

Aplicaciones de Teoría de la Información en Procesamiento de Imágenes (ATIPi 2018)

Proyecto de Evaluación

Consideraciones generales

- El trabajo consiste en la implementación de algoritmos de compresión y descompresión de imágenes en software.
- 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 de grises, modo P5). Ver descripción del formato en http://en.wikipedia.org/wiki/Netpbm_format.
 - Los experimentos requeridos se harán sobre un conjunto de imágenes que estarán disponibles en la página de web del curso.
 - Las decisiones de diseño e implementación que influyen sobre el rendimiento de los programas son importantes y serán evaluadas. Los programas deberán ser suficientemente eficientes para permitir la realización de los experimentos requeridos, sobre todas las imágenes provistas, en tiempo razonable (como máximo unos pocos segundos incluso para las imágenes más grandes). Una vez depurado el programa, y para realizar los experimentos, *recuerde compilar con máxima optimización del tiempo*: opción -O3 con gcc/g++ en Linux/Mac, opción /Ot o versión Release con Visual en Windows. *No utilice la versión Debug (u opción -g en Linux/Mac) en los experimentos.*
- En la letra del proyecto se especifica la funcionalidad mínima requerida. Decisiones de diseño que no estén definidas explícitamente quedan a criterio del programador (p.ej. los detalles del encabezamiento del archivo comprimido, opciones del programa, etc.). Mejoras en funcionalidad, si son interesantes y efectivas, recibirán puntaje extra.
- La entrega del proyecto será un paquete comprimido (formato zip, tar.gz, o tgz) con el contenido siguiente:
 1. Una copia del enunciado (este documento).

2. Una carpeta con los archivos fuente y ejecutables de los programas, incluyendo instrucciones para su compilación y ejecución, y especificando bajo qué sistema operativo y con qué compilador fueron desarrollados.
 3. 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), respuestas a preguntas formuladas en este enunciado, 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 LE12: Compresión de imágenes de escala de grises con modo de run

Se implementarán un compresor y descompresor sin pérdidas para imágenes de escala de grises (8 bpp). El algoritmo de compresión es una variante de LOCO-I, con los elementos siguientes:

- De no indicarse otra cosa, se supone que toda muestra que cae fuera de la imagen (para predicción o modelado) tiene valor cero.
- El predictor y modelo de contextos estarán basados en la misma plantilla causal que se utiliza en JPEG-LS.

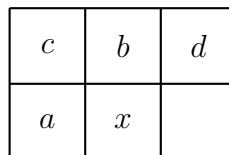


Figura 1: Configuración de contexto.

Nos referimos a la configuración de contexto con valores a, b, c, d como C (esta excluye a x). Se utiliza el mismo predictor fijo MED(a, b, c) utilizado en JPEG-LS. El predictor no tiene parte adaptativa.

- **Modelo de contextos.** Dado el contexto C , se deriva un *extracto* $f(C)$, que es un número calculado de la manera siguiente. Se definen

- *Valor predicho:* $\hat{x} = \text{MED}(a, b, c)$.
- *Bits de textura:* $(c > \hat{x}), (b > \hat{x}), (a > \hat{x})$, interpretados como bits (1 si la comparación es cierta, 0 si es falsa), que se concatenan formando un número binario T , $0 \leq T \leq 7$.
- *Nivel de actividad:* $X = |c - \hat{x}| + |b - \hat{x}| + |a - \hat{x}|$.

P1. *Verifique que 10 bits alcanzan para representar cualquier valor de nivel de actividad en binario.*

Sea s un entero en el rango $0 \leq s \leq 10$, que será parámetro del compresor. El nivel de actividad X se cuantiza a $Q(X)$, definido como sigue:

$$Q(X) = \left\lfloor \frac{X}{2^{10-s}} \right\rfloor.$$

$Q(X)$ es el número formado por los s bits más significativos de X , cuando se toma a este último como un número de 10 bits.

El extracto del contexto C se define como

$$f(C) = 8 \cdot Q(X) + T.$$

Este a su vez define la clase de contextos a la que pertenece C : todos los contextos que derivan el mismo extracto $f(C)$. Utilizaremos a los números $f(C)$ como los “nombres” (o etiquetas) de las clases de contextos, refiriéndonos, por ejemplo a “la clase $f(C)$ ”.

P2. *Determine el rango de valores que puede alcanzar $f(C)$ en función de s , y en consecuencia, el número de clases de contexto.*

El extracto $f(C)$ se utiliza para indexar una estructura de datos donde se asocian a $f(C)$ dos estadísticas:

- N : número de muestras vistas en contextos de la clase $f(C)$,
- A : suma de valores absolutos de errores de predicción en contextos de la clase $f(C)$.

Las estadísticas A y N se utilizan, al igual que en los contextos regulares de JPEG-LS, para determinar el parámetro k de un código de Golomb PO2 (se ignora el trato especial de $k = 0$ en JPEG-LS).

- El código determinado en el inciso anterior se utiliza para codificar los errores de predicción que ocurren en contextos de la clase $f(C)$, luego de ser llevados al rango no negativo por medio de la función

$$M(e) = \begin{cases} -2e + 1 & e < 0, \\ 2e & e \geq 0, \end{cases} \quad (1)$$

No se utiliza la función alternativa $M'(e) = M(-1 - e)$ mencionada en el curso. Tampoco se requiere implementar la reducción de errores de predicción módulo 256 al rango $-128 \leq e \leq 127$.

- Se fija un parámetro entero $R = 128$ (*reset*), que se utilizará en la actualización de las estadísticas. El mecanismo donde se utiliza este parámetro mejora la tasa de compresión, y está también en JPEG-LS.

Al igual que JPEG-LS, el compresor puede operar en dos modos distintos: *modo regular* y *modo de run*. A diferencia de JPEG-LS, en este proyecto podremos desactivar el modo de run a fin de comparar el desempeño del compresor con y sin el modo activado.

Modo regular. En el codificador, las muestras se procesan, y las estadísticas se actualizan de la manera siguiente:

1. Para cada clase $f(C)$, se inicializan las estadísticas correspondientes a $f(C)$ con los valores

$$\begin{aligned} N &= 1, \\ A &= 8. \end{aligned}$$

2. Procesamiento de la muestra x_i (suponiendo un ordenamiento lineal “raster” de las muestras de la imagen y que no se está procesando ni se pasa al modo de run descrito más abajo):

- a) Se determina el contexto C , y se calcula el valor predicho \hat{x}_i .
- b) Se calcula el extracto $f(C)$.
- c) Se determina el parámetro k en función de A y N como en los contextos regulares de JPEG-LS. Se codifica $x_i - \hat{x}_i$ con el mapeo (1) y el código de Golomb PO2 de parámetro k .
- d) Actualización de estadísticas:
Si $N = R$,

$$\begin{aligned} N &\leftarrow N/2, \\ A &\leftarrow A/2. \end{aligned} \tag{2}$$

Las operaciones en (2) son divisiones enteras que descartan el resto, tal como se interpretan en C/C++. Para cualquier N (independientemente de si se efectuó la actualización anterior):

$$\begin{aligned} N &\leftarrow N + 1, \\ A &\leftarrow A + |x_i - \hat{x}_i|, \end{aligned}$$

- e) Si hay más muestras a procesar, se vuelve al paso 2a para la muestra x_{i+1} .

El decodificador predice, calcula extractos de contextos (con el mismo valor de s que utilizó el compresor), y actualiza sus estadísticas de la misma manera, luego de decodificar cada error de predicción y reconstruir la muestra x_i .

- **Modo de run.** El modo de run se habilita o no a través de un argumento en la línea de comando del compresor. En caso de estar habilitado, se entra a este modo cuando los píxeles del contexto satisfacen las igualdades $a = b = c = d$. Cuando esto ocurre, el compresor cuenta la cantidad de veces seguidas que se repite el símbolo a , a partir del lugar marcado como x en la Fig. 1. Este número (el *largo de run*, que puede ser cero o positivo) se codifica con un código de Golomb PO2 de *parámetro fijo* $k = 3$. El run se interrumpe cuando se encuentra un símbolo distinto de a , o se llega al final de la fila en la imagen. Luego de codificar el largo de run, se retorna al modo regular.

P3. *Verifique que la condición $f(C) = 0$ es necesaria pero puede no ser suficiente para entrar al modo de run.*

El compresor le indicará al descompresor si el modo de run está habilitado o no al principio del archivo comprimido. En caso de estar habilitado, el descompresor detecta la entrada al modo sin necesidad de ninguna indicación adicional de parte del compresor—cuando está por procesar la muestra x , el descompresor ya conoce los valores de a , b , c , y d , y puede determinar la condición de entrada al modo. Cuando la condición se cumple, el descompresor interpreta los próximos bits de la entrada como un código de Golomb PO2 con $k = 3$, decodifica el largo de run, y emite ese número de copias del valor a .

Tareas:

1. Incluya en el reporte sus razonamientos y desarrollos para las sugerencias P1–P3.
2. Implementar el compresor y el descompresor correspondiente. Detalles:
 - El compresor recibe los argumentos siguientes en la línea de comando:
 - El valor de s (entero).
 - Un valor binario (0/1) indicando si se activa el modo de run.
 - Nombre del archivo con la imagen de entrada.
 - Nombre del archivo de salida donde se almacenará la imagen comprimida. El compresor almacenará en el cabezal de este archivo de salida el valor de s y el indicador de modo de run utilizados.
 - El descompresor, por su parte, recibe en la línea de comando los nombres del archivo comprimido y del archivo donde se almacenará la imagen reconstruida, y lee el valor de s y el indicador de modo de run del cabezal del archivo comprimido.
 - Recuerde que los archivos comprimidos *no son imágenes*, y por lo tanto no deben usar los formatos PGM/PPM. Defina Ud. un formato mínimo razonable que incluya un cabezal de unos pocos bytes al principio para transmitir parámetros.
 - El compresor y descompresor deben ser programas separados. Sin embargo, pueden (y deben) compartir partes del código con funcionalidad idéntica (p.ej., determinación de contexto y $f(C)$, predicción, determinación del parámetro de Golomb, etc.).

- Si se intentan ejecutar sin argumentos, o con cantidad insuficiente de argumentos, los programas deben imprimir un mensaje de error, y una descripción de la línea de comando esperada.
 - Tanto el compresor como el descompresor deben reportar los tamaños de archivos de entrada y salida (excluyendo cabeceras), y la tasa de compresión en bits por símbolo de 8 bits.
3. Comprimir y descomprimir cada una de las imágenes en el conjunto de prueba (solo las de escala de grises, 8bpp), probando los valores $s = 4, 5, 6, 7, 8, 9, 10$, una vez con el modo de run habilitado y otra vez sin habilitar. Verificar en cada caso que la compresión y descompresión funcionan correctamente (sin pérdidas), y explicar cómo se hizo la verificación.

Sugerencia: puede resultar útil el utilitario `diffpnm`, que estará disponible para descargar de la página del curso. El utilitario compara dos imágenes de las mismas dimensiones en formato PGM o PPM, e imprime la distancia entre ellas como PSNR. En particular, indica si las imágenes son idénticas (las imágenes podrían ser idénticas aunque los archivos no lo sean, dada la flexibilidad existente en la sintaxis del cabezal PGM/PPM).

Tabular las tasas de compresión (en bits/píxel) para cada una de las combinaciones de imagen, valor de s , y opción de modo de run (incluir en la tabla una columna para tasas con el modo habilitado, y otra para tasas con el modo sin habilitar).

4. Para cada valor de s que se haya probado, calcular el promedio simple de las tasas de compresión obtenidas para todos los archivos (con el modo de run habilitado). El promedio calculado se utilizará como medida de desempeño global del algoritmo de compresión sobre todo el conjunto de imágenes. Tabular y graficar el promedio en función de s . En la tabla, indicar el valor de s que produce el mejor promedio de tasa de compresión.
5. Explicar la función del parámetro s , y la forma de la curva graficada en el inciso 4. Recomendar un valor por defecto para s explicando la recomendación.
6. Discutir el impacto del modo de run, observando para qué imágenes tiene impacto positivo, y para cuáles puede tener impacto negativo.
7. Comparar las tasas de compresión obtenidas con el parámetro s recomendado en el inciso 5, y con modo de run habilitado, con las obtenidas con JPEG-LS. Datos de tasas de compresión con JPEG-LS para las imágenes de prueba están disponibles en la página de web del curso. Explicar los resultados de la comparación, especialmente en casos en que se registren diferencias grandes en la tasa de compresión.

Otras sugerencias:

- Con la única excepción del cálculo de tasas de compresión, los programas no deben utilizar variables de punto flotante, ni llamar a funciones de librería de punto flotante (como `log()`, `pow()`, `floor()`, `ceil()`, etc.). Todas las operaciones necesarias para el compresor y descompresor pueden (y deben) hacerse con enteros.

- Los píxeles de las imágenes se leen de la entrada como caracteres sin signo, y si se almacenan partes significativas de imagen (p.ej. líneas de imagen) o imágenes completas en memoria, se debe preservar ese tipo. Sin embargo, al operar sobre píxeles individuales, es correcto promoverlos a enteros de tipo más largo (p.ej. `int`), asegurando que operaciones con signo se ejecuten correctamente.