

Aplicaciones de Teoría de la Información al Procesamiento de Imágenes (ATIPI 2020)

Evaluación—Parte práctica: Denoising

Consideraciones generales

- El trabajo consiste en la implementación, en software, de algoritmos de denoising de imágenes de tono continuo.
- Los programas deberán cumplir con los requisitos siguientes:
 - El lenguaje de programación debe ser C++ o C.
 - Los experimentos solicitados en el trabajo deben ser compilables, ejecutables, y reproducibles por los docentes en plataformas de computación genéricas (PC con Windows o Linux, Mac, etc.), sin requerir paquetes especiales. La entrega debe incluir instrucciones claras de cómo compilar y ejecutar los programas (incluyendo archivos makefile, proyectos de MS Visual Studio, etc., apropiados a la plataforma).
 - Las imágenes de entrada y salida serán en formato PGM (8bpp, escala gris, modo P5) o PPM (24bpp, colores, modo P6). Ver descripción del formato en http://en.wikipedia.org/wiki/Netpbm_format .
 - Los experimentos requeridos se harán sobre un conjunto de imágenes estándar que estarán disponibles en la página web del curso. El conjunto incluye imágenes “ruidosas” y también las imágenes originales “limpias,” que se utilizarán *exclusivamente* para evaluar el desempeño de los algoritmos. Esta evaluación se hará a través de utilitarios independientes de los programas de denoising, que no deberán tener *en ningún momento* acceso a las imágenes limpias.
 - Las decisiones de diseño e implementación que influyen sobre el rendimiento de los programas son importantes y serán evaluadas. Los programas deberán permitir la realización de los experimentos requeridos, sobre las imágenes provistas, en tiempo razonable (medido en segundos).
- En la letra del proyecto se especifica la funcionalidad mínima, y los experimentos de prueba requeridos. Decisiones de diseño que no estén definidas explícitamente quedan a criterio del programador. Mejoras en funcionalidad, si son interesantes y efectivas, o experimentos interesantes adicionales, recibirán puntaje extra.
- La entrega del proyecto se compone de
 1. Un paquete comprimido (formato zip o tgz) con los archivos fuente y ejecutables de los programas, incluyendo instrucciones para su compilación y ejecución.
 2. Un informe en formato pdf conteniendo: descripción de los programas (incluyendo instrucciones de uso, decisiones de diseño, y detalles que no se desprendan obviamente de la letra del proyecto), resultados de los experimentos y su discusión.

- *Está prohibido* bajar soluciones de cualquier tipo, totales o parciales, de internet, o copiarlas de ejercicios similares en ediciones anteriores del curso.

Tenga en cuenta que transgresiones a estas prohibiciones son relativamente fáciles de identificar—existen herramientas sofisticadas de dominio público que permiten detectar con gran confiabilidad la copia, con o sin alteraciones, de trabajos existentes, sea de internet o de otras fuentes. De constatarse dichas transgresiones, serán severamente sancionadas.

Proyecto A12: Denoising con prefiltro para el canal de sal y pimienta (S&P)

Primera parte

Se implementará un denoiser basado en el algoritmo DUDE para imágenes de tono gris (8 bpp, con valores z enteros en el intervalo $0 \leq z \leq 255$), afectadas por ruido S&P con parámetro δ . El denoiser tendrá las siguientes características:

- A efectos de construir contextos para muestras en los bordes de la imagen, se supone que toda muestra que cae fuera de la imagen tiene valor 128.
- Se utilizarán contextos de 3×3 como se indica en la figura siguiente:

NW	N	NE
W	z	E
SW	S	SE

En la figura, z es la muestra modelada, y $N, E, S, W, NW, NE, SW, SE$ son sus vecinos inmediatos.

- Los contextos se cuantificarán de la manera siguiente: dado un contexto C se calcula un *extracto* $f(C)$, que es un vector binario de $4 + k$ bits, donde k es un entero cuyo valor en el intervalo $0 \leq k \leq 8$ se especificara como parámetro del programa. El vector $f(C) = f_1 f_2 f_3 f_4 f_5 \dots f_{4+k}$ ($f_i \in \{0, 1\}$) se construye como sigue:
 1. f_1, f_2, f_3, f_4 se obtienen tomando el resultado de las comparaciones $N > A, E > A, S > A$ y $W > A$, respectivamente, donde A es el promedio redondeado

$$A = \left\lfloor \frac{N + NE + E + SE + S + SW + W + NW}{8} + \frac{1}{2} \right\rfloor.$$

2. Los bits $f_5 \dots f_{4+k}$ son los k bits más significativos de A , o sea

$$f_5 \dots f_{4+k} = \left\lfloor \frac{A}{2^{8-k}} \right\rfloor$$

(cuando $k = 0$ el extracto es simplemente $f_1 f_2 f_3 f_4$, sin la segunda parte).

- $f(C)$ se utiliza como índice en una tabla de 2^{4+k} histogramas condicionados $N(z|f(C))$ (donde $N(a|\mathbf{f})$ es el número de veces que observamos $z = a$ cuando $f(C) = \mathbf{f}$), de los cuales se obtendrán distribuciones condicionales $P(z|f(C))$ que utilizaremos en DUDE como estimaciones de $P(z|C)$.

- Excepto por la cuantificación de los contextos, el resto del algoritmo, incluyendo la regla de denoising, es idéntico al DUDE original.
- Recuerde que la regla de denoising para el canal S&P, cuando $z_i = 0$, es

$$\hat{x}_i = \left\lceil \frac{\delta}{2(1-\delta)n_0} \left(\sum_{i=0}^{M-1} i n_i - \frac{n\delta}{2}(M-1) \right) \right\rceil,$$

donde n_i es el número de veces que ocurrió el símbolo i en el contexto C , $n = \sum_i n_i$, y $\lceil x \rceil$ denota redondeo de x al entero más cercano en $[0, 255]$. La regla para $z_i = M-1$, es, como se vio en el ejercicio práctico, simétrica a la anterior.

Segunda parte

- Se experimentará con prefiltrado e iteración.
- Se agregará al denoiser de la primera parte un argumento más de entrada, que especificará una imagen prefiltrada de la cual se tomarán las muestras para la construcción de contextos (recuerde que los valores que se cuentan en los histogramas $N(z|f(C))$ siguen siendo de la imagen ruidosa de entrada, mientras que C se toma de la imagen prefiltrada).

Se recomienda diseñar el programa desde el principio con el argumento adicional de la imagen prefiltrada. En la primera parte, este argumento se hace igual a la imagen ruidosa, o se omite, y se toma el nombre de la imagen ruidosa como su valor por defecto.

- Como prefiltro inicial se utilizará el mismo denoiser de la primera parte.

Tareas

1. Implementar el denoiser para el canal S&P. El programa tendrá como argumentos el parámetro δ del canal, el parámetro k del cuantificador de contextos, el nombre de un archivo (en formato PGM) con la imagen ruidosa, y el nombre de un archivo en el que se almacenará la imagen “limpiada” (denoised). Para los experimentos con prefiltro, el programa acepta un argumento adicional con el nombre de la imagen prefiltrada. Si este argumento se omite, se toma como valor por defecto la imagen ruidosa original.
2. Para este inciso, se utilizará un utilitario que compara dos imágenes de las mismas dimensiones en formato PGM e imprime su distancia como PSNR en decibeles. El utilitario (`diffpnm`) estará disponible para descargar de la página web del curso.

Para cada una de las imágenes ruidosas del conjunto de prueba, aplicar el denoiser y tabular los resultados siguientes:

- El PSNR de la imagen ruidosa respecto a la imagen (limpia) original.
- El PSNR de la imagen limpiada respecto a la imagen original, para cada uno de los valores de $k = 0, 1, 2, 3, 4, 5, 6, 7, 8$.

3. Agrupar los resultados por imagen original y nivel de ruido, e indicar el valor óptimo de k para cada imagen ruidosa.
4. Graficar el PSNR con respecto a k para una de las imágenes originales, y los niveles de ruido $\delta = 0,10$ y $\delta = 0,35$ (ambas curvas en la misma gráfica).
5. Discutir los resultados, y en particular explicar variaciones del valor óptimo de k respecto al nivel de ruido y el tamaño de la imagen.
6. Prefiltrado e iteración: Para cada imagen y valor de δ , se utilizará como imagen prefiltrada inicial la salida de DUDE obtenida en el inciso 3, con el valor óptimo de k . El mismo valor de k se utilizará para las iteraciones. Correr como mínimo 10 iteraciones, utilizando siempre como imagen prefiltrada la salida de DUDE de la iteración anterior.
7. Tabular los mejores resultados, y comparar con los resultados obtenidos anteriormente sin utilizar prefiltro. Incluir en la tabla los valores siguientes:
 - PSNR de la imagen ruidosa.
 - PSNR de la imagen producida por DUDE sin prefiltro.
 - PSNR de la imagen producida por DUDE con prefiltro e iteraciones (utilizar el número de iteraciones que haya dado el mejor PSNR—es posible que este número sea cero).
8. Observando cómo varía el PSNR con el número de iteraciones, comente qué problema práctico hay en el uso de iteración con el denoiser tal como fue implementado. ¿Qué soluciones posibles hay para ese problema? (No tiene que implementar la solución.)