

Compresión fractal de imágenes digitales

Lucas Damian Catalano

18 de Noviembre de 2011

1. Introducción

Debido a la inmensa cantidad de información digital que se maneja cotidianamente, es de vital importancia obtener métodos que permitan que dicha información sea representada de la manera más reducida posible ya sea por cuestiones de almacenamiento o para poder ser enviadas a mayor velocidad. Entre toda la información digital, las imágenes (estáticas y como video) son, por un lado, de las más usuales y también las más pesadas desde el punto de vista de la cantidad de información que contienen.

En este caso voy a comentar, brevemente, uno de los métodos más eficaces para reducir el peso de imágenes (en principio estáticas, aunque puede extenderse de diferentes maneras a videos y animaciones).

A pesar de ser un método con el que se obtienen muy buenos resultados en relación a su capacidad de reducción de peso sin pérdida relevante de calidad visual, el mismo no es comunmente usado en los sistemas informáticos, probablemente, debido a los largos tiempos de ejecución de sus algoritmos.

2. Compresión

La compresión de imágenes (o más en general de datos) puede clasificarse de varias maneras, una de ellas es la compresión con y sin pérdida de información. Esto quiere decir que se puede reducir el peso de cierta imagen asumiendo que no queremos perder la información exacta que nos brinda cada uno de sus píxeles (los píxeles son cada uno de los puntos de la imagen, un píxel tiene ubicación en la imagen y color -o todo de gris-), o bien podemos intentar reducir la información permitiendo que cierta información que consideramos irrelevante se pierda; en general, en el caso de las imágenes, se “permite” perder aquella información que visualmente, el ojo humano no registre o registre sutilmente de forma tal que la imagen no pierda su esencia. Por supuesto, los métodos de compresión con pérdida son mucho más eficientes desde el punto de vista de la reducción de información a guardar o enviar. Es por esto que en el caso de las imágenes y sonidos, en donde nos podemos dar la licencia de omitir información que consideremos imperceptible o irrelevante, se utilizan este tipo de compresión. Uno de esos métodos es el que se abordará en el presente apunte.

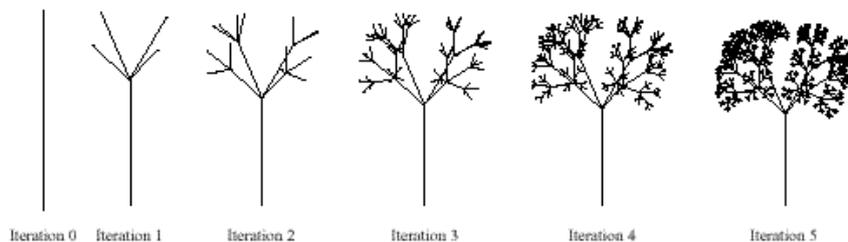


Figura 1: Sistema iterado de funciones cuyo atractor es la imagen de un arbol.

3. Representación de imágenes

Trabajaremos con imágenes en niveles de gris, ya que el trabajo con imágenes en colores se puede pensar como la superposición de tres imágenes en escala de grises, una representando a cada uno de los colores necesarios para codificar todos los colores visibles por el ojo humano (como rojo, verde y azul o cian, magenta y amarillo). Entonces, las imágenes digitales pueden representarse de diferentes maneras, la que utilizaré en este apunte es $\Delta = \{\tau : I^2 \rightarrow I\}$ donde I^2 es el cuadrado unidad con $I = [0; 1]$, la función τ da el nivel de gris en cada punto de la imagen. Para trabajar con imágenes de otros tamaños las ideas se pueden extender sin mayores inconvenientes.

También se necesitará una noción métrica para estos elementos $d(\tau_1, \tau_2)$ que será comentada en la sección 4 y 6.

4. Compresión fractal

Utilizando sistemas iterados de funciones (SIF), podemos obtener infinidad de conjuntos, que pueden ser interpretados como imágenes. Muchos de ellos son muy conocidos como el Triángulo de Sierpinski, o la Curva de Koch, pero fácilmente se pueden generar, de manera similar, otros conjuntos (o imágenes) como se muestra en la figura 1.

En este caso es muy claro que, si quisieramos almacenar esta imagen en, por ejemplo, una resolución de 1000×1000 monocromática (donde $\tau(i, j) = 0$ indica que el pixel en la posición (i, j) es negro y $\tau(i, j) = 1$ indica que es blanco), es mucho más conveniente hacerlo mediante el almacenamiento del SIF que guardar la información de cada pixel de su atractor.

La cuestión es entonces la siguiente, ¿Será posible, dada una imagen cualquiera, obtener un SIF tal que tenga a esa imagen como atractor? La respuesta es trivial si la imagen dada es autosemejante, pero claro, en la realidad, por ejemplo, en una fotografía de un paisaje no parece ocurrir que el mismo paisaje se viera repetido dentro de la imagen por doquier.

Es por esto que el concepto de SIF puede extenderse a algo más general para que nos dé la libertad de poder buscar semejanza entre algunas partes de la imagen en lugar de hacerlo con la imagen completa como se puede ver en la figura 2.



Figura 2: Zonas similares en la imagen de Lenna

4.1. Sistemas de funciones iteradas particionadas

Sean f_1, \dots, f_n un conjunto de transformaciones contractivas y sea $F = \cup f_i$.

Definición 1 Sea X un espacio métrico completo (Δ es un espacio métrico completo) y sea $A_i \subset X$ con $i = 1, \dots, n$. Un sistema de funciones iteradas particionadas SFIP es una colección de aplicaciones contractivas $f_i : A_i \rightarrow X$ con $i = 1, \dots, n$.

Dada una, esta puede ser subdividida por un lado en conjuntos D_1, \dots, D_n y por otro en R_1, \dots, R_n , a los primeros los llamaremos *domínios* y a los segundos *Rangos*. Notar que estos no son los dominios y rango de las funciones f_i mencionadas, sino $D_i \times I$ y $R_i \times I$.

Sean $v_1, \dots, v_n : I^3 \rightarrow I^3$ un conjunto de aplicaciones, definimos $f_i = v_i |_{D_i \times I}$. Las aplicaciones f_i conforman el SFIP. Cada función f_i funciona como $f_i(\tau) = f_i(x, y, \tau(x, y))$.

Diremos que F *enloza* a I^2 si ocurre que $I^2 = \cup_{i=1}^n R_i$. Esto será necesario para poder codificar imágenes.

Definición 2 Si $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ es una aplicación tal que $f(x, y, z_1) = (x', y', z'_1)$ y $f(x, y, z_2) = (x', y', z'_2)$, entonces diremos que f es contractiva z si existe un número real positivo $s < 1$ tal que

$$d(z'_1, z'_2) \leq s \cdot d(z_1, z_2)$$

donde d es la distancia euclidia usual y si, además, x' e y' son independientes de z_1 o de z_2 para todo x, y, z_1 y z_2 .

Proposición 1 Si f_1, \dots, f_n son contractivas z , entonces F es contractiva en Δ con la métrica del supremo ($d_{sup}(f(x), g(x)) = \sup_x \{|f(x) - g(x)|\}$).

Demostración $d_{sup}(W(f), W(g)) = \sup\{|W(f)(x, y) - W(g)(x, y)| : (x, y) \in I^2\}$

$$\begin{aligned}
&= \sup\{\text{comp } z \text{ de } |w_i(x, y, f(x, y)) - w_i(x, y, g(x, y))| : (x, y) \in D_i, i = 1, \dots, n\} \\
&\leq \sup\{s_i |f(x, y) - g(x, y)| : i = 1, \dots, n\} \\
&\leq \sup\{s |f(x, y) - g(x, y)|\} \\
&\leq \text{ssup}\{|f(x, y) - g(x, y)|\} \\
&\leq \text{sd}_{\text{sup}}(f, g)
\end{aligned}$$

A partir de lo anterior, es posible obtener una forma general para las aplicaciones f_i como se muestra a continuación:

$$f_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} u_i \\ v_i \\ o_i \end{pmatrix}$$

con $s_i < 1$, entonces $F = \cup f_i$ es contractiva bajo d_{sup} .

La transformación que controla la escala de grises es $m_i(z) = s_i \cdot z + o_i$ donde s_i es el contraste y o_i el brillo.

La parte geométrica de la transformación transforma el dominio D_i al tamaño y posición exactos de R_i por lo que la distancia entre los bloques en el sentido geométrico es cero. **Es la distancia entre los valores de gris la que hay que minimizar.**

4.2. Teorema del Collage

Luego de haber discutido las cuestiones que nos permitirán obtener un SFIP contractivo de alguna imagen τ es pertinente hacernos la pregunta ¿para cualquier imagen es posible hallar algún SIFP? La respuesta a esta pregunta es el Teorema del Collage.

Teorema 1 Sea X un espacio métrico completo y $f : X \rightarrow X$ un mapa contractivo s cuyo atractor (o punto fijo) es x_f , entonces,

$$d(x, x_f) \leq \frac{1}{1-s} d(x, f(x))$$

Luego, obtener un SIFP razonable será posible en la medida que podamos hallar f tal que $d_{\text{sup}}(x, f(x))$ sea mínima.

4.3. Observaciones sobre el ratio de compresión

En todo proceso de compresión es importante tener en cuenta cuál es el *ratio de compresión*, es decir, cuanto se ha reducido el peso de la imagen, pero también es importante saber cuanta información se ha perdido. A grandes rasgos, es importante señalar que:

- Mientras mayor sea el tamaño de los rangos, menos aplicaciones f_i serán necesarias, con lo que el ratio de compresión será mayor. Sin embargo, al tener rangos de mayor tamaño es menos probable hallar un dominio al que sea similar, con lo cual, seguramente la pérdida de información será también superior.
- La cantidad de afinidades que se permitan para comparar los D_i y los R_i también influirán en estas cuestiones. Al haber más posibilidad de transformaciones, más información acumulará el SIFP, con lo cual será menos

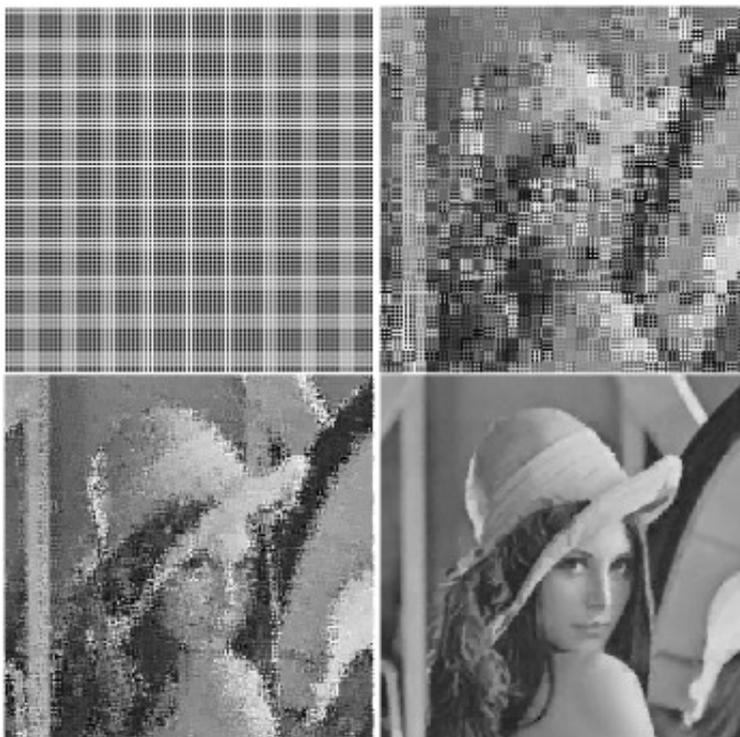


Figura 3: Teración de un SFIP cuyo atractor es la imagen de Lenna.

el ratio de compresión, pero por otro lado, habrá más posibilidades de hallar un dominio similar, con lo cual, la calidad será mejor.

5. Algunas variaciones del método

La compresión fractal, como todo proceso de compresión, tiene ventajas y desventajas. Quizás la desventaja más notoria sea los largos tiempos de ejecución en el proceso de compresión (obtención del SIFP). Algunas modificaciones que pueden hacerse para reducir un poco los tiempos son las siguientes:

- Reducción de búsqueda de dominios a vecinos.
- Pre-clasificación de dominios.

Por otro lado, si bien los resultados que se obtienen con este método han demostrado buenas tasas de compresión con relativa poca pérdida de calidad, hay algunas mejoras que pueden hacerse para obtener mejores resultados.

- Variación en el tamaño de los rangos y dominios (ver figura 4).
- Superposición de rangos y dominios.
- Ampliación de las posibles afinidades.

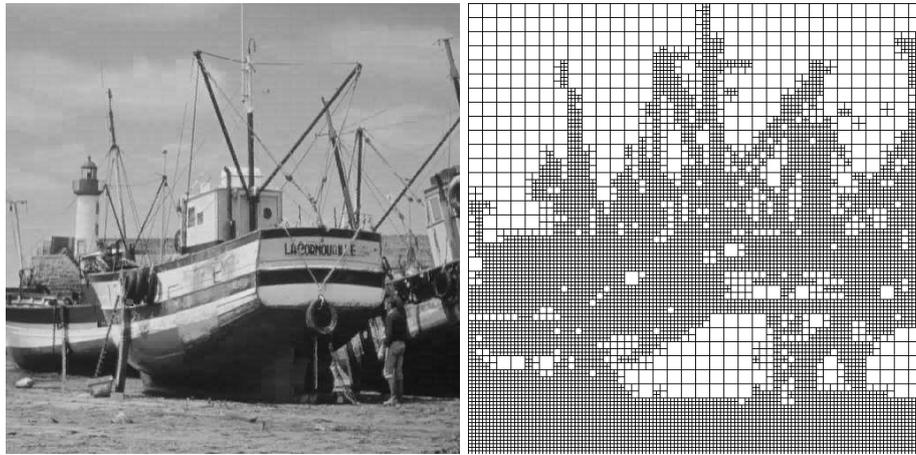


Figura 4: Imagen de un barco y división en rangos convenientes.

- Posibilidad de rangos y dominios no cuadrados.

Para más información, ver [1] y [2].

6. Problemas de las distancias entre imágenes

Este método, como tantas otras circunstancias en que es necesario comparar imágenes, ponen de manifiesto la importancia de trabajar con medidas que sean consistentes con el objetivo de la comparación. En nuestro caso, si queremos comparar imágenes y queremos ver si “son parecidas” nos estamos refiriendo a cuestiones visuales, es decir, esperaríamos que nuestra distancia de valores pequeños cuando las imágenes sean visualmente parecidas y valores altos en otros casos. Por desgracia, las medidas que computacionalmente son más fáciles de implementar y más rápidas de ejecutar (como por ejemplo la noción propuesta en este apunte $-d_{sup}$ -) tienen serias dificultades en este punto. Pensemos en dos imágenes, u y v tales que $u(x, y) = v(x, y)$ para todo x e y excepto para un $x = x'$ e $y = y'$ en el que $u(x', y') = 0$ y $v(x', y') = 1$; en este caso las imágenes serían visualmente muy parecidas, sin embargo, al calcular $d_{sup}(u, v)$, obtendremos el valor más grande posible. Con ejemplos similares podríamos mostrar las falencias de otras medidas. El problema, en este punto, es abierto.

Referencias

- [1] Yuval Fisher (ed.), Fractal Image Compression: Theory and Application, Springer Verlag, New York, 1995.
- [2] Pérez Ortiz, Juan Antonio. Codificación Fractal de Imágenes (1998). <http://www.dlsi.ua.es/japerez/pub/pdf/mastertesi1998.pdf>.