Reducción de imágenes utilizando la descomposición quadtree

Aranzazu Jurio, Daniel Paternain, Edurne Barrenechea, Cedric Marco-Detchart, and Humberto Bustince

Universidad Publica de Navarra
Campus de Arrosadía s/n, 31006, Pamplona (España)
{aranzazu.jurio, daniel.paternain, edurne.barrenechea, cedric.marco, bustince}@unavarra.es

Resumen En este trabajo presentamos una mejora sobre un algoritmo de reducción propuesto en [3] que utiliza intervalos y operadores K_α. En la nueva propuesta utilizamos una descomposición quadtree para seleccionar el mejor valor de α en cada área de la imagen, dependiendo de sus propiedades de homogeneidad. Además, probamos de forma experimental que nuestro método obtiene mejores resultados que el original, con una media de mejora del 20.20 % en MSE.

Keywords: Reducción de imagen, ampliación de imagen, descomposición quadtree, medida de homogeneidad

1. Introducción

La reducción de imágenes consiste en disminuir la resolución de una imagen manteniendo tantas propiedades como sea posible de la imagen original. Estas propiedades pueden ser: intensidad, contraste, textura, bordes, entre otras. La reducción de imagen puede ser utilizada como parte del preprocesamiento cuando vamos a aplicar algoritmos que son computacionalmente costosos, ya que la reducción de la resolución permite reducir su coste de ejecución. Además, también puede ser utilizada para reducir el coste de almacenamiento de una imagen. En estos casos, los algoritmos de reducción se asocian con otros de reconstrucción (o ampliación) que obtienen, a partir de la imagen reducida, una nueva imagen en su dimensión original [1,3,4,7].

En este trabajo nos basamos en el método propuesto en [3] para reducir imágenes. Este algoritmo consta de tres pasos:

- primero, dividir la imagen en bloques disjuntos (por ejemplo, en bloques de tamaño 2×2 píxeles);
- segundo, asociar un intervalo a cada bloque;
- tercero, aplicar un operador K_α [2] para obtener un sólo valor que represente a todo el bloque.

Sin embargo, en el trabajo original no se estudió el proceso de elección del mejor K_α para cada bloque de la imagen, ya que el coste computacional de este proceso

puede llegar a ser muy elevado. Basándonos en este algoritmo, el objetivo de este trabajo es mejorar los resultados obtenidos por el algoritmo de reducción (en términos de Error Cuadrático Medio - MSE) utilizando un método eficiente de hallar, para cada área de la imagen, el mejor operador K_α teniendo en cuenta las propiedades de homogeneidad del área concreta.

Nuestro algoritmo para encontrar esos valores óptimos de α consta de dos pasos:

- en el primero, dividimos la imagen en áreas de tamaño variable utilizando una descomposición quadtree. Este tipo de descomposición permite, de forma recursiva, dividir la imagen en zonas de intensidad homogénea, es decir, en zonas en las que las intensidades de todos sus píxeles tienen valores similares.
- el segundo paso consiste en encontrar, para cada área homogénea, el mejor operador K_α. De esta forma, aplicamos el mismo operador K_α a cada uno de los bloques pertenecientes a esa área.

Gracias a la descomposición quadtree, el coste computacional de nuestra propuesta es menor que seleccionar para cada bloque el mejor K_α, ya que cada una de nuestras áreas engloban varios bloques. Además, comprobamos experimentalmente que nuestros resultados mejoran significativamente los obtenidos por el algoritmo original. De media, la mejora está en torno al 20 %.

El resto de este trabajo se organiza de la siguiente manera: en la Sección 2 recordamos algunos conceptos preliminares. En la Sección 3 explicamos el algoritmo en que nos hemos basado, así como nuestra propuesta de reducción de imágenes. En la sección 4 mostramos un estudio experimental del algoritmo y finalizamos, en la Sección 5, con algunas conclusiones y líneas futuras.

2. Preliminares

En este trabajo, representamos una imagen Q de $M \times N$ píxeles en escala de grises (con L niveles de gris) como una matriz de M filas y N columnas. La intensidad normalizada del píxel situado en la i-ésima fila y en la j-ésima columna se denota por q_{ij}, y se obtiene dividiendo la intensidad de dicho píxel entre $L-1$, por lo que $0 \leq q_{ij} \leq 1$.

A la hora de trabajar con intervalos, denotamos por $L([0,1])$ el conjunto de todos los subintervalos cerrados del intervalo unidad $[0,1]$, es decir, $L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | 0 \leq \underline{x} \leq \overline{x} \leq 1\}$.

El operador K_α [2] permite asociar un valor real en $[0,1]$ a un intervalo, mediante una combinación convexa de sus extremos.

Definición 1 *Sea $\alpha \in [0,1]$. El operador $K_\alpha : L([0,1]) \to [0,1]$ se define como una combinación convexa de los extremos de su argumento, dada por*

$$K_\alpha(\mathbf{x}) = \underline{x} + \alpha(\overline{x} - \underline{x})$$

para todo $\mathbf{x} \in L([0,1])$.

Es fácil comprobar que $K_0(\mathbf{x}) = \underline{x}$ y $K_1(\mathbf{x}) = \overline{x}$ para todo $\mathbf{x} \in L([0,1])$.

3. Algoritmo de reducción de imágenes basado en la descomposición quadtree

El algoritmo original, en el que basamos nuestra propuesta, queda descrito en Algoritmo 1.

Algoritmo 1 Algoritmo base de reducción

ENTRADA: Imagen Q de $M \times N$ píxeles. Tamaño de reducción n.
SALIDA: Imagen reducida Q' de $\frac{M}{n} \times \frac{N}{n}$ píxeles.
Dividir la imagen Q en bloques disjuntos de $n \times n$ píxeles.
para cada bloque en Q **hacer**
Tomar $\alpha \in [0, 1]$
A partir de los píxeles del bloque $q_{1,1}, \ldots, q_{n,n}$ calcular el intervalo $[\underline{q}, \overline{q}]$ dado por
$$[\underline{q}, \overline{q}] = [\text{mín}(q_{1,1}, \ldots, q_{n,n}), \text{máx}(q_{1,1}, \ldots, q_{n,n})]$$
Calcular la intensidad del píxel de la imagen reducida: $K_\alpha([\underline{q}, \overline{q}]) = \underline{q} + \alpha(\overline{q} - \underline{q})$.
fin para

El Algoritmo 1 divide la imagen en bloques disjuntos de $n \times n$ píxeles. Para cada bloque, se construye un intervalo que representa la variación de intensidades de los píxeles pertenecientes al bloque. Finalmente, dado un valor de $\alpha \in [0, 1]$, se obtiene la intensidad del píxel de la imagen reducida aplicando el operador K_α al intervalo construido. Por tanto, cada bloque de $n \times n$ píxeles se transforma en un único píxel (proceso de reducción).

El paso clave del Algoritmo 1 es la elección del parámetro α utilizado para reducir cada bloque. Para evaluar cuál es la mejor reducción obtenida (mejor valor del parámetro α), se utiliza el siguiente procedimiento:

(i) reducir la imagen utilizando diferentes operadores K_α (cambiando el valor de α);
(2) reconstruir la imagen a su dimensión original mediante interpolación;
(3) medir las diferencias entre la imagen original y la reconstruida utilizando el Error Cuadrático Medio (MSE).

Recordamos que el MSE entre dos imágenes A, B de $M \times N$ píxeles se obtiene como

$$MSE(A, B) = \frac{L - 1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (a_{ij} - b_{ij})^2.$$

Cuanto menor sea la diferencia entre la imagen original y la reconstruida (menor MSE), mejor será el algoritmo de reducción.

Aunque en [3] se presentan algunos métodos para obtener el valor de α que se debe usar, los mejores resultados se obtienen tomando el mismo valor de α para toda la imagen, concretamente $\alpha = 0{,}5$. La idea principal de este trabajo es mejorar los resultados obtenidos utilizando valores específicos de α para cada bloque, teniendo en cuenta las propiedades de los píxeles en ese bloque. Sin

embargo, cuando el tamaño de la imagen es muy grande, encontrar el mejor valor de α para cada bloque puede ser una tarea computacionalmente muy costosa.

Para solucionar este problema, nuestra propuesta se basa en dos pasos: primero, aplicar una descomposición quadtree de la imagen, teniendo en cuenta la propiedad de homogeneidad. Esto nos permite dividir la imagen en cuadrantes de diferentes tamaños en los que las intensidades de los píxeles son homogéneas; y segundo, encontrar, para cada cuadrante, cuál es el mejor valor de α. De esta forma, cada bloque de $n \times n$ píxeles que esté dentro del cuadrante se procesará utilizando ese valor de α.

Debido a la descomposición quadtree, a partir de ahora vamos a considerar siempre imágenes de $M \times M$ píxeles donde $M = 2^k$, $k \in \mathcal{N}^+$ y un tamaño de reducción de $n = 2$.

3.1. Descomposición quadtree basada en homogeneidad

Dada una imagen Q de $M \times M$ píxeles, la descomposición quadtree [8] devuelve una estructura de árbol cuaternario en la que el nodo raíz representa la imagen completa. El proceso divide la imagen en cuatro cuadrantes del mismo tamaño si la homogeneidad del bloque es baja (o al menos menor que un umbral th). Esta regla se aplica recursivamente a cada cuadrante hasta que una de estas dos condiciones se cumpla: (i) el bloque es suficientemente homogéneo o (ii) se ha alcanzado la división máxima. En nuestro caso, fijamos la división máxima en el tamaño 2×2, por lo que un bloque de 2×2 píxeles nunca puede ser dividido. Tras este proceso, el resultado de la descomposición quadtree es una conjunto de t cuadrantes $Q^1, \ldots, Q^t$ de dimensiones $n'_1 \times n'_1, \ldots, n'_t \times n'_t$ (con $n'_1, \ldots, n'_t$ números naturales pares).

Sean $q^l_{1,1}, \ldots, q^l_{n'_i,n'_i}$ los píxeles de un cuadrante Q^l. La medida de homogeneidad de dicho cuadrante viene dada por $H(Q^l) = 1 - \text{máx}(q^l_{1,1}, \ldots, q^l_{n'_i,n'_i}) + \text{mín}(q^l_{1,1}, \ldots, q^l_{n'_i,n'_i})$ [6]. Obsérvese que si todos los píxeles tienen la misma intensidad, entonces $H(Q^l) = 1$ (homogeneidad máxima). Por el contrario, si existe al menos un píxel con intensidad 0 y otro píxel con intensidad 1, entonces $H(Q^l) = 0$ (mínima homogeneidad).

La estructura obtenida por la descomposición quadtree es muy útil para la reducción de imágenes. Por un lado, si un gran área de la imagen (cuadrante) es muy homogénea (las intensidades de todo el área son muy similares), podemos utilizar el mismo valor de α para cada uno de los subbloques 2×2 que están dentro de ese área. Por el contrario, si un área es muy heterogénea, es mejor dividirla en áreas más pequeñas y homogéneas, donde el mismo valor de α se pueda aplicar (ver Algoritmo 2).

3.2. Selección del mejor α para cada área homogénea

Una vez que la imagen se ha dividido en un conjunto de cuadrantes homogéneos, el siguiente paso consiste en encontrar el mejor valor de α para cada cuadrante. Para ello, estudiamos cada cuadrante individualmente, comenzando

Algoritmo 2 Descomposición quadtree

ENTRADA: Imagen Q. Umbral th
SALIDA: Lista de cuadrantes $Q^1, \ldots, Q^t$.
  si $H(Q) < th$ **entonces**
    Dividir Q en cuatro cuadrantes $Q^1, \ldots, Q^4$
    para cada Q^l desde Q^1 hasta Q^4 **hacer**
      Descomposición quadtree (Q^l, th)
    fin para
  en otro caso
    Añadir Q^l a la lista de cuadrantes
  fin si

con el aquel que tenga mayor homogeneidad. Reducimos cada cuadrante utilizando 11 posibles valores de α ($\alpha = 0, 0{,}1, \ldots, 1$) y después lo ampliamos utilizando la interpolación bilineal. Para decidir cuál es el mejor valor de α, comparamos cada uno de los 11 cuadrantes ampliados con respecto al mismo cuadrante de la imagen original. Tomamos como α aquel asociado al cuadrante ampliado menos disimilar. El esquema de este algoritmo se muestra en Algoritmo 3.

Algoritmo 3 Algoritmo para seleccionar el mejor α

ENTRADA: $Q^1, \ldots, Q^t$ cuadrantes obtenidos en la descomposición quadtree
SALIDA: $\alpha_1, \ldots, \alpha_k$ valores óptimos de α para cada cuadrante
  para cada cuadrante Q^l **hacer**
    para $\alpha = 0, 0{,}1, \ldots, 1$ **hacer**
      Reducir cada bloque 2×2 dentro de Q^l utilizando el operador K_α (Definición 1).
      Reconstruir el cuadrante reducido a un nuevo cuadrante $Q_\alpha^{l'}$ utilizando interpolación bilineal.
      Medir la diferencia entre Q^l y $Q_\alpha^{l'}$ utilizando $MSE(Q^l, Q_\alpha^{l'})$
    fin para
    Take $\alpha_l = \arg\min_\alpha MSE(Q^l, Q_\alpha^{l'})$
  fin para

4. Estudio experimental

En esta sección probamos nuestro algoritmo de reducción sobre 14 imágenes en escala de grises de 256×256 píxeles, obtenidas del conjunto de imágenes de test disponible en http://decsai.ugr.es/cvg/dbimagenes (ver Figura 1). La descomposición quadtree se basa en la función *qtdecomp* que ofrece Matlab.

En nuestra propuesta, la elección del umbral en la descomposición quadtree es un paso clave. Este umbral indica si un bloque debe dividirse en cuatro cuadrantes o no dependiendo de su homogeneidad (si es menor que el umbral). Si el

Figura 1. Imágenes utilizadas en el experimento.

umbral es pequeño, la imagen se divide en pocos cuadrantes, por lo que utilizamos el mismo valor de α en grandes regiones de la imagen. En el caso extremo de que el umbral valga 0, $th = 0$, la imagen no se divide y por tanto aplicamos un valor de α constante para toda la imagen ($\alpha = 0{,}5$) como en el algoritmo original de reducción en el que nos hemos basado. Por contra, cuando el valor del umbral aumenta, la imagen se divide en un mayor número de bloques, y por tanto un mayor número de valores α se pueden utilizar. Cuando el umbral es máximo, $th = 1$, aplicamos un valor específico de α a cada bloque de reducción (excepto en las áreas planas). Por tanto, diferentes valores del umbral th nos proporcionan reducciones muy diferentes. Para medir cómo de buenos son los resultados del algoritmo propuesto, reducimos cada imagen original de la Figura 1 a tamaño 128×128 píxeles, y después ampliamos la versión reducida mediante interpolación bilineal, para obtener una imagen del mismo tamaño que la original. Entonces, podemos comparar estas dos imágenes mediante el MSE.

A continuación, analizamos los resultados obtenidos en función de diferentes valores del umbral th. En la Figura 2 mostramos algunos de los resultados para la imagen original *Cameraman*. Cada fila corresponde a un umbral diferente ($th = 0$, $th = 0{,}25$, $th = 0{,}5$, $th = 0{,}75$ and $th = 1$). En la segunda columna mostramos los valores de α obtenidos para cada bloque (el negro representa $\alpha = 0$ y el blanco $\alpha = 1$). En la tercera columna mostramos la imagen reducida obtenida, y en la cuarta columna su correspondiente ampliación utilizando interpolación bilineal. La quinta columna muestra dos valores diferentes: el primero es el error cuadrático medio (MSE) entre la imagen ampliada y la original y el segundo muestra el tiempo de ejecución. Obsérvese que conforme aumenta el valor del umbral, el valor de α varía en bloques más pequeños. Este hecho implica que la calidad de los resultados es mejor, pero también aumenta el tiempo de ejecución. Todos los experimentos presentados en este trabajo se han ejecutado con un procesador Intel Core i5-4570 @ 3.20GHz con 8 GB de Ram.

En la Tabla 1 mostramos el MSE resultante para cada imagen (columnas) y para cada uno de los umbrales estudiados (filas). En este experimento tomamos valores para el umbral desde 0 hasta 1 con un incremento de 0,05.

Finalmente, en la Figura 3, mostramos: el valor medio de MSE para las 14 imágenes y el tiempo medio de ejecución para cada umbral. Observe que conforme el valor del umbral aumenta, también lo hacen la calidad de la reducción y el tiempo consumido. En este sentido, si necesitamos un reducción muy rápida, tenemos que llegar a un compromiso entre calidad y velocidad. Por el contrario, si sólo nos interesa la calidad, deberíamos utilizar un umbral cercano a 1.

Umbral	Valores α	Reducida	Ampliada	MSE/Tiempo(s)
0				274.4660/ 0.9609
0.25				248.3203/ 10.9688
0.5				234.6124/ 20.8802
0.75				228.1822/ 32.5806
1				224.3304/ 120.3823

Figura 2. Resultados obtenidos con el algoritmo propuesto. Cada fila se corresponde con un valor de umbral. Valores α obtenidos (columna 2), imagen reducida (columna 3), imagen ampliada (columna 4) y MSE y tiempo de ejecución (columna 5).

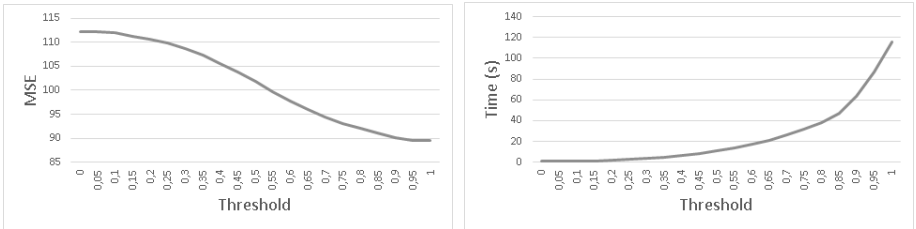

Figura 3. (Izquierda)MSE medio de las 14 imágenes y (derecha) tiempo de ejecución medio.

Th.	Im. 1	Im. 2	Im. 3	Im. 4	Im. 5	Im. 6	Im. 7	Im. 8	Im. 9	Im. 10	Im. 11	Im. 12	Im. 13	Im. 14
0	158.77	60.57	221.95	274.47	65.76	37.17	107.24	71.94	87.93	128.52	251.63	38.82	27.17	37.72
0.05	158.77	60.57	221.95	274.47	65.76	37.17	107.24	71.94	87.93	128.52	251.63	38.82	27.17	37.72
0.1	158.77	60.57	221.95	273.15	65.76	37.17	107.24	71.94	87.93	128.52	251.63	38.82	27.17	37.72
0.15	158.72	60.57	221.84	263.48	65.76	37.17	107.24	71.92	87.93	128.50	251.63	38.82	27.17	37.72
0.2	158.59	60.57	221.03	255.07	65.49	37.17	107.24	71.72	87.93	128.10	251.63	38.82	27.17	37.72
0.25	158.55	60.57	219.73	248.32	65.14	37.17	106.56	71.40	87.50	127.32	251.63	38.82	27.17	37.61
0.3	156.63	60.57	216.19	243.46	63.87	36.93	105.84	70.82	86.87	126.09	251.60	38.80	27.17	37.11
0.35	152.79	60.57	211.58	240.34	63.11	36.69	104.31	69.71	85.14	124.39	251.40	38.74	27.06	36.21
0.4	147.78	60.40	206.13	238.34	60.21	36.04	103.40	68.33	83.60	122.34	251.04	38.62	26.99	35.18
0.45	141.64	59.81	199.66	235.97	58.74	35.65	102.69	67.14	81.89	120.42	249.82	38.48	26.91	33.89
0.5	135.76	59.24	194.19	234.61	56.52	35.05	101.92	65.05	79.93	118.49	247.68	38.24	26.54	32.69
0.55	130.42	58.73	189.36	232.38	54.70	34.22	100.64	63.50	77.37	115.93	243.47	37.91	25.78	31.72
0.6	126.65	57.59	186.26	231.18	52.80	33.33	99.93	62.09	74.95	113.90	237.76	37.40	24.66	30.39
0.65	123.15	56.42	183.29	230.18	51.27	32.26	99.40	60.84	73.31	112.27	232.97	36.89	23.36	29.12
0.7	120.85	54.56	181.16	229.12	49.91	31.21	98.66	59.54	70.76	110.85	227.78	36.47	22.57	28.12
0.75	119.06	53.62	179.36	228.18	49.36	29.95	97.45	58.48	68.98	109.82	223.31	35.73	22.06	27.08
0.8	118.00	52.57	178.30	227.22	48.77	28.93	96.07	57.55	67.63	109.04	221.00	35.03	21.64	26.17
0.85	117.10	51.64	177.24	225.98	48.33	28.14	94.91	56.47	66.67	108.37	219.33	34.14	21.32	25.30
0.9	116.32	50.95	176.42	224.98	47.81	27.63	93.71	55.25	65.74	107.74	218.28	32.58	21.02	24.57
0.95	115.71	50.52	175.59	224.48	47.50	27.30	92.88	54.56	65.12	107.12	217.84	31.50	20.73	23.93
1	115.64	50.39	175.44	224.33	47.08	27.16	92.78	54.53	64.94	106.97	217.83	31.46	20.33	23.77

Cuadro 1. MSE de cada imagen $1-14$ (columnas) obtenido utilizando valos de umbral desde 0 hasta 1 con un incremento de 0,05 (filas).

5. Conclusiones

Basándonos en los resultados experimentales, probamos que la elección de un valor α específico para cada área de la imagen es mejor que seleccionar un valor constante para toda la imagen. Nuestra propuesta es capaz de obtener una solución intermedia entre calidad y tiempo consumido. Sin embargo, en el futuro queremos estudiar una manera automática de encontrar el mejor valor de α en lugar de probar diferentes valores. Además, queremos extender este algoritmo de reducción de imágenes para trabajar con imágenes en color, siguiendo las ideas presentadas en [5].

Agradecimientos

Este trabajo está parcialmente financiado por el proyecto TIN2013-40765-P.

Referencias

1. G. Beliakov, H. Bustince and D. Paternain, "Image Reduction Using Means on Discrete Product Lattices," IEEE T. Image Process. **21**, 1070–1084 (2012).
2. H. Bustince, T. Calvo, B. De Baets, J. Fodor, R. Mesiar, J. Montero, D. Paternain and A. Pradera, "A class of aggregation functions encompassing two-dimensional OWA operators," Inform. Sciences **180**, 1977–1989 (2010).
3. H. Bustince, D. Paternain, B. De Baets, T. Calvo, J. Fodor, R. Mesiar, J. Montero and A. Pradera, "Two Methods for Image Compression/Reconstruction Using OWA Operators," in Recent Developments in the OWA Operators, STUDFUZZ 265, 229–253 (2011).
4. F. Di Martino, P. Hurtik, I. Perflieiva and S. Sessa, "A color image reduction based on fuzzy transform," Inform. Sciences **266**, 101–111 (2014).
5. M. Galar, A. Jurio, C. Lopez-Molina, D. Paternain, J. Sanz and H. Bustince, "Aggregation functions to combine RGB color channels in stereo matching," Opt. Express **21** 1247–1257 (2013).

6. A. Jurio, H. Bustince, M. Pagola, P. Couto, W. Pedrycz, "New measures of homogeneity for image processing: an application to fingerprint segmentation," Soft Comput. **18** 1055-1066 (2014).
7. D. Paternain, J. Fernandez, H. Bustince, R. Mesiar and G. Beliakov, "Construction of image reduction operators using averaging aggregation functions," Fuzzy Set. Syst. In Press.
8. G.J. Sullivan and R.L. Baker, "Efficient Quadtree Coding of Images and Video," IEEE T. Image Process. **3**, 327–331 (1994).