Laboratorio 4: Imágenes híbridas

Ramón Emiliani
Universidad de los Andes
201125694
rd.emiliani689@uniandes.edu.co

Alejandro Posada
Universidad de los Andes
20121
a.posada10@uniandes.edu.co

Abstract

Las imágenes híbridas son imágenes cuya interpretación varía según la distancia a la cual son observadas. Debido al procesamiento multiescala del sistema visual humano, estas imágenes parecen adoptar formas distintas a medida que el observador se acerca o se aleja. En este laboratorio se emplearon dos imágenes de caras para crear una imagen híbrida en la que aparecen dos personas diferentes según la distancia a la que se observe la imagen obtenida.

1. Introducción

En 2006, los investigadores Aude Oliva y Antonio Tarralba, del MIT, y Philippe Schyns, de la Universidad de Glasgow, presentaron una técnica para obtener imágenes híbridas [2]. Estas son imágenes que son percibidas de dos maneras distintas según la distancia a la que se observen. Estas imágenes se forman al superponer dos imágenes, una en una escala espacial baja y la otra en una escala espacial alta [2]. La primera se obtiene al filtrarla con un filtro pasa-bajas, mientras que la segunda se obtiene mediante un filtro pasa-altas. La imagen híbrida consiste en la suma de las dos imágenes anteriores.

Más allá de ser una simple ilusión óptica, las imágenes híbridas pueden tener diferentes aplicaciones en la vida real. Por ejemplo, pueden ser utilizadas con propósitos de privacidad (impedir que personas a cierta distancia puedan leer un texto), mostrar objetos o personas en diferentes momentos del tiempo en una sola imagen, etc. En el presente laboratorio se generaron imágenes híbridas según los conceptos propuestos por Oliva et al.

2. Materiales y métodos

Para construir la imagen híbrida se utilizaron dos imágenes (I_1 e I_2) de la cara de los dos miembros del grupo (anexo A, figura A.1). La primera imagen corresponde a la cara de Alejandro Posada y fue tomada recientemente específicamente para ser utilizada en el presente laboratorio. La segunda imagen corresponde a la cara Ramón Emiliani y, de igual manera, fue tomada con el objetivo de formar la imagen híbrida. Se implementó en Matlab un código para obtener la imagen deseada (anexo B). Primero se cargaron las imágenes y luego se recortó la de mayor tamaño con la función *imresize* para ajustarla al tamaño de la otra imagen (anexo B). Por último, se utilizó un filtro gaussiano (G) con $\sigma = 8$. Se eligió este valor de σ pues, tras utilizar diferentes valores, se encontró que con $\sigma = 8$ se obtenían los mejores resultados. Siguiendo la metodología propuesta por [2], se obtuvo la imagen híbrida H definida por:

$$H = I_1 \cdot G + I_2 \cdot (1 - G)$$

Para construir la pirámide se utilizó el código de James Hays [1] (anexo C). Este código está disponible en http://cs.brown.edu/courses/cs143/proj1/. Para obtener la pirámide, este código hace un downsampling progresivo a la imagen híbrida y finalmente concatena las imágenes.

3. Resultados

La imagen híbrida obtenida se muestra en la figura 1.

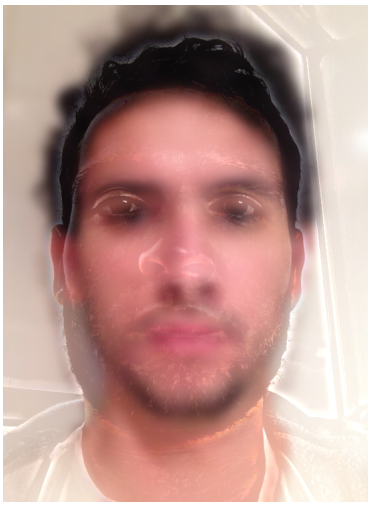

Figure 1: Imagen híbrida obtenida.

Esta imagen es la suma de la versión de baja frecuencia de la cara de Alejandro y la versión de alta frecuencia de la cara de Ramón. Por esta razón, al alejar la imagen se observa claramente la cara de Alejandro y, al observarla de cerca, se observa la cara de Ramón. En particular, de cerca se observan detalles y actividades de alta frecuencia como la barba, el pelo, los ojos y la nariz de Ramón. En este caso, como el método de filtrado es una convolución con una gaussiana, los resultados no son tan buenos como si se hiciera implementando una transformada de Fourier y la posterior eliminación de las frecuencias bajas. Al realizar el filtrado utilizando `fft2` o `dct2` en Matlab los bordes de la imagen de frecuencias altas se preservan mejor. Otro aspecto importante para resaltar es que para que la imagen híbrida se vea congruente es necesario que los ojos, la boca y el pelo (donde se encuentran la mayoría de las altas frecuencias) estén ubicados en la misma parte de la imagen y abarquen tamaños similares.

La figura 2 muestra la pirámide de 5 niveles de la imagen híbrida obtenida.

Figure 2: Pirámide de 5 niveles de la imagen híbrida.

Se observa que en los niveles bajos de la pirámide (principalmente en el primer y segundo nivel) predominan los contornos y detalles de alta frecuencia de la imagen de la cara de Ramón. A partir del nivel 3, estos detalles son prácticamente imperceptibles y predomina la forma de la cara de Alejandro.

4. Conclusiones

Las imagenes hibridas son la combinación de dos imagenes a partir de la div

Referencias

[1] J. Hays. Project 1: Hybrid images, 2017.

[2] A. Oliva, A. Torralba, and P. G. Schyns. Hybrid images. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 527–532. ACM, 2006.

Apéndices

A. Imágenes utilizadas

Figure A.1: Imágenes originales.

Figure A.2: Imagen recortada. Cabe resaltar que el recorte es muy pequeño: las dimensiones originales eran 788×549 y las finales 744×545.

B. Código: imagen híbrida

```
clc; clear all; close all;
sigma = 8; % Param. of the Gaussian radius

%Load and resize images
I1 = imread('alejandro_orig.png');
I2 = imread('ramon_orig.png');
I1_crop = imresize(I1, [size(I2,1), size(I2,2)]);
imwrite(I1_crop, 'alejandro_crop.png')

%Compute hybrid image
hybrid_im = I2-imgaussfilt(I2, sigma) + imgaussfilt(I1_crop, sigma);
imshow(hybrid_im)
```

C. Código: pirámide

```
function output = vis_hybrid_image(hybrid_image)
%visualize a hybrid image by progressively downsampling the image and
%concatenating all of the images together.

scales = 5; %how many downsampled versions to create
scale_factor = 0.5; %how much to downsample each time
padding = 5; %how many pixels to pad.

original_height = size(hybrid_image,1);
num_colors = size(hybrid_image,3); %counting how many color channels the input has
output = hybrid_image;
cur_image = hybrid_image;

for i = 2:scales
    %add padding
    output = cat(2, output, ones(original_height, padding, num_colors));

    %dowsample image;
    cur_image = imresize(cur_image, scale_factor, 'bilinear');
    %pad the top and append to the output
    tmp = cat(1,ones(original_height - size(cur_image,1), size(cur_image,2), num_colors), cur_image);
    output = cat(2, output, tmp);
end


%code by James Hays
```