

Práctica 1 - MTF del ojo humano

Daniel Torres Robledo
shadow.cat6333@gmail.com

Andrés Fuentes Hernández
andres7233@hotmail.com

Abstract—En este documento se describen las aplicaciones de la *MTF*, el cálculo de ecuaciones para generarla, su implementación en *python*, así como experimentos utilizando este estímulo con diferentes personas a distintas distancias, que para obtener medidas invariantes a la distancia de observación, el cálculo de esta frecuencia será expresado en ciclos/grado.

Index Terms—*MTF*, *SVH*, *python*, sinusoidal modulada, experimentos, invariantes

I. Objetivo

- Encontrar la *MTF* del ojo experimentalmente.
- Encontrar la frecuencia de máxima sensibilidad del ojo humano.
- Generar los cálculos necesarios para obtener la función sinusoidal en frecuencia y amplitud.
- Desplegar el estímulo visual sinusoidal modulada.

II. Introducción

La *MTF* (*Modulation Transfer Function*) o función de transferencia de modulación se ha convertido en una herramienta muy utilizada para especificar el rendimiento y la resolución de toda clase de sistemas ópticos y de visión, que van desde lentes, cintas magnéticas y películas hasta telescopios, la atmósfera y el ojo humano.

La *MTF* es usada para caracterizar sistemas lineales e invariantes en el espacio, sin embargo, a pesar de que el Sistema Visual Humano (*SVH*) no cumple con estas dos propiedades, la *MTF* se ha utilizado para caracterizarlo bajo condiciones de iluminación controladas.

La *MTF* también puede interpretarse como la capacidad de un sistema óptico para percibir contraste, es decir, la capacidad de resolver o diferenciar líneas a una determinada frecuencia espacial.

Para entender de manera simplificada el concepto de frecuencia espacial, hay que notar que las frecuencias espaciales bajas equivalen a repeticiones de patrones muy separados espacialmente (por ejemplo, líneas muy separadas) y que las altas equivalen a repeticiones más compactas.

Hay que considerar que todo sistema de visión tiene un límite superior a partir del cual ya no le es posible distinguir más detalles. Este límite está directamente relacionado con la frecuencia de *Nyquist* [4]. En los estudios de sensibilidad del *SVH* a base de experimentación psico-física, se ha encontrado que su *MTF* consiste en una función del tipo paso-banda con un pico o frecuencia espacial máxima en el rango de 2 a 6 ciclos por grado [1] [2].

En esta practica se examinará la sensibilidad del *SVH* a distintas frecuencias espaciales; la idea general de crear un

patrón de sinusoidales, cuya frecuencia aumenta en el eje horizontal y el contraste en el eje vertical. La envolvente del patrón visibles generalmente muestra un comportamiento similar a la curva de la *MTF*.

Comprobaremos que dicho pico, que representa la frecuencia máxima de sensibilidad del ojo humano se moverá conforme el observador se aleja o acerca del estímulo. Para obtener una medición invariante a la distancia de observación, el cálculo de esta frecuencia de observación debe ser expresado en ciclos/grado.

A. Función Sensibilidad al Contraste

En 1967 *Cambell* y *Robson* del laboratorio de psicología de la universidad de *Cambridge* [5] definieron a la función de sensibilidad al contraste como la variación de la sensibilidad a lo largo de un rango de frecuencias espaciales. Ellos sostenían que nuestra habilidad para percibir el detalle de una escena visual esta determinado por el tamaño relativo y el contraste del detalle presente.

Para obtener el limite del contraste ellos diseñaron la siguiente prueba:

Generaron objetivos de rejillas con diferentes modulaciones y formas de onda. Una trama de 2000 líneas fue producida en una pantalla de rayos catódicos a una frecuencia de 100 ciclos/segundo.

La intensidad del rayo fue modulada por una señal de un generador de señales. La luminancia fue de $500 \cdot \frac{cd}{m^2}$. En frente de una pantalla blanca fue puesta una hoja de cartón blanco de 30cm de diámetro con una apertura de 10cm × 10cm o 2cm × 2cm. Al sujeto de prueba se le aplicó homotropina en un ojo para obtener una pupila de 2.5mm de diámetro, la prueba fue monocular. Los limites de contraste fueron determinados por el sujeto ajustando el contraste de una rejilla de cierta frecuencia espacial, hasta que el patrón era casi indistinguible.

Es de estos experimentos que surge la gráfica *Cambell* y *Robson CSF* (figura 1), ya que al tener un barrido en frecuencias asemeja el barrido realizado por ellos en el tubo de rayos catódicos. En sus resultados estipulan que la máxima sensibilidad del ojo humano se encuentra al rededor de los 4 ciclos por grado.

Estos resultados son útiles en el diagnostico de enfermedades como el glaucoma [6], pero también en determinar que es visible para el ojo humano, y así, un sistema inteligente de un automóvil pueda decidir en que momentos es necesario asistir al usuario.

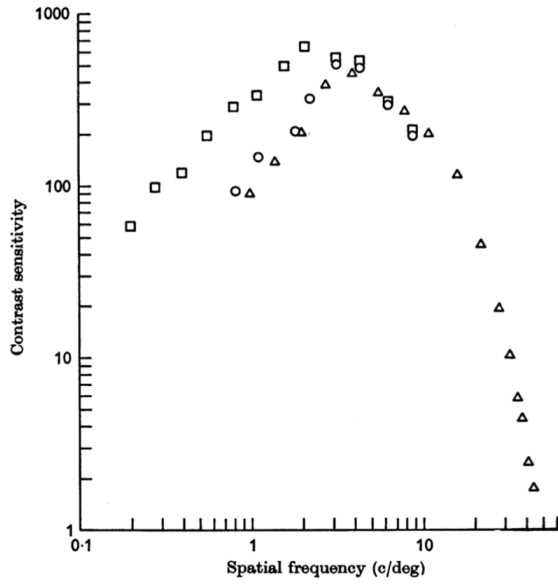


Fig. 1. Gráfica Cambell y Robson CSF.

III. Desarrollo

En esta sección se muestra el cálculo para obtener las constantes de la función sinusoidal modulada en frecuencia y en amplitud.

Sea la función I , que dará los valores de cada pixel de la imagen:

$$I(n, m) = g(n) \cdot \sin(2\pi \cdot f(m))$$

h = altura de la imagen

w = ancho de la imagen

Sea g la función de contraste, donde es necesario encontrar las constantes c, d :

$$g(n) = d \cdot e^{-c \cdot n}$$

Para el eje horizontal, utilizaremos f :

$$f(m) = a \cdot e^{b \cdot x}$$

Por lo que la frecuencia instantánea está definida como la derivada de f :

$$f'(m) = a \cdot b \cdot e^{b \cdot x}$$

Es necesario encontrar las constantes a, b tal que $f'(w) = 1/2$ para obtener la frecuencia máxima, y $f'(0) = 1/w$ para encontrar la frecuencia mínima, en *ciclos/segundo*

$$1) f'(0) = a \cdot b \cdot e^{b \cdot 0} = a \cdot b \cdot 1 = a \cdot b = 1/w$$

$$2) f'(w) = a \cdot b \cdot e^{b \cdot w} = 1/2$$

Para encontrar b , sustituimos 1 en 2:

$$(1/w) \cdot e^{b \cdot w} = 1/2$$

$$e^{b \cdot w} = w/2$$

$$\ln(e^{b \cdot w}) = \ln(w/2)$$

$$b \cdot w \cdot \ln(e) = \ln(w/2)$$

$$b = \frac{\ln(w/2)}{w \cdot \ln(e)}$$

$$b = \frac{\ln(w/2)}{w \cdot \ln(e)}$$

$$b = \frac{\ln(w/2)}{w}$$

Por lo que para encontrar a sustituimos b en 1)

$$a \cdot b = 1/w$$

$$a = \frac{1}{b \cdot w}$$

Por lo tanto ya tenemos los valores de a, b :

$$a = \frac{1}{b \cdot w}$$

$$b = \frac{\ln(w/2)}{w}$$

Para encontrar las constantes c, d de $g(n)$:

$$g(n) = d \cdot e^{-c \cdot n}$$

Donde $\epsilon > 0$.

$$1) g(0) = d \cdot e^{-c \cdot 0} = d = \epsilon$$

$$2) g(h) = d \cdot e^{-c \cdot h} = 1$$

Para encontrar c , sustituimos 1 en 2:

$$\epsilon \cdot e^{-c \cdot h} = 1$$

$$\ln(\epsilon \cdot e^{-c \cdot h}) = \ln(1)$$

$$\ln(\epsilon) + \ln(e^{-c \cdot h}) = 0$$

$$-c \cdot h \cdot \ln(e) = 0$$

$$c = \frac{\ln(\epsilon)}{h}$$

Por lo tanto, para $\epsilon > 0$, en este caso utilizaremos $\epsilon = 0.02$:

$$d = \epsilon = 0.02$$

$$c = \frac{\ln(\epsilon)}{h}$$

A continuación se muestra el estímulo *MTF* utilizando dichas constantes:

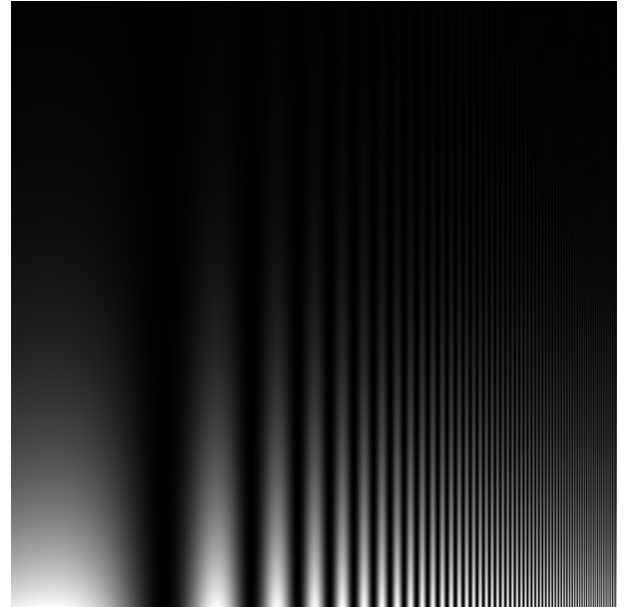


Fig. 2. Estímulo visual (sinusoidal modulada).

Para calcular los valores que se reportarán en los resultados son:

X : coordenada x donde se encontró experimentalmente el punto más alto visible por el usuario.

$\phi = f'(x) = a \cdot b \cdot e^{b \cdot x}$: Qué es la frecuencia instantánea evaluada en x , que son ciclo/pixel.

$dd = \text{screensize}/w$: Es la distancia entre dos pixels, que es la medida de la imagen en centímetros entre la cantidad de pixels.

$\alpha = \tan^{-1}(dd/\text{distancia}) \cdot (180/\pi)$: Resultado en grados/pixels, el valor de distancia es la distancia a la que se encuentra el observador en centímetros.

$f = \frac{\phi}{\alpha}$: Frecuencia de observación en ciclos/grado

Debido a que las herramientas para mostrar una imagen usualmente realizan un escalamiento, la prueba puede verse afectada, por lo que también se decidió probar con una imagen en la *web* generada por un *shader 3d*, lo que presenta como ventaja en la prueba es que el renderizado va de acuerdo al dispositivo con el que se accede a la pagina, es decir, si se ingresa con un dispositivo cuya ventana de navegador es de 1024×512 *pixels*, la imagen sera generada de acuerdo a estas dimensiones sin escalamientos. El código para generar el estímulo de esta prueba se muestra en la figura 10.

Este conjunto de pruebas pertenecen al mismo sujeto, donde todas las imágenes fueron realizadas utilizando el mismo monitor a una distancia de 65cm , la primera gráfica (figura 3), es con el uso de lentes que corrigen miopía y astigmatismo. En esta se puede notar que la frecuencia a la cual se encuentra la sensibilidad máxima es a los 4.012 cpd .

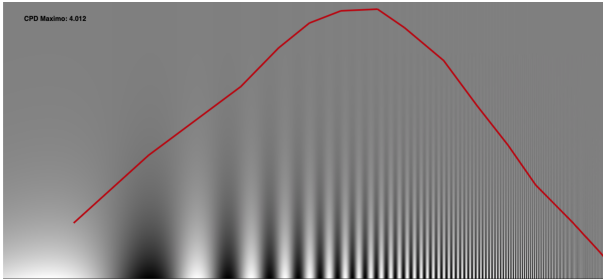


Fig. 3. Estímulo visual, con lentes que corrigen miopía y astigmatismo.

La segunda prueba se realizó con solo un ojo, el cual tiene miopía y astigmatismo, se puede mostrar que la curva se mueve considerablemente a la izquierda y la sensibilidad máxima disminuye (figura 4).

Finalmente la misma prueba se realizó utilizando otro ojo con miopía se puede observar que aunque también se puede observar un desplazamiento en la curva no es tan significativo como el que tiene astigmatismo (figura 5).

IV. Resultados

Para este experimento, se utilizó el estímulo visual *MTF* descrito en la sección de Desarrollo, el cual fue mostrado a tres diferentes personas a las distancias de 30cm , 60cm y 100cm , en donde fueron calculados para cada distancia los valores de

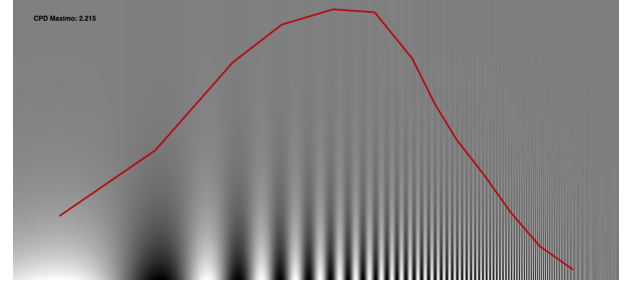


Fig. 4. Estímulo visual, sin lentes que corrigen miopía y astigmatismo.

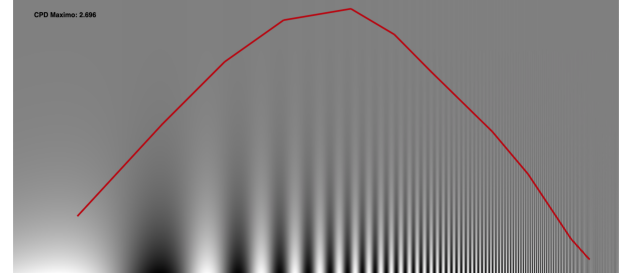


Fig. 5. Estímulo visual, para un ojo con miopía.

x , ϕ , α y frecuencia de observación f . La descripción de estas ecuaciones también se encuentra en el apartado de Desarrollo.

En este experimento se utilizaron las siguientes constantes:

- $height = 768$: pixels de altura de la imagen
- $width = 768$: pixels de longitud de la imagen
- $screensize = 13\text{cm}$ medida de la imagen en centímetros
- $distancia \in [30\text{cm}, 60\text{cm}, 100\text{cm}]$: distancias en centímetros a las que se encuentra el usuario
- $\epsilon = 0.01$: constante para calcular el contraste.

	$D_1 = 30\text{cm}$	$D_2 = 60\text{cm}$	$D_3 = 100\text{cm}$
X	513	430	359
ϕ'	0.088	0.045	0.025
α	0.034	0.017	0.010
f	2.569	2.623	2.460

Fig. 6. Tabla de resultados, experimento sin lentes y en morado resultados con lentes 1.1.

	$D_1 = 30\text{cm}$	$D_2 = 60\text{cm}$	$D_3 = 100\text{cm}$
X	627	541	475
ϕ'	0.221	0.110	0.064
α	0.034	0.017	0.010
f	6.469	6.446	6.295

Fig. 7. Tabla de resultados, experimento sin lentes y en morado resultados con lentes 1.2.

V. Conclusiones

Se puede observar como el valor de x se mueve a la izquierda conforme se aleja el usuario de la imagen., sin embargo el valor de la frecuencia de observación f , en las

	$D_1 = 30cm$	$D_2 = 60cm$	$D_3 = 100cm$
X	498	418	358
ϕ'	0.078	0.041	0.025
α	0.034	0.017	0.010
f	2.275	2.380	2.440

Fig. 8. Tabla de resultados, experimento 1.3.

figuras 6, 7 y 8 se mantiene aproximadamente igual, este error puede ser en parte por la acción del usuario al dar *click* sobre la imagen y mover el cursor una cierta cantidad de pixels.

Por lo que se cumplió la hipótesis de tener los ciclos/grado constantes, a pesar de la distancia a la que se encuentra el observador.

Todas las pruebas fueron realizadas con el mismo nivel de brillo de la pantalla del dispositivo.

VI. Código fuente

A continuación se muestra la implementación de las ecuaciones mostradas en el Desarrollo para calcular el estímulo visual y los valores reportados en los resultados de x , ϕ , α y frecuencia de observación f .

Es posible modificar las constates referentes al tamaño de la imagen, la medida en centímetros de ésta, y la distancia a la que se encuentra el usuario en centímetros, esto en las variables definidas en el *main* del programa.

Este programa permite al usuario dar *click* sobre la imagen para tener el valor de la coordenada en x , así como los valores reportados en la tabla de resultados calculados de manera automática.

```
1 import cv
2 import math
3
4 def user_x(event,x,y,flags,param):
5     if event == cv.EVENT_LBUTTONDOWN:
6         param = map(float, param)
7         n_pixels = param[0]
8         a = param[2]
9         b = param[3]
10        screen_size = param[4]
11        user_distance = param[5]
12
13        snap = a*math.e**(b*x)
14        d = screen_size/n_pixels
15        alpha = math.atan(d/user_distance)*(180.0/math.pi)
16        f = snap/alpha
17        print "Distance:%.3f, X:%.3f, Snap:%.3f, Alpha:%.3f, F:%.3f"%(
18            user_distance, x, snap, alpha, f)
19
20 def create_mtf(name, width, height, screen, distance_user):
21     out = cv.CreateImage((width, height), 8, 1)
22
23     b = math.log(width/2.0)/width
24     a = 1.0/(b*width)
25
26     d = 0.01
27     c = math.log(d)/height
28
29     x_vector = [0.5+0.5*math.sin(2*math.pi*a*math.e**(b*val)) for val in
30                 xrange(width)]
31     y_vector = [d*math.e**(-c*val) for val in xrange(height)]
32
33     for i in xrange(width):
34         for j in xrange(height):
35             out[j,i] = 255.0*(x_vector[i]*y_vector[j])
36
37     cv.ShowImage("mtf", out)
38     cv.MoveWindow("mtf", 0, 0)
39
40     cv.SetMouseCallback("mtf", user_x, [width,height,a,b,screen,distance_user])
41
42     cv.SaveImage("mtf.png", out)
43     cv.WaitKey(0)
44
45 if __name__ == "__main__":
46     size = 728
47     screen_cm = 13.0
48     distance_user_cm = 100.0
49     create_mtf("out.png", size, size, screen_cm, distance_user_cm)
```

Fig. 9. Código utilizado para la implementación en *python*.

Para la implementación, la función sinusoidal de multiplicó por 0.5 y se sumó 0.5 para ajustar los valores al rango $[0, 1]$, ya que estaban en $[-1, 1]$.

Para la implementación de la versión en *web* se utilizó el siguiente código:

```
#define PI 3.14159265359
uniform vec2 u_resolution;
uniform float Frequency_B;
uniform float Frequency_A;
uniform float Atenuacion_C;
uniform float Amplitud;

void main( void ) {
    vec2 position = gl_FragCoord.xy;
    float Frequency = Frequency_A*exp(Frequency_B*position.x);
    float VaryngSin = Amplitud*sin(2.0*PI*Frequency);

    float Atenuacion = exp(-1.0*position.y*Atenuacion_C);
    float final = Atenuacion * VaryngSin + Amplitud;

    vec3 color = vec3(final);
    gl_FragColor = vec4( color, 1.0 );
}
```

Fig. 10. Código utilizado para la implementación en *web*.

El cual se puede visualizar con el siguiente enlace: <https://afh7233.github.io/>

REFERENCES

- [1] Pratt, W. k., Digital Image Processing, John Wiley & Sons Inc, 2001.
- [2] Levine, M.D., Vision in man and machine, McGraw-Hill, 1985.
- [3] Norman Koren, *Resolution and MTF curves in scanners and sharpening*. Available at <http://www.normankoren.com/Tutorials/MTF2.html>. [Accessed Agosto 31, 2019]
- [4] Norman Koren, *Nyquist frequency*. Available at https://en.wikipedia.org/wiki/Nyquist_frequency. [Accessed Agosto 31, 2019]
- [5] Campbell, F. W. and Robson, J. G. (1968) Application of Fourier analysis to the visibility of gratings. Journal of Physiology (London) 197, 551-566.
- [6] Allison M. McKendrick, *Contrast Sensitivity Changes Due to Glaucoma and Normal Aging* <https://iovs.arvojournals.org/article.aspx?articleid=2164531> [Accessed Agosto 31, 2019]