

# Memoria

Gustavo Rivas Gervilla y Luis Suárez Lloréns

## 1 Nombre del proyecto

Implementar la creación de panoramas lineales con proyección en superficies cilíndricas o esféricas y mezcla de imágenes usando el algoritmo de Burt-Adelson.

## 2 Descripción del problema

Nuestro objetivo es crear panoramas. Esta aplicación se ha visto con anterioridad y se encuentra implementada en diversos dispositivos móviles — teléfonos inteligentes, cámaras digitales,... — lo cual hace parecer este proceso algo muy sencillo. Sin embargo, esta aplicación tiene varias dificultades que afrontar para que los resultados sean tan naturales como parecen en las soluciones que podemos encontrar en el mercado actualmente.

Por un lado, al trabajar con varias imágenes, necesitamos conocer sus posiciones relativas de una manera completamente automática. Con esto sabremos como colocar las imágenes para poder generar el panorama.

Por otro lado, hay que unir las imágenes. Debido a pequeñas variaciones, debidas a factores que varían desde la iluminación hasta posibles errores cometidos en la fase anterior, la superposición de ambas imágenes nos ofrece una imagen poco satisfactoria. La mezcla se debe hacer de una manera más suave, y para ello utilizaremos el algoritmo propuesto por Burt y Adelson.

Añadir a este proceso que la proyección que buscamos es sobre superficies cilíndricas o esféricas, lo que nos obliga a realizar alteraciones en las imágenes para conseguir este efecto.

## 3 Descripción de la solución

Vamos a separar este apartado en los principales problemas resueltos durante el desarrollo de este proyecto.

### 3.1 Proyección cilíndrica o esférica

La gran mayoría de imágenes, son representadas en un plano. Para conseguir obtener la proyección cilíndrica o esférica de dichas imágenes, tendríamos que ver en que punto del cilindro o esfera corta la línea que hay entre el punto donde se toma la imagen, y el punto de la proyección original —imagen original—.

Siendo  $(x_c, y_c)$  el centro de la imagen,  $f$  la distancia focal y  $s$  el llamado radio del cilindro o esfera, las fórmulas para la obtención de las proyecciones son:

#### 3.1.1 Proyección cilíndrica

La fórmula para la obtención de los valores es:

$$x' = s \tan^{-1} \frac{x - x_c}{f} + x_c$$
$$y' = s \frac{y - y_c}{\sqrt{(x - x_c)^2 + f^2}} + y_c$$

#### 3.1.2 Proyección esférica

La fórmula para la obtención de los valores es:

$$x' = s \tan^{-1} \frac{x - x_c}{f} + x_c$$
$$y' = s \tan^{-1} \frac{y - y_c}{\sqrt{(x - x_c)^2 + f^2}} + y_c$$

### 3.2 Construcción del panorama

Una vez tenemos las imágenes proyectadas, tenemos que reconocer donde estarán colocadas en nuestro panorama.

Para ello, hay que tener en cuenta que, al realizar una de estas dos proyecciones, un giro en la cámara se traduce en una traslación de las mismas. Por tanto, dentro de todo el espacio de posibles movimientos, vamos a considerar únicamente las traslaciones horizontales.

Con este espacio de búsqueda tan reducido, el cálculo del mejor desplazamiento se simplifica mucho. Aún así, debido a pequeños problemas de ajuste, preferimos usar BRISK/Fuerza bruta, para obtener un resultado mucho más preciso.

Una vez tenemos esto, podríamos simplemente superponer las imágenes. Esto da resultados poco deseables, pues en la imagen obtenida se podían ver claramente una imagen encima de la otra. Para realizar la mezcla de una manera más correcta, utilizamos el algoritmo propuesto por Burt y Adelson.

### 3.3 Algoritmo Burt-Adelson

Esta técnica pretende obtener la mezcla suave de dos imágenes. Para explicar esta técnica, vamos a suponer que tenemos dos imágenes que se superponen y queremos mezclar la parte izquierda de una con la derecha de la otra. Más tarde explicaremos como generalizar estas condiciones.

El problema de la mezcla de imágenes, como ya hemos comentado anteriormente, es que si simplemente tomamos los píxeles de cada imagen, sin ningún paso más, se generan bordes y otras aberraciones visuales que no queremos obtener. Una primera aproximación al problema sería promediar de alguna manera los píxeles centrales de ambas imágenes. Pese a ser una mejora del algoritmo anterior, también produce efectos no deseados en las imágenes.

Estos efectos depende del tamaño que hayamos seleccionado para la franja central que promediamos. Si la franja es pequeña, se sigue apreciando el borde, aunque suavizado. Si la franja es muy grande, se mezclan los detalles de las dos imágenes, deformando el resultado.

La solución propuesta por Burt-Adelson propone realizar una mezcla similar a la comentada anteriormente, pero separando primero las imágenes en diferentes bandas de frecuencias, mezclando las bandas y reconstruyendo la imagen. Esto nos ofrece unas imágenes mucho más convincentes ya que al tener imágenes con un espacio de frecuencias menor el tamaño de la banda usada es más adecuado.

Además, vamos a realizar una versión del algoritmo que permite realizar la mezcla utilizando una máscara para seleccionar qué píxeles queremos de cada imagen.

#### 3.3.1 Algoritmo

El algoritmo, a grandes rasgos, se compone de los siguientes pasos:

- 1) Construcción de las pirámides Laplacianas de las imágenes.

Como ya hemos comentado anteriormente, queremos realizar la mezcla de las imágenes para diferentes anchos de banda. Las pirámides Laplacianas son perfectas para esta tarea por dos motivos. Por un lado, divide la imagen en diferentes anchos de banda, paso fundamental para este algoritmo. Y por otro lado, nos facilita a realizar la mezcla, pues genera automáticamente la franja que vamos a mezclar para cada ancho de banda.

Veamos esta última propiedad. Los detalles más finos se almacenan en la capa más baja y extensa de la pirámide y la mezcla se realiza en una franja muy estrecha — en el algoritmo más básico propuesto por los autores, solo se promedia la columna central, dejando los valores de la imagen izquierda a la izquierda y los de la derecha a la derecha—. En capas superiores, se realiza de la misma manera, pero la columna central ya no representa una sola columna

de píxeles, si no que resume información de varias columnas de la imagen. Entonces, a mayor sea la capa que vayamos a mezclar, tenemos cada vez menos detalles — amplitud de onda mayor— y franjas donde mezclar también mayores en términos de la imagen final, justo lo que se quería.

#### 2) Construcción de la pirámide Gaussiana de la máscara.

Algo similar ocurre con la pirámide Gaussiana de la máscara. La usaremos en el siguiente paso para generar los pesos de cada imagen para cada pixel de una nueva pirámide Laplaciana que será nuestra imagen resultado. Destacar que la estructura de la pirámide Gaussiana — número de capas y tamaño de las mismas — encaja perfectamente con las pirámides Laplacianas del paso anterior.

#### 3) Mezcla de las pirámides Laplacianas.

Mezclaremos los valores de las pirámides Laplacianas de las imágenes, formando una nueva pirámide Laplaciana. Básicamente, realizaremos una media ponderada de ambas pirámides, usando la pirámide Gaussiana de la máscara como pesos para dicha media ponderada.

Notamos  $LA$  y  $LB$  a las pirámides Laplacianas de las imágenes originales,  $LS$  a la pirámide Laplaciana solución y  $GM$  a la pirámide Gaussiana de la máscara. Notamos como  $L_k(i, j)$  al valor de la posición  $(i, j)$  en el nivel  $k$ . Entonces, la fórmula usada es la siguiente:

$$LS_k(i, j) = GR_k(i, j)LA_k(i, j) + (1 - GR_k(i, j))LB_k(i, j)$$

#### 4) Reconstrucción de la imagen.

A partir de la pirámide Laplaciana generada en el paso anterior, podemos recuperar una imagen. Para ello simplemente usamos el modo en que se construye esta pirámide que es la diferencia entre gaussianas de distintos niveles. Entonces partiremos del nivel mayor de la pirámide que es por definición una gaussiana y lo iremos sumando, sucesivamente, con los niveles siguientes, para lo que será necesario una operación de expansión.

## 4 Resultados

A continuación vamos a mostrar los resultados de las distintas cuestiones que hemos ido resolviendo a lo largo del desarrollo de este trabajo.

En primer lugar vamos a ver las proyecciones cilíndricas de una imagen para ver simplemente el efecto que se produce. Para implementar esto no hemos tenido más que seguir las fórmulas que podemos ver en el Szal. y construir una nueva imagen copiando la anterior según nos marquen estas fórmulas. Aquí mostramos distintas proyecciones cambiando la distancia focal de la cámara, como es lógico a más lejos se encuentre la cámara más ancho será el cilindro

donde proyectamos.

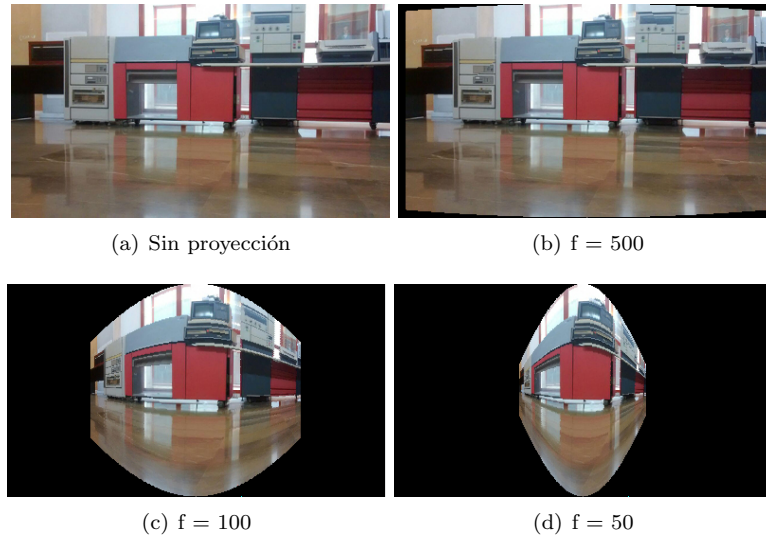


Figure 1: Ejemplo de proyección cilíndrica

Ahora vamos a entrar en el algoritmo de Burt-Adelson, este algoritmo se basa en la combinación de las pirámides laplacianas de dos imágenes según una cierta máscara binaria. Aquí podemos ver cómo hemos emulado el resultado que se muestra en el artículo de referencia que es la combinación de una naranja y una manzana. En primer lugar vemos lo que ocurre cuando no hacemos más que unir las dos mitades de las imágenes, como vemos el corte entre ambas es muy claro. En cambio en la segunda imagen podemos ver el resultado tras aplicar el algoritmo de Burt-Adelson, en el que claramente se ve cómo el corte entre ambas se ha suavizado.

Hay que señalar que hemos de tener cuidado con las imágenes de las que partimos puesto que la que teníamos poseía una línea negra en la parte superior y esto da problemas a la hora de los cálculos ya que produce números negativos que producirán errores que se propagarán a lo largo del cálculo de la laplaciana, con lo que hemos recortado las imágenes para evitar tal línea.

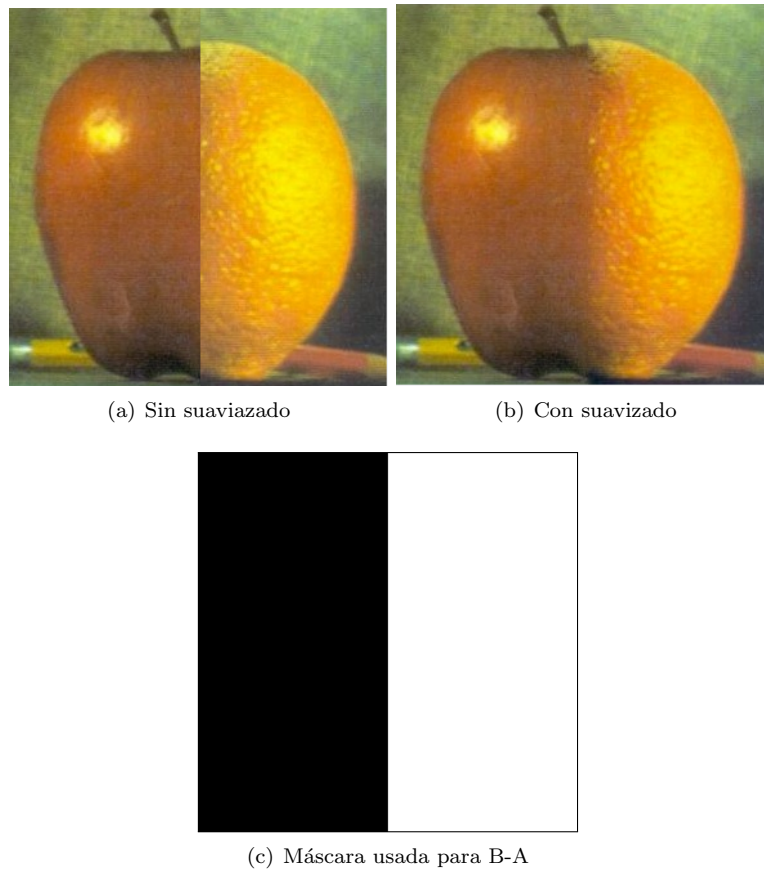


Figure 2: Ejemplo funcinamiento Burt-Adelson

Y por último vamos a ver la creación de panoramas con proyección cilíndrica y el algoritmo de Burt-Adelson, aquí realmente hemos obtenido unos resultados no muy buenos ya que las imágenes que hemos usado para tal propósito han sido tomadas a mano y como vamos a apreciar en las dos capturas que vamos a mostrar a continuación no hay mejora al aplicar Burt-Adelson:

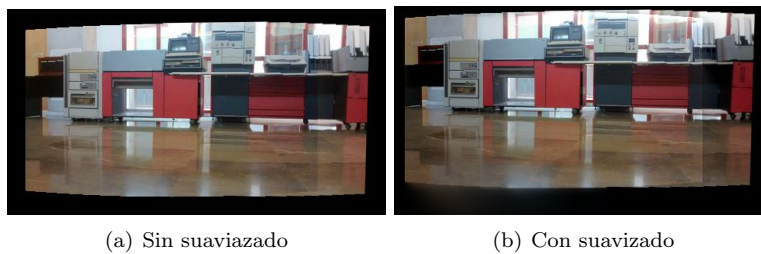


Figure 3: Ejemplo funcinamiento Burt-Adelson para la creación de panoramas

## 5 Posibles mejoras

La mayor crítica es al resultado general del panorama, no por la colocación de los elementos, si no por el resultado al integrar las imágenes. Ahora bien, este resultado es extraño, pues podemos ver que el algoritmo de Burt y Adelson funciona perfectamente, en el ejemplo de la manzana.

Podríamos haber creado funciones propias para buscar la traslación para los panoramas. Como ya dijimos en la sección sobre la construcción del panorama, al realizarse sobre imágenes cilíndricas o esféricas, la rotación de la cámara se transforma en una simple rotación. Aún así, por pequeños defectos como pueden ser un ajuste incorrecto y sobre todo, el no poder asegurar que las imágenes se han tomado exactamente en la misma posición y con únicamente rotación con respecto al eje vertical, acumulan errores que hace que no sea solo una traslación horizontal.

Cuando hacemos un panorama esférico, una rotación cualquiera se traduciría en un desplazamiento. Aunque quede fuera del alcance básico de los panoramas, podría ser un buen añadido.