

Inpainting

Pirámide Gaussiana

Visión Artificial

Universidad Pública de Navarra

Uxue Ayechu Abendaño

Rubén Sesma Redín

Índice

1 Introducción

2 Descripción del algoritmo

2.1 Creación de pirámide Gaussiana

2.1.1 Interpolación de bordes

2.1.2 Reducción

2.2 Ampliación con interpolación de la zona desconocida

3 Mejora del algoritmo

3.1 Interpolación de bordes mejorada

4 Resultados experimentales

5 Conclusión

1 Introducción

La **restauración de imagen** (del inglés inpainting) es un proceso que nos permite recuperar una parte deteriorada de la imagen o que tiene algún objeto que la oculta, con el objetivo de mejorar su calidad.

En este trabajo hemos desarrollado un algoritmo para resolver este problema descrito a continuación.

2 Descripción del algoritmo

El proceso está basado a grandes rasgos en crear la Pirámide Gaussiana interpolando los bordes de la zona desconocida hasta que esta zona desaparezca. Después se va ampliando la imagen sin zona desconocida para reconstruir la zona desconocida en cada nivel.

2.1 Creación de pirámide Gaussiana

La pirámide Gaussiana se crea a partir de la imagen original. El primer nivel de la pirámide se denomina G_0 , e inicialmente es una copia de la imagen original. Para conseguir el siguiente nivel de la pirámide denominado G_1 ...

Para conseguir el siguiente nivel de la pirámide, G_1 , el proceso que se sigue es: Interpolamos los bordes de la zona desconocida y reducimos, explicados a continuación.

2.1.1 Interpolación de bordes

La aproximación que hemos utilizado es, para cualquier pixel desconocido que a su lado tiene 2 pixeles conocidos, interpolamos su valor de la forma

$$G_0(i) = 2 G_0(i+1) - G_0(i+2)$$

Para realizar esta interpolación es necesario aplicar un filtro de suavizado a la imagen para que los valores $G(i+1)$ y $G(i+2)$ sean parecidos. Si no pasaríamos el filtro y la imagen tuviese ruido podría ocurrir que nos salieran valores fuera del rango.

Es decir se suaviza la imagen únicamente para interpolar los valores de los pixeles de los bordes de la zona desconocida, y el resto de pixeles mantenemos los originales.

En este momento se reduce la imagen con los bordes interpolados

2.1.2 Reducción

Cuando en un nivel de la pirámide hemos interpolado los bordes de la zona desconocida reducimos la imagen a la mitad de filas y columnas. En este algoritmo hemos usado el Operador de reducción local de la media, con cuidado de que si en un bloque a agregar existe algún pixel desconocido, hace la media de los pixeles conocidos, así conseguimos que la zona desconocida se vaya reduciendo

El proceso de creación de la pirámide Gaussiana finaliza cuando desaparece la zona desconocida



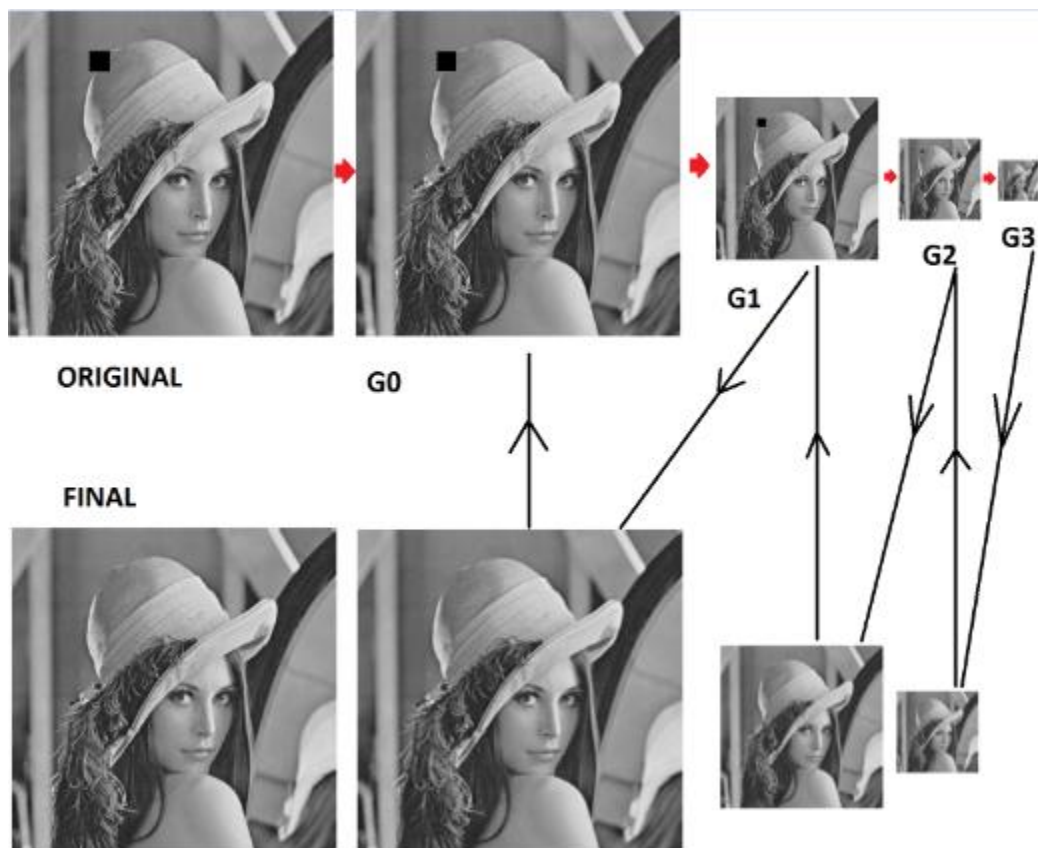
2.2 Ampliación con interpolación de la zona desconocida

Para este proceso hemos usado la función `imresize` que permite seleccionar el método de interpolación. En esta primera fase usaremos el más simple, `nearest`.

El proceso comienza en ampliando la cima de la pirámide Gaussian, en este ejemplo, G_3 del paso anterior que no tiene zona desconocida, la imagen ampliada la denominamos Aux, entonces Aux la usaremos para rellenar el trocito que nos falta en G_2 .

Lo siguiente es repetir el proceso, ahora ampliaríamos G_2 rellenada, y así hasta llegar al nivel más bajo de la pirámide y en el momento que rellenemos G_0 con el correspondiente trozo de G_1 ampliada termina el algoritmo y obtenemos la imagen final.

Una visión gráfica del algoritmo entero se puede apreciar en la siguiente imagen:



3 Mejora del algoritmo

Hay varias formas de mejorar el algoritmo usando técnicas de interpolación en los bordes de la zona desconocida mucho más sofisticadas

3.1 Interpolación de bordes mejorada

El método usado en la explicación anterior se puede refinar si tenemos en cuenta más puntos, nosotros lo hemos hecho para tres puntos. Para crear una función que pase por tres puntos hemos empleado la función polyfit. Podría hacerse para más puntos y podría mejorar la reconstrucción pero también empeorarla dependiendo del pixel a reconstruir. Hemos considerado que con tres puntos conseguimos una restauración lo suficientemente buena.

Interpolación simple: Valor interpolado es: $2 \cdot 100 - 80 = 120$

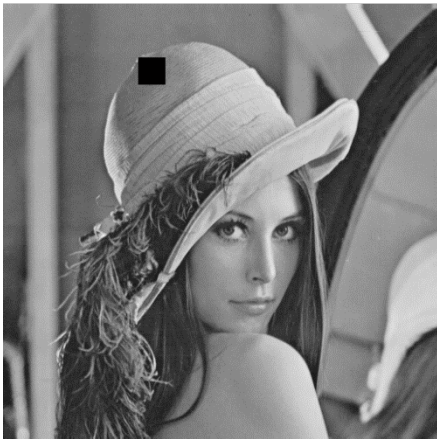
120	100	80
-----	-----	----

Interpolación mejorada: El valor interpolado es el 90, creado a partir de los puntos (100,1), (110,2) y (120,3). Lo que hacemos nosotros en nuestro algoritmo es pasarle los tres puntos contiguos al píxel desconocido (el más cercano será el 1 y el más lejano el 3) y la función polyfit devuelve una función de estimación realizada a partir de los tres puntos. Con polyval calculamos el valor de la función de estimación en el punto 0 que es el valor que queremos calcular. Como podemos apreciar en el ejemplo calcula el nuevo valor siguiendo la relación que tienen los anteriores valores, en este caso va disminuyendo de diez en diez luego es lógico que el valor de la posición 0 también lo haga.

90	100	110	120
0	1	2	3

4 Resultados experimentales

Zona desconocida bajo detalle: Gorro



Zona desconocida alto detalle: Ojo



Interpolacion bordes simple y ampliación nearest

MSE = 15.4875



MSE = 36.1524



Interpolacion bordes simple y ampliación bicubic

MSE = 14.6915



MSE = 34.1627



Interpolacion bordes polyfit y ampliación nearest

MSE = 42.9848



MSE = 69.9238



Interpolacion bordes polyfit y ampliación bicubic

MSE = 38.7212



MSE = 60.4621



Tabla comparativa de errores

	Interpolacion bordes simple		Interpolacion polyfit	
	Zona gorro	Zona ojo	Zona gorro	Zona ojo
Ampliacion nearest	15.4875	36.1524	42.9848	69.9238
Ampliacion bicubic	14.6915	34.1627	38.7212	60.4621

5 Conclusión

Como podemos observar en la tabla anterior, en cuanto al método de ampliación es mejor en todos los casos el bicubic.

Además en cuanto a las dos formas de interpolar en todas las zonas y en ambas ampliaciones es mejor el interpolado simple, aunque a la vista pueda parecer en algunos casos mejor el interpolado de polyfit. ¿Por qué? Porque con el interpolado simple no creamos nuevos píxeles sino que a partir de dos que tenemos calculamos el tercero. Sin embargo, en el método de interpolado de polyfit lo que sucede es que calculamos el nuevo valor intuyendo cómo evolucionará la imagen. Esto puede fallar (o aportar mayor MSE) debido a que los píxeles no siempre siguen un orden exacto de crecimiento o decrecimiento respecto a sus vecinos.