900 250]);
plot(alf,costmean, '.')
hold on
plot(alf,costmean + devstd, '--r')
plot(alf,costmean- devstd, '--r')

xlabel('\alpha','FontSize',14)
ylabel('Error Promedio','FontSize',14)
legend('Error Promedio','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\er-
rorbarscost','.jpg');
saveas(f6,filename,'jpg');

```

```

%errorbarcosta
fe = figure('position',[900 250 900 250]);
plot(ventanas2,meancep5,'-s')
hold on
plot(ventanas2,meancep5+stdcep5,'--r')
plot(ventanas2,meancep5-stdcep5,'--r')
set(gca,'XTick',ventanas2)
xlabel('Tamaño de Subset [pixeles]','FontSize',14)
ylabel('Error Promedio [mm]','FontSize',14)
legend('Error Promedio [mm]','Limites de confiabilidad')
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\errorbardisplcep-
bre2','.jpg');
saveas(fe,filename,'jpg');

```

```
%%
```

```

%errorbar resumen
f9 = figure('position',[900 250 900 250]);
%sin suavizado
errorbar(0,2.549*10^-4,2.235*10^-4,'s')
hold on
%smoothing
errorbar(1,4.9631e-05,4.8658e-05,'ms')
%smoothing corregido
errorbar(2,2.841531*10^-5,2.9175*10^-5,'rs')
%PLS
errorbar(3,4.97*10^-5,4.91*10^-5,'gs')
%costo
errorbar(4,7.066*10^-5,7.229*10^-5,'ks')
% Add title and axis labels
xlim([-1 5])
xlabel('Métodos de Obtención del Campo Deformación','FontSize',14)
ylabel('Error promedio','FontSize',14)
box on
% Change the labels for the tick marks on the x-axis
nombres = {'Gradiente', 'Smoothn.m', 'Smoothn.m', 'PLS.m', 'Costo.m'};
set(gca, 'XTick', 0:4, 'XTickLabel', nombres)
set(gca, 'YTick', 0:5e-05:5e-04)
ylim([-8e-06 5e-04])
% Create labels for the legend
irisMeas = {'Gradiente de Desplazamientos sin Suavizar', 'Suavizado para s=
91.81 ', 'Suavizado con Corrección para s= 848.44 ', 'Método PLS para Ventana
de 17x17', 'Método de "Función de Costo" para \alpha = 0.0806'};
legend(irisMeas, 'Location', 'Northeast')
xlabel('\alpha','FontSize',14)
ylabel('Error Promedio','FontSize',14)
set(gca,'FontSize', 14)
filename=strcat('C:\Users\Edgar\Desktop\ME6909\Imagenes','\errorbarre-
sumenst','.jpg');
saveas(f9,filename,'jpg');

```

