

# Informe Tarea 3 -Restauración de imágenes

Kaori Kanno

Pontificia Universidad Católica de Chile

Las imágenes pueden presentar cierto nivel de degradación o de ruido. Restaurarlas resulta de importancia para reconstruir el pasado. En esta tarea se degradaron dos imágenes y a partir de la matriz de degradación se trató de restaurar la imagen original de distintas formas, así como mejorar la resolución de una tercera imagen.

## Solución propuesta

En esta tarea se nos entregaron entonces 3 imágenes distintas, dos de las cuales estaban en alta resolución, y una tercera que debido a la lejanía de la toma tenía un texto borroso. Para cada imagen se nos pidieron distintas tareas:

Para la primera imagen se nos pidió hacer una degradación horizontal y luego restaurarla. Se ocupó el kernel  $h$ , y con este se construyó la matriz  $H$ . Para construir  $A$  se utilizaron los colabs vistos en clases y se fue cambiando  $W$  según lo que se pedía, y también se calculaba el error respecto a la imagen original, los cuales en general fueron pequeños. Para Fourier, debido al peso de RAM que exigía la matriz  $D$ , solo se usó  $W = K$  para aplicar directamente el filtro buttersworth, el cual se construyó en base al filtro dado en clases. Pero también se probó aplicar directamente el filtro en el espacio de fourier de la imagen para restaurarla, sin embargo en este segundo proceso la imagen mantenía el tamaño de la imagen degradada, por tanto no se podía comparar el error de restauración. Ambos procesos mejoraron la imagen.

Para la segunda imagen se construyó una matriz  $H$ , que para valores  $m$  y  $n$  pequeños se construye como se nos mostró en clases, sin embargo este no funcionaba exactamente como se esperaba, por lo que probablemente  $H$  posee algún error, lo cual se aprecia en los bordes de cada segmento que se generó en la imagen y en cómo se restauraron los segmentos de los bordes de esta. Para mejorar los errores de la imagen, la fragmentación de  $30 \times 30$  se tomó superponiendo píxeles de un fragmento y otro, de modo de al reconstruir la imagen final, recortar estas superposiciones y que la imagen quede sin tanto borde negro como ocurría al inicio. Además se agregó un suavizado justo en los bordes que debería mejorar la calidad de la imagen. Se agregó además un padding a los lados derecho y abajo, pero esto no mejoró la imagen.

Finalmente para la última imagen se aplicaron filtros para mejorar la imagen. Se hicieron muchos intentos, los mejores dos son el primer y tercer intento, siendo este tercero una mezcla entre el primero y el tercero. Algunos de estos filtros

asumen que la imagen tiene cierto kernel de degradación que se trató de estimar aquí.

## Experimentos realizados

Para la parte 1, se trató de construir la matriz  $D$ , sin embargo la memoria RAM no fue suficiente para esta tarea, por lo que se optó por otros métodos que se muestran en la tarea.

Para la parte 2, se hicieron distintos intentos para construir  $H$ , siendo la última versión la que funcionaba mejor con matrices de pequeñas dimensiones para poder visualizarlas correctamente, por tanto el error podría ser algo que se visualiza mejor para matrices grandes, pero no pude encontrar el error. No parece estar el error en la reconstrucción de la imagen ni la fragmentación de esta. Traté de importar la función pedida del kernel gaussiano, mas no logré ejecutarla allí, pues no estaba numpy definido, por lo que tuve que copiar y pegar la función. Se presenta además otro intento de solución que se tuvo, donde visualmente la imagen se ve más degradada, pero el tamaño de la imagen se preserva mejor.

Para la parte 3 se hicieron muchos intentos distintos por mejorar la imagen como se puede apreciar. Se trató por ejemplo con un filtro pasa altos el fourier, lo cual empeoró la legibilidad de la imagen, lo mismo con otros tipos de filtros pasa altos. Se intentó entonces con filtros pre-hechos, a los que se les introducía un kernel que simulaba la degradación de la imagen, y se fue jugando con estos para tratar de simular la misma degradación que tendría la imagen. Como se ve en el segundo, tercer intento y quinto intento, se usó la técnica de ecualización de histogramas, lo cual mejoraba un poco la imagen. Estos filtros fueron uno laplaciano, que tuvo el mejor desempeño, y otro una gaussiano, que funcionaban simulando que la degradación era de ese tipo. Finalmente se probó con una interpolación para cambiar un poco la resolución de la imagen, lo cual tampoco ayudó mucho. Además comentar que a veces la imagen se descarga en blanco y negro y otras veces en 3 dimensiones. Mi código funciona bien para cuando se descarga en la primera modalidad.

## Conclusiones

En esta tarea se aprendió sobre restauración de imágenes ante las distintas situaciones que uno tenga que enfrentar. Si bien no se lograron hacer todas las restauraciones a la perfección, se aprendió mucho sobre el proceso detrás de estos.