



Facultado de Ciencias

**Trabajo Fin de Carrera  
Licenciatura en Físicas**

**GENERACIÓN DE MÁSCARAS PARA  
PROCESADO DE SPECKLE A ALTA VELOCIDAD  
APLICADO A LA DETECCIÓN DE  
ULTRASONIDOS**

**Autor:** Lucía Ruisoto Palomera

**Director:** Adolfo Cobo

**Fecha:** Septiembre-2017

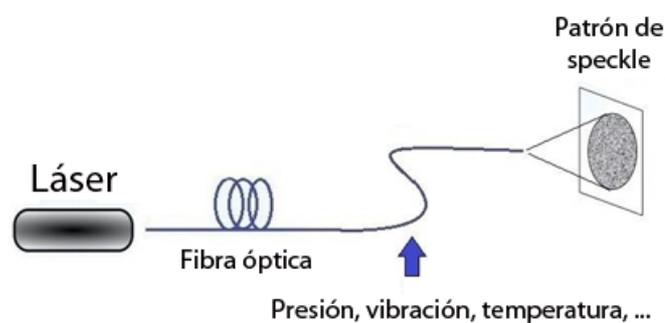
# Índice del proyecto

<b>1. Objetivo .....</b>	<b>2</b>
<b>2. Speckle y fibras ópticas .....</b>	<b>3</b>
2.1 Introducción .....	3
2.2 Speckle al iluminar una superficie rugosa .....	3
2.3 Conclusiones .....	4
2.4 Speckle en fibras multimodo .....	5
2.5 Concepto de modo.....	6
2.6 Ruido Modal .....	7
<b>3. Evaluación de diferentes máscaras .....</b>	<b>9</b>
3.1 Máscara con forma de tablero de ajedrez .....	10
3.2 Máscara circular .....	15
3.3 Máscara con forma de corona circular .....	18
<b>4. Conclusiones y líneas futuras .....</b>	<b>23</b>
<b>5. Referencias .....</b>	<b>23</b>

## 1. Objetivo

Se puede observar un patrón de speckle cuando un haz de luz coherente, como la de un láser, se propaga por una fibra óptica multimodo y se produce por la interferencia de los modos propagados. Este patrón de speckle, es muy sensible a cualquier alteración que se produzca en la fibra (tensión, vibración, torsión, temperatura, ...) y en el medio ambiente, alteraciones que producen fluctuaciones de la distribución espacial de los puntos brillantes del speckle.

La forma tradicional de obtener un patrón de speckle, es utilizando una cámara digital CCD con un montaje similar al que se muestra en la Figura 1.



**Figura 1.** Montaje para obtener un patrón de speckle al someter a la fibra multimode a una perturbación

De forma que con el procesamiento de estos patrones de speckle podemos obtener una señal que varía de forma similar a la alteración que se aplica en la fibra actuando a modo de sensor. En este trabajo lo que se pretende es comprobar si es posible diseñar una máscara empleando Matlab que junto con un fotodiodo permita reemplazar la cámara CCD y obtener esta correspondencia entre la perturbación en la fibra y de la intensidad detectada en el fotodiodo. Estas máscaras serán diseñadas de forma que tengan zonas que dejen pasar totalmente la luz y zonas opacas.

Una de las ventajas de utilizar el fotodiodo con máscara en lugar de la cámara CCD como sistema de detección, es que una cámara CCD simple presenta el problema de que no permite estudiar alteraciones en la fibra óptica que sean muy rápidas, tales como vibraciones, ya que únicamente permite capturar del orden de unos 50 fotogramas o frames por segundo (tasa de refresco) mientras que el fotodiodo es capaz de medir la intensidad con una frecuencia considerablemente mayor (orden de GHz). Otra ventaja de este sistema, es su bajo coste, ya que los fotodiodos son más baratos que cámaras que midan un alto número de fotogramas por segundo y una vez conocido el diseño óptimo de la máscara podemos construirla nosotros mismos.

## 2. Speckle y fibras ópticas

### 2.1 Introducción

Se podría decir que la primera persona que observó el speckle es Newton a partir del centello de las estrellas. Durante los años 60, el descubrimiento del láser produjo un tremendo impacto en el mundo de la óptica y sus técnicas, promoviendo el estudio de fenómenos de interferencia de luz dispersada por anillos de Newton y en los anillos de difracción de Fraunhofer que se producen cuando la luz coherente es difractada por placas de vidrio cubiertas con pequeñas partículas. En un principio el fenómeno speckle fue considerado como ruido, un efecto indeseado que afectaba la exactitud de las medidas, por lo que se intentó eliminar, aunque más tarde se averiguó que era también una gran fuente de información y se empezaría a encontrarle aplicaciones prácticas. Gracias a ello se crearon diferentes áreas de aplicación y estudio de este fenómeno, algunas como medición de rugosidades superficiales, determinación de estadísticas, procesamiento de imágenes y aplicaciones en la medicina, astronomía o la física entre otras.

El speckle es un fenómeno óptico que se puede producir de dos maneras, el primero iluminando una superficie rugosa y el segundo inyectando luz a través de una fibra óptica multimodo, este último será el que se ha utilizado para este proyecto.

### 2.2 Speckle generado por superficie rugosa

El speckle es un fenómeno óptico que se obtiene cuando se ilumina una superficie rugosa mediante luz coherente o cuando se transmite a través de un objeto difusor, en cuyo caso se observa patrones de interferencia en forma de granos, esto se puede entender viendo la Figura 2 donde se ve que las ondas que parten de la fuente de luz, interfieren constructiva o destructivamente entre ellas en cada plano de observación que es alcanzado por la luz, formando un patrón aleatorio de puntos brillantes y oscuros.

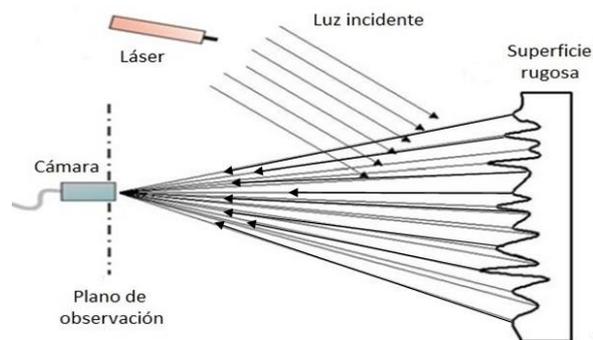
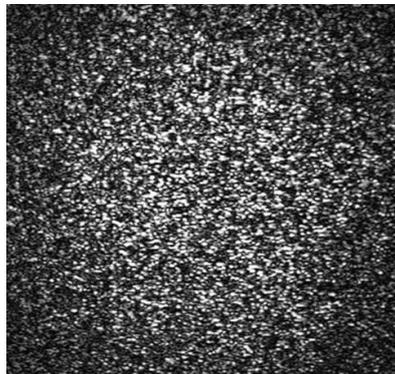


Figura 2: Formación de un patrón de speckle

Si los objetos dentro del plano se mueven, los granos del patrón también se mueven y cambian de forma, por ello el patrón de speckle depende del tiempo. Cuando se habla de coherencia entre dos ondas, se refiere a la condición necesaria para que se produzca interferencia entre ambas. En el caso de ondas de luz, esta coherencia conlleva que las ondas tienen la misma frecuencia, polarización, fase inicial y origen (fuente de luz). De aquí en adelante la coherencia estará implícita en todos los análisis que se realicen.

Dicha estructura granular no parece seguir ni patrón determinado ni tener relación alguna con las propiedades del objeto que se ilumina, aunque en los últimos tiempos se han desarrollado técnicas que utilizan el patrón de speckle como fuente de información haciendo uso de fotogramas y capturas de video sobre ello para un posterior análisis Figura 3.



**Figura 3:** Patrón de speckle tomado con una cámara CCD

La medida de estos patrones de speckle es función de la longitud de onda de la luz, el ancho del haz que ilumina la superficie que estamos estudiando y la distancia de ésta a la pantalla donde se forman los mismos. La distribución de speckle generada por un objeto rugoso ocupa todo el espacio que atraviesa la luz difundida y se puede observar interponiendo una pantalla (speckle objetivo) o visualizando la imagen que se obtiene mediante un sistema óptico (speckle subjetivo).

### **2.3 Condiciones**

Para poder explicar las condiciones necesarias para que se produzca el fenómeno de speckle, realizamos a continuación una pequeña introducción del concepto de la luz. La luz se define como una onda electromagnética compuesta por fotones, cuya frecuencia y energía determinan la longitud de onda de un color, el cual puede ser percibido por el ojo humano. Esta se compone de tres características principales que son longitud de onda, frecuencia y velocidad. Dentro de los fenómenos ópticos que se pueden producir gracias a estas características, ahí dos muy útiles como son el modelo cuántico y el modelo ondulatorio.

En nuestro caso nos centraremos en el modelo ondulatorio, ya que nos permite explicar dos fenómenos muy característicos como son el fenómeno de coherencia y la interferencia.

- Coherencia: Se dice que dos puntos de una onda son coherentes cuando guardan una relación de fase constante, es decir cuando conocido el valor instantáneo del campo eléctrico en uno de los puntos, es posible predecir el del otro.
- Interferencia: Se denomina interferencia a la superposición coherente, tanto espacial como temporal, de dos o más ondas en un punto dado formando una onda resultante de mayor o menor amplitud.

## 2.4 Speckle en fibras multimode

Existen dos tipos de fibras, las fibras monomodos y las fibras multimodos, para este proyecto se han utilizado estas últimas. Las fibras multimodos son fibras que permiten el paso de varios haces de luz, también conocidos como modos, a través del núcleo que se reflejan con distintos ángulos y diferentes velocidades de fase. Este tipo de fibras tienen el núcleo (core) con un diámetro mucho mayor que las fibras monomodo donde los rayos viajan variando distancias, dependiendo del ángulo de propagación y como consecuencia sufren diferentes retardos de fase pasando desde la entrada hasta la salida de la guía. En el interior de la fibra se producen dos fenómenos como son la interferencia entre modos y la iteración entre el propio modo Figura 4.

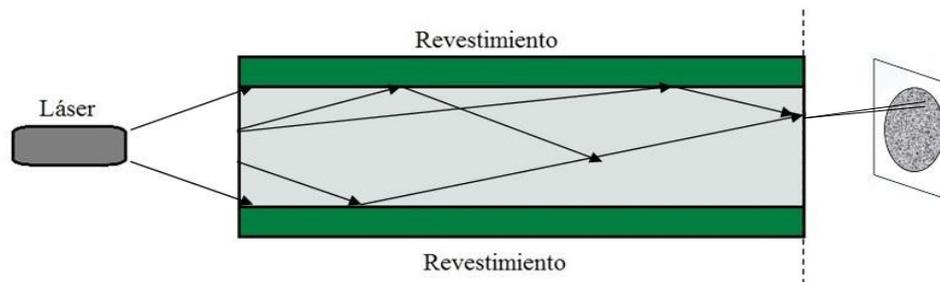


Figura 4: Fenómeno de speckle en fibra óptica

Si se hace un análisis electromagnético más preciso de la guía mostrara que existen diferentes posibilidades de modos de propagación, y estos modos se propagan con diferentes velocidades de fase. Siguiendo este argumento, la luz que llega al punto de la cara exterior de la guía está formada por la suma de múltiples contribuciones lumínicas individuales. Si el retardo de fase sufrido por esas contribuciones varía más de  $2\pi$ , y si la fuente es suficientemente coherente entonces, los efectos producidos por la estructura se observarán como una distribución de intensidades a través del final de la guía.

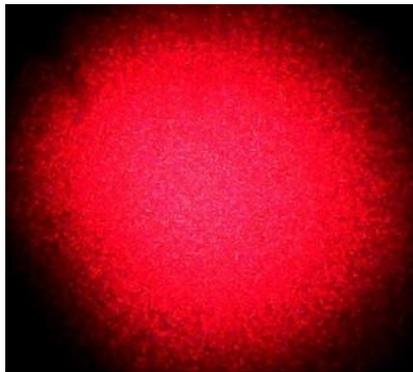


Figura 5: Patrón de speckle obtenido con fibra multimodo

## 2.5 Concepto de Modo

En cuanto a los modos ópticos propagados por la fibra existen dos formas de describirlos, una de ellas utiliza la teoría de rayos, en la que se considera que la longitud de onda es cero, es decir, que la luz es representada por un rayo, por lo que un modo es la agrupación de un conjunto de rayos que tienen similares caminos ópticos de la fibra. La otra forma utiliza la teoría electromagnética, es decir, que la luz es considerada como una onda, también es llamada teoría modal, donde el modo óptico es una distribución de energía en la sección transversal de la fibra considerando una onda estacionaria.

Para explicar el concepto de modo utilizaremos la teoría modal, que se utiliza para describir el comportamiento de la luz en el interior de una fibra óptica. Esta teoría permite explicar las propiedades que no es capaz de resolver la teoría de rayos. El conjunto de ondas electromagnéticas guiadas por el interior de la fibra se les denomina modos. La teoría modal sugiere que la luz puede representarse como una onda plana, que se caracteriza por su dirección, amplitud y longitud de onda. Se trata de una onda de frecuencia constante cuyos frentes de onda (superficie con fase constante) son planos paralelos infinitos normales a la dirección de propagación. Se calcula a partir de las ecuaciones de Maxwell resolviendo la ecuación de onda de Helmholtz.

Los modos se distribuyen transversalmente en el núcleo de la fibra y se extienden hasta la interfaz núcleo-cubierta. Cabe destacar que algunos de estos modos penetran en la cubierta, lo cual indica que son refractados hacia el exterior del núcleo.

El número de modos transportado por una fibra multimodo viene determinado por la siguiente ecuación:

$$M \approx \frac{2\pi}{\lambda_0} (NA) a \quad (1.1)$$

Donde  $a$  es el radio del núcleo de la fibra,  $\lambda_0$  es la longitud de onda de la luz en el vacío

y  $NA$  es la apertura numérica de la fibra. La apertura numérica es directamente proporcional a los índices de refracción del núcleo y cubierta de la fibra, como se ve a continuación:

$$(1.2) \quad NA \approx n_1 \sqrt{2\Delta}$$

Donde conocemos  $n_1$  es el índice de refracción del núcleo y  $n_2$  es el índice de refracción de la cubierta de la fibra:

$$(1.3) \quad \Delta = \frac{n_1 - n_2}{n_1}$$

Material del núcleo de la fibra	Diámetro del núcleo (um)	Índice de refracción	Apertura numérica	Peso (g/m)	Número de modos*
Polimetilacrilato (PMMA)	9 80	1.49	0. 5	1.0	3.082.691
Polimetilacrilato (PMMA)	2 40	1.49	0.5	0.01	184.884
Perfluorato	5 0	(1.32-1.358)	0.2	---	665

\* $\lambda = 0.6328 \text{ um}$

**Tabla 1:** Características de las fibras ópticas multimodo

## 2.6 Ruido modal

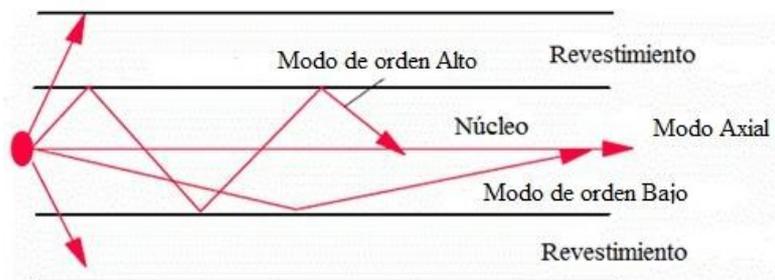
Las propiedades de dispersión intermodal de las fibras ópticas multimodo crean otro fenómeno que afecta a las señales transmitidas en canal óptico (Figura.11). Se muestra el patrón de speckle en fibras multimodo como fluctuaciones que tienen tiempos característicos más largos que el tiempo de resolución del detector y es conocido comúnmente por ruido modal.

Las perturbaciones a lo largo de la fibra tales como las vibraciones, discontinuidades, conectores, empalmes y fuentes de acoplamiento pueden causar oscilaciones en el patrón de speckle y por lo tanto crear ruido modal. Esto se genera cuando la correlación entre dos o más modos que da la interferencia original se retrasa debido a estas perturbaciones, así las condiciones que dan lugar a ruido modal son:

- 2.6.1 Una fuente coherente con una anchura espectral estrecha.
- 2.6.2 Perturbaciones a lo largo de la fibra que producen un retraso de modo diferencial o modal.
- 2.6.3 La correlación de fases entre modos.

Cuando un conector está mal alineado algunos modos pueden alejarse del núcleo por completo y viajar por el revestimiento de la fibra, donde experimentan una atenuación muy alta. En cambio otros modos pueden simplemente transferir su potencia parcialmente o por completo.

Por ejemplo, cuando la luz de un VCSEL (vertical-cavity surface-emitting laser, diodo láser semiconductor con un haz de emisión láser perpendicular sobre la superficie) se acopla en una fibra multimodo después de que los modos viajen una distancia corta a lo largo de la fibra, la luz en cada modo todavía tendrá la misma longitud de onda, pero tendrá un retardo de fase debido a las diferentes velocidades de los modos. Por lo tanto es crucial entender que el proceso de acoplamiento de los modos o el patrón de speckle resultante no es ruido en sí mismo o una fuente directa de ruido. Es una manera normal en que la luz del láser viaja a través de una fibra multimodo.



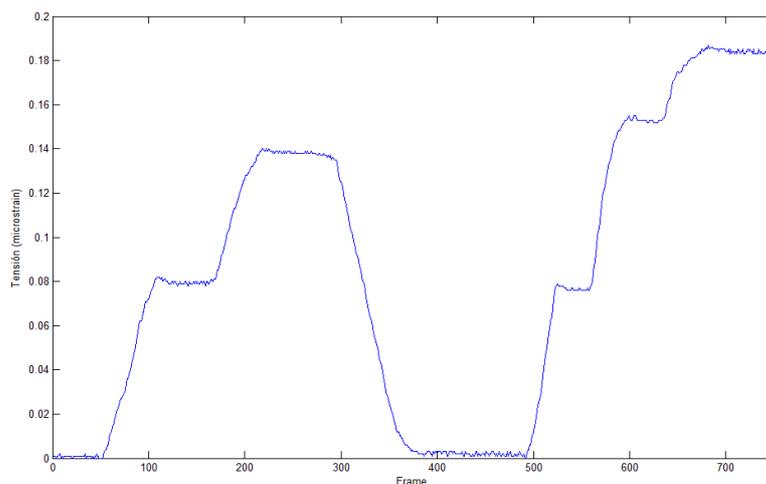
**Figura 6:** Diferentes modos a través de la fibra

### 3. Evaluación de diferentes máscaras

En este trabajo nos centraremos en el diseño de una máscara cuando la fibra óptica se somete a una alteración de tensión en la misma. El objetivo es encontrar una máscara transmitiva que, superpuesta al patrón de speckle, produzca en el fotodiodo una variación de la potencia eléctrica detectada lo más similar posible a la perturbación de la tensión a la que se ha sometido la fibra óptica, consiguiéndose de esta manera la creación de un sensor de bajo coste y alta velocidad de respuesta.

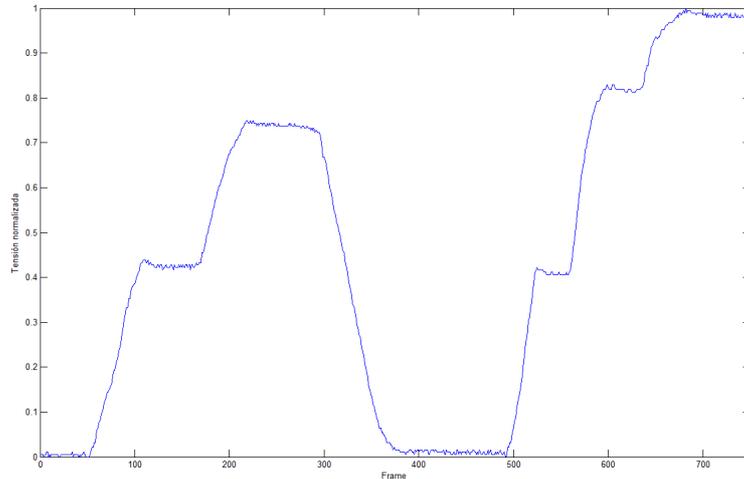
Para ello contamos con un montaje con el cual se hace pasar la luz del láser de helio-neón, que emite a una longitud de onda de 632 nm, a través de la fibra multimodo de 50  $\mu\text{m}$  de diámetro en la que se aumenta la tensión al producir un estiramiento. Este cambio de tensión será medido con un sensor calibrado de referencia y con una cámara se grabará la perturbación del patrón del speckle que posteriormente procesaremos con Matlab. La cámara utilizada es capaz de tomar 25 frames por segundo con una resolución de 320 x 240 (ancho x alto de la imagen).

En la Figura 7, se muestra la representación gráfica de los valores de la perturbación en la fibra para cada fotograma o frame del video del patrón de speckle.



**Figura 7.** Tensión aplicada a la fibra óptica en cada frame del video

Pero lo que nos va a interesar es la representación de los valores normalizados de la tensión para cada frame que es la que emplearemos para comparar con la intensidad normalizada obtenida a la salida al aplicar las máscaras que vamos a diseñar (Figura 8).



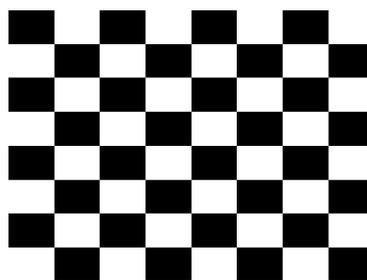
**Figura 8.** Tensión normalizada aplicada a la fibra óptica en cada frame del video

Hemos creado tres tipos de máscaras, una con forma de tablero de ajedrez y las dos restantes centradas en la imagen, una circular y otra con forma de corona circular.

Estas máscaras las hemos aplicado sobre cada frame del video con el fin de obtener el valor de la intensidad detectada en el fotodiodo que compararemos con el valor medido de la tensión para ese frame. Para compararlas emplearemos el error cuadrático medio (ECM). Además, para cada tipo de máscara, se ha creado empleando Matlab una función optimizador que nos permite variar el valor de los parámetros característicos de la máscara para obtener para qué valores de dichos parámetros se obtiene el menor valor del error.

### 3.1 Máscara con forma de tablero de ajedrez

Este tipo de máscaras tiene únicamente un parámetro, que es  $N$ , que debe ser un número entero que haremos variar entre 1 y 10 y que corresponde al número de cuadros que tendrá la máscara por cada fila y columna.



**Figura 9.** Máscara generada para un valor de  $N = 8$

El código utilizado para generar las máscaras con forma de tablero se muestra a continuación.

```
function [ mascara ] = ajedrez( m,n,N )
% La función ajedrez permite generar una máscara en forma de tablero de
% ajedrez que corresponde a una matriz "mascara" cuyas dimensiones son
% mxn. Los parámetros de entrada son m y n que corresponden al número de
% filas y columnas de la matriz que queremos generar respectivamente y N
% es un número entero comprendido entre 1 y 10 que define la
% configuración del tablero siendo (N-1) el número de divisiones de el
% largo y ancho de la imagen de la máscara.

if (N<1 || N>10)

    % Creamos un mensaje de error que aparezca en el caso que el valor
    % de N sea negativo o no este comprendido en el intervalo [0,10]
    disp('N debe ser un número entero contenido en el intervalo [1,10]')
    N=input('Ingrese un nuevo valor para N: ')

end

mN = round(m/N);
nN = round(n/N);

    if (N==1)

        A=zeros(m,n,'uint8');

    elseif (N==2)

        B=zeros(mN,nN,'uint8');
        C=ones(mN,nN,'uint8');
        A=[B C;C B];

    elseif (N==3)

        B=zeros(mN,nN,'uint8');
        C=ones(mN,nN,'uint8');
        A=[B C B;C B C;B C B];

    elseif (N==4)

        B=zeros(mN,nN,'uint8');
        C=ones(mN,nN,'uint8');
        A=[B C B C;C B C B;B C B C;C B C B];

    elseif (N==5)

        B=zeros(mN,nN,'uint8');
        C=ones(mN,nN,'uint8');
        A=[B C B C B;C B C B C;B C B C B;C B C B C;B C B C B];

    elseif (N==6)

        B=zeros(mN,nN,'uint8');
```

```

        C=ones(mN,nN,'uint8');
        A=[B C B C B C;C B C B C B;B C B C B C;C B C B C B;B C B
C B C;C B C B C B];

        elseif (N==7)

            B=zeros(mN,nN,'uint8');
            C=ones(mN,nN,'uint8');
            A=[B C B C B C B;C B C B C B;B C B C B C;C B C B C B
C;B C B C B C;C B C B C B;B C B C B C];

        elseif (N==8)

            B=zeros(mN,nN,'uint8');
            C=ones(mN,nN,'uint8');
            A=[B C B C B C B C;C B C B C B C B;B C B C B C B C;C B C
B C B C B;B C B C B C B C;C B C B C B C B;B C B C B C B C;C B C B C B C
B];

        elseif (N==9)

            B=zeros(mN,nN,'uint8');
            C=ones(mN,nN,'uint8');
            A=[B C B C B C B C B;C B C B C B C B C;B C B C B C B C
B;C B C B C B C B C;B C B C B C B C B;C B C B C B C B C;B C B C B C B C
B;C B C B C B C B C;B C B C B C B C B];

        elseif (N==10)

            B=zeros(mN,nN,'uint8');
            C=ones(mN,nN,'uint8');
            A=[B C B C B C B C B C;C B C B C B C B C B;B C B C B C B
C B C;C B C B C B C B C B;B C B C B C B C B C;C B C B C B C B C B;B C B
C B C B C B C;C B C B C B C B C B;B C B C B C B C B C;C B C B C B C B C
B];

        else

            disp('Debe ser menor o igual que 10')

        end

% Al emplear el comando round(), pueden salir máscaras que no lleguen a
% las dimensiones
% mxn o que sean más grandes por lo que en estos casos debemos
% rellenarlas o recortarlas

if (size(A,1) > m) % Si son más grandes se recorta
    A = A(1:m,:);
end

if (size(A,2) > n)
    A = A(:,1:n);
end

if (size(A,1) < m || size(A,2) < n) % Si son más pequeñas se rellena
con ceros

```

```

        A(m,n) = 0;
    end

mascara=A;
imshow(mascara*255)

end

```

El optimizador utilizado para este tipo de máscaras tiene el siguiente código.

```

function [Nmin,ECMmin]=optimizador(videoname)
% La función optimizador permite obtener para que valor de N, que toma
% valores de 1 a 10 % cada uno de los cuales corresponde a una
% configuración de la máscara en forma de tablero de ajedrez, el error
% cuadrático medio respecto a los valores medidos de la tensión en la
% fibra. El único parámetro de entrada que requiere es el nombre del
% vídeo con su extensión correspondiente del speckle al aplicar la
% tensión.
% A la salida la función nos devuelve el menor valor de N y el error
% correspondiente.

%Leemos los datos del vídeo que vamos a procesar
Vspeckle=VideoReader('incremento_FBG.avi');

%Mostramos las características del vídeo y las guardamos en una
%variable
info=get(Vspeckle);
%Almacenamos en una variable el número de frames del vídeo
Nf=Vspeckle.NumberOfFrames;

%Leemos el primer frame del vídeo que va a ser nuestro punto de partida
frame1=read(Vspeckle,1);
%Convertimos la imagen correspondiente al primer frame en una imagen en
%escala de grises(imagen de intensidad)
frame1_1=rgb2gray(frame1);

%Guardamos en un array la altura (Height)y la anchura (Width)de la
%imagen
[m,n]=size(frame1_1);

%Generamos la máscara llamando a la función 'ajedrez'para lo cual
%debemos proporcionar un valor entero entre 1 y 10 a N.
N=0;

for N=1:10

mascara=ajedrez(m,n,N);

%Creamos un bucle que nos permita aplicar la máscara a cada frame
I=[];
for i=1:Nf
    frame=read(Vspeckle,i);

```

```

        frame=rgb2gray(frame);
        Ii=mascara.*frame;
        I(i,1)=sum(sum(Ii));
end

if N==1

    I_nor=I;

else

%Normalizamos I dividiendo cada término de la matriz por el valor máximo
I_nor=I./max(max(I));

end

%Guardamos en un array los valores medidos de la tensión y le
%normalizamos
strain='Strain_profile.mat';
load(strain)
strain_nor=strain./max(max(strain));

%Comparamos este valor con el esperado empleando el error cuadrático
%medio
%Introducimos el array que contendrá el valor esperado del strain
%normalizado para cada frame del vídeo
dif=[];
for i=1:Nf

    dif(i,1)=(I_nor(i,1)-strain_nor(i,1))^2;

end

ECM=(1/Nf)*sum(sum(dif));

ECM_N(N,1)=ECM;

end

%Mostramos los valores del array ECM_N en el cada elemento corresponde
%al error cuadrático medio para cada valor de N
ECM_N
[ECMmin Nmin]=min(ECM_N)

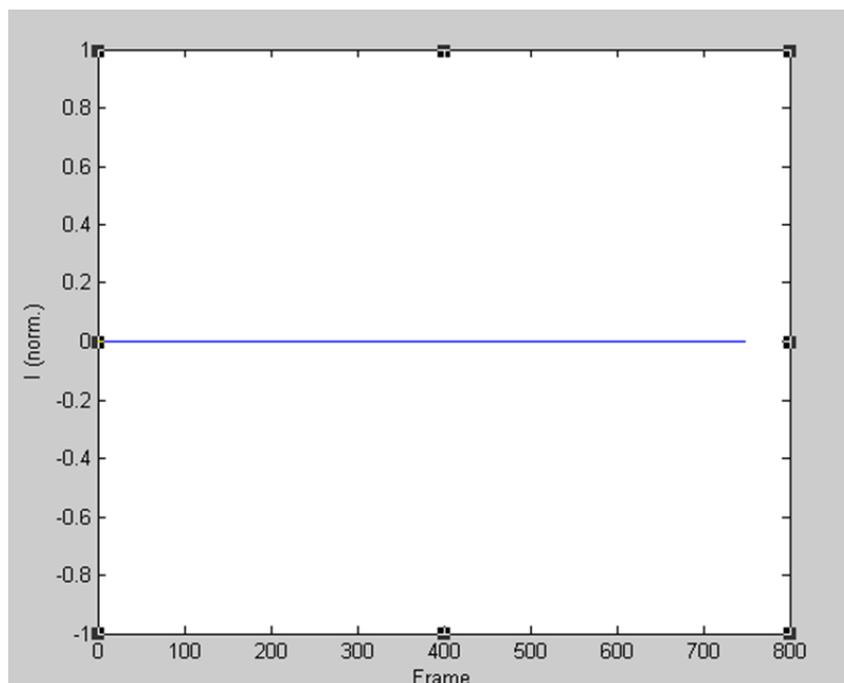
```

Con este optimizador se han obtenido los resultados que se muestran en la Tabla 2.

N	1	2	3	4	5	6	7	8	9	10
ECM	32.8	37.3	35.0	35.8	36.1	38.5	35.3	39.0	38.5	34.6
(%)	3	2	0	4	7	5	1	4	8	5

**Tabla 2.** Valores del error cuadrático medio para cada valor del parámetro N

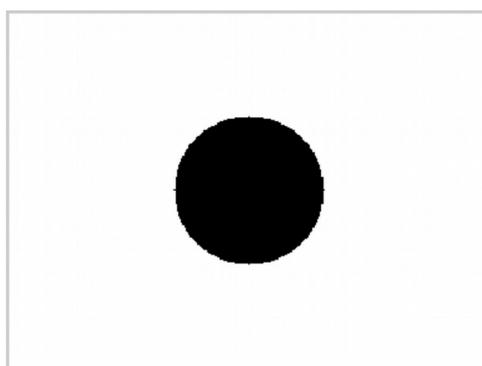
Comprobamos que el mejor resultado (menor error cuadrático medio) se obtiene para una máscara totalmente opaca correspondiente a  $N=1$  por lo que las máscaras con este diseño no ofrecen buenos resultados.



**Figura 10.** Intensidad total normalizada obtenida al aplicar la máscara en cada frame del video para  $N=1$

### 3.2 Máscara circular

Esta máscara tiene una forma circular opaca centrada en la imagen cuyo único parámetro es el radio del círculo,  $R$ . Hemos hecho variar  $R$  de 10 a  $m/2$ , donde  $m$  es la anchura de la imagen, aumentando progresivamente el radio de 10 en 10.



**Figura 11.** Máscara para un valor de  $R=50$

El código utilizado para generar este tipo de máscaras se muestra a continuación.

```
function [mascara]=circulo_centrado(m,n,radio)
%La función circulo_centrado tiene como parámetros de entrada las
%dimensiones de la imagen mxn y el valor del radio del círculo

R=radio;

%Elegimos en que posición vamos a situar el centro del círculo que en
%este caso será el centro de la imagen
xo=m/2;
yo=n/2;

x=1:m;
y=1:n;

[xx,yy]=meshgrid(x,y);

r=sqrt(((xx-xo).^2)+((yy-yo).^2))<=R;

%Trasponemos la matriz r para que tenga las dimensiones que queremos
%mxn
r1=r';

r11=uint8(r1);

r111=ones(m,n,'uint8')-r11;

mascara=r111;
imshow(mascara*255)
```

El código del optimizador en este caso es el siguiente.

```
function [ECMmin]=optimizador_circulo(videoname)

%Leemos los datos del vídeo que vamos a procesar
Vspeckle=VideoReader(videoname);

%Mostramos las características del vídeo y las guardamos en una
%variable
info=get(Vspeckle);
%Almacenamos en una variable el número de frames del vídeo
Nf=Vspeckle.NumberOfFrames;

%Leemos el primer frame del vídeo que va a ser nuestro punto de partida
frame1=read(Vspeckle,1);
%Convertimos la imagen correspondiente al primer frame en una imagen en
%escala de grises(imagen de intensidad)
frame1_1=rgb2gray(frame1);
```

```

%Guardamos en un array la altura (Height)y la anchura (Width)de la
%imagen
[m,n]=size(frame1_1);

%Guardamos en un array los valores medidos de la tensión y le
%normalizamos
strain='Strain_profile.mat';
load(strain)
strain_nor=strain./max(max(strain));

%Generamos la máscara llamando a la función 'circulo_centrado'para lo
%cual debemos proporcionar el valor del radio círculo
R=0;
p=10;

for R=10:p:m/2

    mascara=circulo_centrado(m,n,R);
    t=((R-p)/p)+1;

    %Creamos un bucle que nos permita aplicar la máscara a cada frame
    I=[];
    for i=1:Nf
        frame=read(Vspeckle,i);
        frame=rgb2gray(frame);
        Ii=mascara.*frame;
        I(i,1)=sum(sum(Ii));
    end

    %Normalizamos I dividiendo cada término de la matriz por el valor
    %máximo
    I_nor=I./max(max(I));

    %Comparamos este valor con el esperado empleando el error
    %cuadrático medio
    %Introducimos el array que contendrá el valor esperado del strain
    %normalizado para cada frame del vídeo
    dif=[];

    for i=1:Nf
        dif(i,1)=(I_nor(i,1)-strain_nor(i,1))^2;
    end

    ECM=(1/Nf)*sum(sum(dif));

    %Guardamos los valores del radio y del error cuadrático medio
    %correspondiente
    %en dos arrays, radio y ECM_R
    radio(t,1)=R;
    ECM_R(t,1)=ECM;

end

%Guardamos en dos variables el valor mínimo del error cuadrático medio
%y la posición que ocupa en el array
[ECMmin posicion]=min(ECM_R);

```

```

radio
ECM_R
%Guardamos en una variable el valor del radio al que corresponde ECM_R
Rmin=radio(posicion,1)
ECMmin

return;

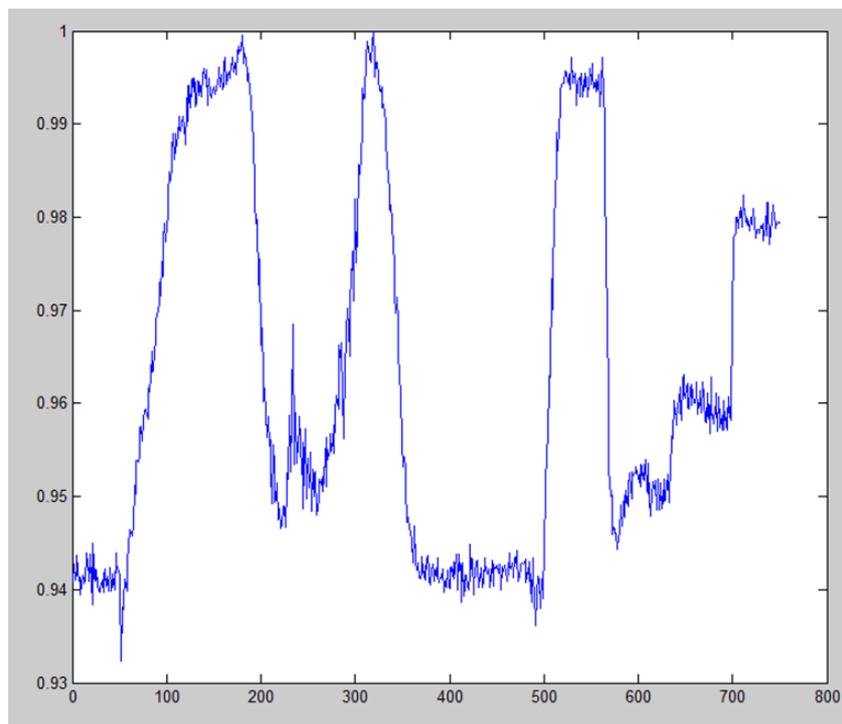
```

Con este optimizador se han obtenido los resultados que se muestran en la Tabla 3.

R	10	20	30	40	50	60	70	80	90	100	110	120
ECM (%)	37.80	38.28	38.37	38.00	38.75	39.36	38.69	38.60	38.78	38.26	39.31	38.73

**Tabla 3.** Valores del error cuadrático medio para cada valor de R

El mejor resultado en este caso se obtiene con la máscara circular de menor radio 37.8%.



**Figura 12.** Intensidad total normalizada obtenida al aplicar la máscara en cada frame del video para R = 10

### 3.3 Máscara con forma de corona circular

En este caso, haremos variar el radio menor de la corona circular  $R_1$  de 10 a  $(m/2-\text{delta})$  siendo delta la anchura de la corona aumentándole como hemos hecho en el caso anterior progresivamente de 10 en 10.



**Figura 13.** Máscara para una delta =15 y un  $R_1 =10$

El código utilizado para generar este tipo de máscaras se muestra a continuación.

```
% La función corona_circular permite crear una máscara con forma de
% corona circular centrada en el punto (xo,yo). Los parámetros de
% entrada de la función son el punto central de la corona definido por
% xo e yo, el radio menor y mayor de la misma y la altura m y anchura n
% de la imagen. A la salida nos devuelve la matriz generada
% correspondiente a esta máscara de dimensiones mxn.

function[mascara]=corona_circular(m,n,xo,yo,radio_menor,radio_mayor)

R1=radio_menor;
R2=radio_mayor;

x=1:m;
y=1:n;

[xx,yy]=meshgrid(x,y);

r=sqrt(((xx-xo).^2)+((yy-yo).^2))<=R1;

% Trasponemos la matriz r para que tenga las dimensiones que queremos
% mxn
r1=r';
r11=uint8(r1);
r111=ones(m,n,'uint8')-r11;

R=sqrt(((xx-xo).^2)+((yy-yo).^2))<=R2;
r2=R';
r22=uint8(r2);
```

```

mascara=r111.*r22;
mascaral=ones(m,n,'uint8')-mascara;
imshow(mascaral*255)

```

El código del optimizador en este caso es el siguiente.

```

function [ECM_min,I_nor]=optimizador_coronaCircular(videoname,delta)

%Leemos los datos del vídeo que vamos a procesar
Vspeckle=VideoReader(videoname);

%Mostramos las características del vídeo y las guardamos en una variable
info=get(Vspeckle);
%Almacenamos en una variable el número de frames del vídeo
Nf=Vspeckle.NumberOfFrames;

%Leemos el primer frame del vídeo que va a ser nuestro punto de partida
frame1=read(Vspeckle,1);
%Convertimos la imagen correspondiente al primer frame en una imagen en
%escala de grises(imagen de intensidad)
frame1_1=rgb2gray(frame1);

%Guardamos en un array la altura (Height)y la anchura (Width)de la
%imagen
[m,n]=size(frame1_1);

%Guardamos en un array los valores medidos de la tensión y le
%normalizamos
strain='Strain_profile.mat';
load(strain)
strain_nor=strain./max(max(strain));

% Seleccionamos como punto central de la corona el centro de la imagen
% (xo,yo)=(m/2,n/2)
xo=m/2;
yo=n/2;

%Generamos la máscara llamando a la función 'circulo_centrado'para lo
%cual debemos proporcionar el valor del radio círculo
R1=10;
%R1=0;
%R2=delta;
p=10;

%for R1=10:p:(m/2-delta)

    t=((R1-p)/p)+1;
    R2=R1+delta;

    mascara=corona_circular(m,n,xo,yo,R1,R2);

%Creamos un bucle que nos permita aplicar la máscara a cada frame

```

```

I=[];
for i=1:Nf
    frame=read(Vspeckle,i);
    frame=rgb2gray(frame);
    Ii=mascara.*frame;
    I(i,1)=sum(sum(Ii));
end

%Normalizamos I dividiendo cada término de la matriz por el valor
%máximo
I_nor=I./max(max(I))

%Comparamos este valor con el esperado empleando el error
%cuadrático medio
%Introducimos el array que contendrá el valor esperado del strain
%normalizado para cada frame del vídeo
dif=[];

for i=1:Nf
    dif(i,1)=(I_nor(i,1)-strain_nor(i,1))^2;
end

ECM=(1/Nf)*sum(sum(dif));

% Guardamos los valores del radio y del error cuadrático medio
%correspondiente
% en dos arrays, radio y ECM_R
R1_valores(t,1)=R1;
R2_valores(t,1)=R2;
ECM_R(t,1)=ECM;

R1_valores
R2_valores
ECM_R
% Guardamos en dos variables el valor mínimo del error cuadrático medio
%y la posición que ocupa en el array
[ECM_min posicion]=min(ECM_R);

% Guardamos en una variable el valor del radio al que corresponde ECM_R
R1_min=R1_valores(posicion,1)
R2_min=R2_valores(posicion,1)

return;

```

Puesto que la anchura de la corona es un parámetro cuyo valor podemos elegir probaremos con varios valores.

Delta	10	15	20	30	40	50	60
R1min	10	10	10	10	10	10	10
R2min	20	25	30	40	50	60	70
ECMmin	19.3	19.2	23.4	27.3	26.1	27.1	31.4
(%)	2	8	0	4	1	1	1

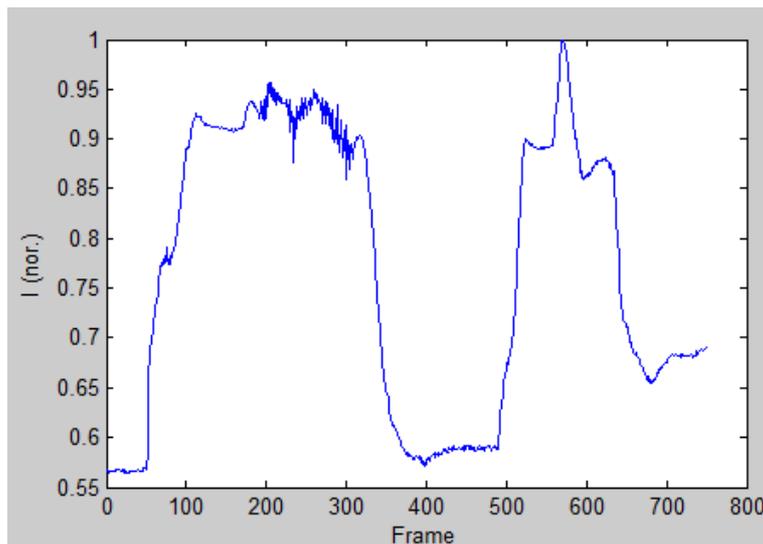
**Tabla 4.** Error cuadrático medio mínimo y los valores correspondientes del radio menor R1 y radio mayor R2 para diferentes anchuras de la corona.

Se comprueba que el error es menor para la anchura de 15 y que para estos valores el radio menor es el mínimo que se ha tomado para nuestro estudio. Por ello, mostraremos los resultados obtenidos empleando esta anchura.

R1	10	20	30	40	50	60	70	80	90	100
R2	25	35	45	55	65	75	85	95	105	115
EC	19.2	26.7	26.4	21.8	33.9	34.4	28.6	24.4	29.4	34.0
M (%)	8	9	6	4	3	3	0	9	6	5

**Tabla 5.** Valores del error cuadrático medio para cada valor de R1 y R2 para una delta de 15

Los mejores resultados se obtienen para una anchura de 15 y un radio interior  $R_1=10$ . Con estos valores vemos que la representación que se muestra en la Figura 16 se aproxima bastante a la variación de la tensión normalizada por cada frame.



**Figura 14.** Intensidad total normalizada obtenida al aplicar la máscara para delta=15 en cada frame del video para  $R_1=10$

Si comparamos esta figura con la Figura 8, comprobamos que ambas gráficas presentan un aspecto similar por lo que para este tipo de máscara habríamos conseguido demostrar la correspondencia que buscábamos con un error cuadrático medio de un 19.28 % que es aceptable.

#### 4. Conclusiones y líneas futuras

El patrón de speckle es muy sensible a cualquier perturbación que se produzca en el medio ambiente y sin aplicar ninguna perturbación los puntos de luz cambian de posición lentamente. Al aplicar una perturbación tal como vibraciones, torsión, temperatura, etc, se producen cambios muy rápidos que pueden ser grabados con una cámara. El problema es que hay perturbaciones que son tan rápidas que no pueden ser grabadas con cámaras convencionales y por ello, es conveniente emplear un fotodiodo que es capaz de medir la señal con una frecuencia un millón de veces mayor.

En este trabajo se ha demostrado que es posible generar con Matlab una máscara que al situarla frente al fotodiodo nos permita obtener una señal que tenga una correlación con la perturbación a la que se ha sometido la fibra óptica a partir de un procesamiento adecuado de las imágenes de la variación del speckle. Esto nos permite conseguir un sensor basado en speckle.

El trabajo que queda es:

- Probar diferentes algoritmos genéticos para que la generación de las máscaras sea óptima.
- Probar en el laboratorio imprimiendo la máscara en una transparencia y situarla en frente del fotodiodo.
- Generar la máscara con un generador de máscaras en transmisión DLP (Digital Light Processing).

#### 5. Referencias

- [1] Lomer, M.; Rodríguez-Cobo, L.; López-Higuera, J.M. *Specklegrams correlation in optical fibers and their applications to sensors* 2015, 9ª Reunión de optoelectrónica OPTOEL'15
- [2] Revilla Varona, P. Patrones de Speckle en fibra óptica multimodo para la medida del ritmo cardiaco en pacientes. Trabajo de fin de grado [Online], Universidad de Cantabria, Julio 2014.  
<https://repositorio.unican.es/xmlui/handle/10902/5743>