

# Funciones de Fitness en la Compresión Fractal en Bloques

Paula Millado<sup>1</sup>, María Laura Ivanissevich<sup>1</sup> y Claudio Delrieux<sup>2</sup>

<sup>1</sup>Universidad Nacional de la Patagonia Austral - Río Gallegos - ARGENTINA

<sup>2</sup> Universidad Nacional del Sur - Bahía Blanca - ARGENTINA

e-mail:mivanissevich@uarg.unpa.edu.ar - claudio@acm.org

## 1. Introducción

Los sistemas de funciones iteradas (IFS) constituyen una manera de representar imágenes por medio de conjuntos fractales (Barnsley, 1985). La factibilidad de representar imágenes con IFS se fundamenta en el Teorema del Collage, según el cual una imagen cualquiera puede ser arbitrariamente aproximada por un atractor fractal adecuadamente representado por un IFS (Barnsley, 1988a y 1988b). Esta propiedad hace tentadora la idea de buscar un método de compresión de imágenes que encuentre el collage adecuado para cualquier imagen de entrada. Esto constituye el problema inverso del IFS. Las estrategias hasta ahora propuestas para solucionar esta dificultad son dos. Por un lado, es posible restringir el espacio de búsqueda a un subconjunto de las transformaciones afines, como se propone en la compresión fractal en bloques (CFB) (ver Fisher 1995, Jacquin 1990). En la CFB se utiliza un conjunto relativamente grande de mapas entre segmentos de la imagen, utilizando escalas fijas, y rotaciones cuantizadas a cuartos de circunferencia. Por lo tanto, el espacio de búsqueda es relativamente pequeño para cada transformación. Los resultados obtenidos con la CFB poseen una buena relación de compromiso entre tiempo de cómputo, compresión resultante, y calidad final, aunque están evidentemente lejos de las posibilidades teóricas (Fisher 1997).

Por otro lado, es posible utilizar un mecanismo de búsqueda adaptativo, que tenga cierta inteligencia como para guiarse dentro del un subconjunto del espacio de búsqueda en el cual los resultados se acercan a la solución deseada. En estas técnicas se trabaja con una *población* de individuos, cada uno de los cuales representa un punto específico en un espacio de potenciales soluciones al problema dado. La población es capaz de evolucionar hacia una solución óptima, por medio de un proceso de supervivencia de los individuos mejor adaptados (respecto de una función de adaptación), y la aplicación de un proceso de generación aleatoria de nuevos individuos por medio de operadores de mutación o cruzamiento. Los esquemas mayoritariamente utilizados dentro de los algoritmos evolutivos son la programación evolutiva (PE) (Rechenberg 1973) y la programación genética (PG) (Michalewicz 1994).

El uso de PE y PG para la solución del problema inverso ha sido estudiado en trabajos anteriores (Millado 2003) donde se muestra que, para imágenes sencillas permite obtener mejores compresiones, tal vez con mayores tiempos de cómputo, que la CFB sin búsqueda. Sin embargo, el costo en tiempo de la compresión no es en general un obstáculo, dado que la misma se realiza una sola vez (desde el lado “servidor”), mientras que la descompresión es rápida y económica, y se realiza muchas veces (del lado “cliente”). Por dicha razón, este trabajo está orientado a investigar y testear estrategias evolutivas para la CFB, donde el énfasis está puesto en la calidad final de la imagen y en la tasa resultante de compresión.

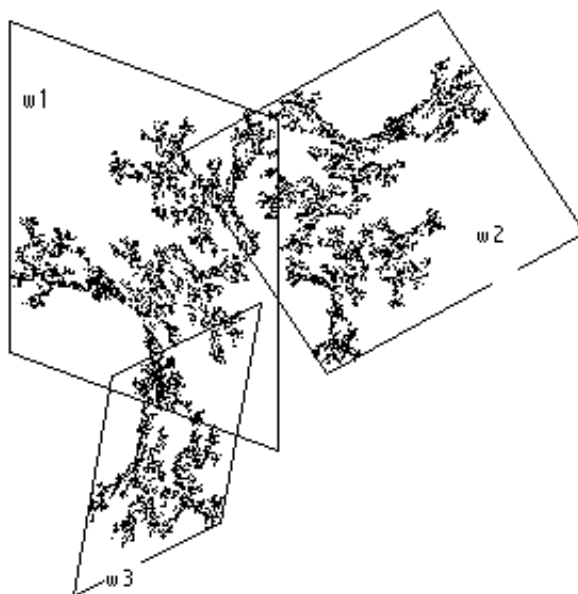


Figura 1: El atractor de un IFS con tres mapas (“coral”).

## 2. Sistemas de funciones iteradas y compresión fractal en bloques

Un sistema de funciones iteradas (IFS) consiste en una colección de transformaciones afines contractivas que mapea al plano  $\mathbb{R}^2$  sobre sí mismo. Esta colección de transformaciones define un mapa

$$W(\cdot) = \bigcup_{i=1}^n w_i(\cdot).$$

Las transformaciones afines contractivas tienen la importante característica de que cuando son aplicadas repetidamente, convergen a un punto fijo. El mapa  $W$  no es aplicado a todo el plano sino a un subconjunto, es decir, a una colección de puntos del plano. Dado un conjunto inicial  $S$ , podemos calcular  $w_i(S)$  para cada  $i$ , tomar la unión de estos conjuntos y obtener un nuevo conjunto  $W(S)$ .  $W$  es un mapa en el espacio de los subconjuntos del plano. Según el teorema de Hutchinson (Barnsley 1988b), cuando las transformaciones  $w_i$  son contractivas en el plano, entonces  $W$  es contractiva en el espacio (cerrado y acotado) de subconjuntos del plano. Por lo tanto, la iteración de un IFS converge a un único subconjunto del plano, el cual es un punto fijo del mapa  $W$ , denominado *atractor*  $A$  del IFS (ver Fig. 1).

Si por  $d(I, J)$  denotamos la distancia de Hausdorff entre subconjuntos  $I$  y  $J$  del plano en una métrica adecuada, y  $r$  es la contractividad global de  $W$ , entonces el teorema del collage permite afirmar la desigualdad

$$d(A, I) \leq \frac{1}{1-r} d(I, \bigcup w_i(I)), \quad (1)$$

lo cual significa que la distancia entre la figura  $I$  y el atractor  $A$  del IFS está acotada por la distancia entre  $I$  y el cubrimiento determinado por  $W$ . A mejores cubrimientos, mejor aproximación a la imagen final.

Las primeras aplicaciones del Teorema del Collage fueron realizadas manualmente, es decir, que se debía utilizar un proceso supervisado. Hasta ahora, los procesos automáticos

para resolver el problema inverso de la IFS con imágenes arbitrarias, no producen resultados satisfactorios. Arnaud Jacquin presenta una modificación al planteo original del problema, la cual puede automatizarse, en lo que se denomina “Sistemas de Funciones Iteradas Particionadas (PIFS)”, origen de la CFB. En los PIFS, las transformaciones no se mapean de la imagen entera a las partes, sino de partes más grandes a las más pequeñas. Un PIFS relaciona esas áreas de la imagen original que son similares en apariencia. Las áreas grandes son llamadas “Bloques Dominio” las pequeñas son llamadas “Bloques Rango”. El patrón de bloques rango es llamado la partición de una imagen. Una búsqueda de buenos bloques dominio para cada bloque rango es en realidad la clave de la CFB, dado que es lo que permitiría una mejor calidad final.

La importancia del factor búsqueda se puede dimensionar al calcular el tamaño total del pool de bloques dominio. Una imagen de 256x256 posee un total de 62001 bloques dominio de 8x8. Teniendo en cuenta las 8 simetrías isométricas se incrementan a 496008. Por otro lado hay 64009 bloques promedio de 4x4 lo que implica un total de 31748976072 pruebas de apareamiento. De aquí que mejorar la velocidad es el principal reto en la CFB.

### 3. Programación genética y evolutiva

La Programación Genética (PG) y la Programación evolutiva (PE) son técnicas bien conocidas que surgen de imitar lo que conocemos de la evolución natural. Existen individuos (fenotipos) que expresan una información genética (genotipo), y además están sujetos a la presión evolutiva del medio (*fitness*). Los individuos más exitosos (con mejor *fitness*) ven aumentadas sus posibilidades reproductivas, generando nuevos individuos con su mismo genotipo. Sin embargo, en el proceso reproductivo ocurre el fenómeno distintivo de la evolución natural: la aparición de mutaciones al azar. Este mismo esquema dinámico, con un sinnúmero de variaciones, es especialmente adecuado para resolver problemas que con los métodos de Inteligencia Artificial de alto nivel tradicionales (búsqueda heurística, lógica, etc.) son o bien intratables o bien imprácticos. Supongamos la existencia de una población con  $\mu$  ancestros, cuyo genotipo  $\bar{g}_i$  se encuentra codificado con números. Entonces se crea un conjunto  $\lambda$  de descendientes siguiendo el esquema

$$\bar{p}_i = (\bar{g}_i, \bar{\sigma}_i) \xrightarrow{\text{mut}} (\bar{g}_i + n_0(\bar{\sigma}_i), \alpha(\bar{\sigma}_i)),$$

donde  $n_0(\bar{\sigma}_i)$  denota la elección de un número de una muestra aleatoria normal con media cero y variancia  $\sigma_i$ .  $\alpha$  define una función de adaptación, la cual puede ser la distancia de Hamming o la distancia de Hausdorff entre el atractor  $A$  del IFS resultante y la imagen  $I$  de entrada. En nuestro caso,  $\bar{g}_i$  representa los parámetros de la IFS  $W$  que estamos tratando de encontrar.

Como estrategia híbrida se puede introducir, además de la mutación, el *crossover* o cruzamiento entre el genotipo de dos ancestros para obtener el genotipo del sucesor. Para evitar los mínimos locales, y quedar atrapado en un “nicho ecológico”, la estrategia mantiene una diversidad genotípica seleccionando un grupo numeroso de individuos con la mejor adaptación. Entre estas técnicas de *niching* podemos contar el *fitness sharing* (escalar el *fitness* en un entorno de cada individuo), el *crowding* (reemplazar al padre más cercano del individuo recientemente creado), y el *clustering* o agrupamiento, la cual resulta ser la más eficiente en la práctica.

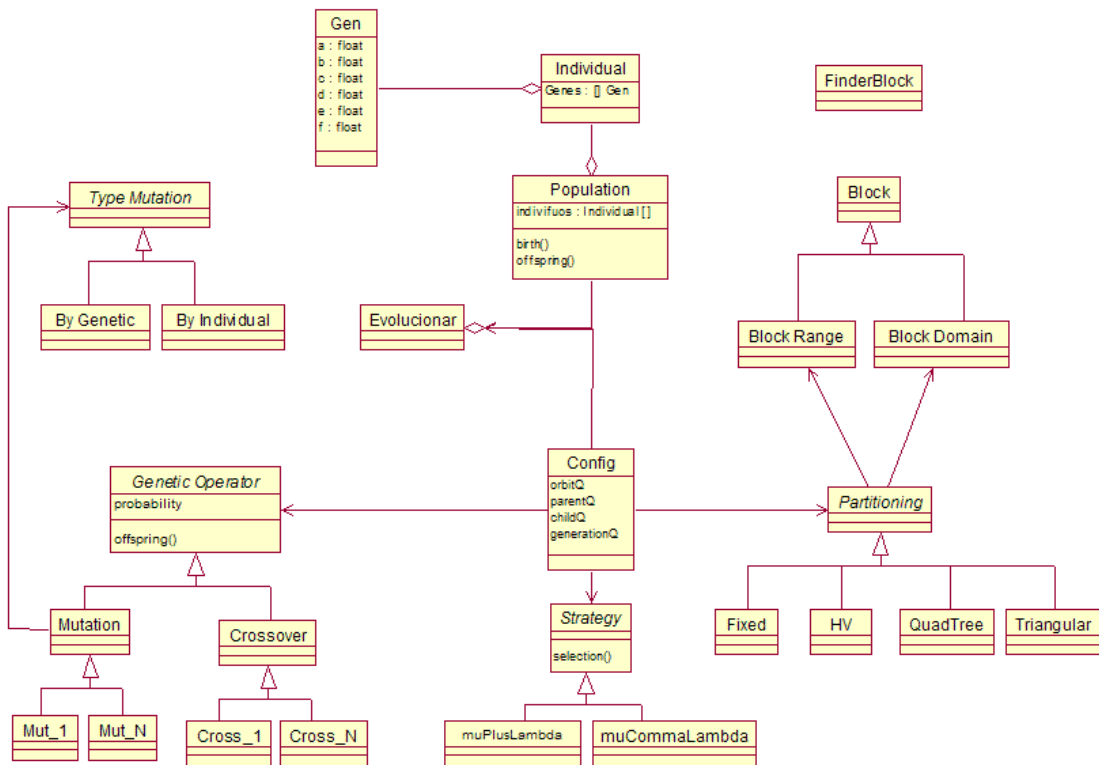


Figura 2: Diagrama UML del prototipo.

#### 4. Implementación de un sistema de CFB

Encontrar PIFS de buen comportamiento involucra 5 problemas básicos:

1. Particionar la imagen en bloques promedio.
2. Formar el set de bloques dominio.
3. Escoger el tipo de transformaciones a considerar.
4. Seleccionar una métrica de distancia entre los bloques.
5. Especificar un método para aparear los bloques rango con los bloques dominio.
6. Evaluar la calidad del PIFS resultante.
7. Si el resultado es adecuado, parar. En caso contrario, mutar y recomenzar.

Todos estos aspectos, incluyendo las estrategias evolutivas para generar nuevas poblaciones y las métricas para la función de fitness, están siendo consideradas en la implementación de un prototipo para poder investigar y cuantificar los efectos de las distintas decisiones. El diagrama UML del sistema puede verse en la Fig. 2

## 5. Conclusiones

En este trabajo presentamos un conjunto de estrategias evolutivas para resolver el problema inverso de las PIFS y proveer una buena alternativa para la CFB. A partir de este trabajo, la investigación está centrada en implementar y testear las estrategias evolutivas que permitan una CFB con menor error RMS y mayor tasa de compresión. También se considerarán dos métricas diferentes, las distancias de Hamming y de Hausdorff, como funciones de fitness en el proceso selectivo.

## Referencias

- [1] M. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. *Proc. Roy. Soc. London*, A399:243–275, 1985.
- [2] M. Barnsley, A. Jacquin, and F. Malassenet. Harnessing Chaos for Image Synthesis. *ACM Computer Graphics*, 22(3):131–14, 1988.
- [3] M. F. Barnsley. *Fractals Everywhere*. Academic Press, Boston, 1988.
- [4] A. Evans and M. Turner. Specialisation of Evolutionary Algorithms and Data Structures for the IFS Inverse Problem. In *Proceedings of the Second IMA Conference on Image Processing: Mathematical Methods, Algorithms and Applications*, pages 70–81, 1998.
- [5] Yuval Fisher, editor. *Fractal Image Compression: Theory and Application*. Springer-Verlag, New York, NY, USA, 1995.
- [6] Yuval Fisher, editor. *Fractal Image Encoding and Analysis*, volume 159 of *NATO ASI Series. Series F: Computer and Systems Sciences*. Springer-Verlag, New York, NY, USA, 1997.
- [7] Arnaud E. Jacquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. In Murat Kunt, editor, *Visual Communications and Image Processing '90*, volume 1360 of *SPIE Proceedings*, pages 227–239, Lausanne, Switzerland, October 1990.
- [8] R. Klette and P. Zamperoni. Measures of Correspondence Between Binary Patterns. *Image and Vision Computing*, 5:287–295, 1987.
- [9] E. Lutton. Mixed IFS - Resolution of the Inverse Problem using Genetic Programming. Technical Report Rapport 2631, INRIA, 1995.
- [10] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, second edition, 1994.
- [11] Paula Millado, Luis Sierpe, Laura Ivanissevich, and Claudio Delrieux. Testing Genetic and Evolutionary Strategies in the Inverse IFS Problem. In *Proceedings of the CISST 2002 Conference*, pages 267–274, CSREA Press, ISBN 1-892512-94-7, 2002.
- [12] Paula Millado, Laura Ivanissevich y Claudio Delrieux, Funciones de Fitness en el Problema Inverso de los IFS. RPIC, Reunión de Trabajo en Procesamiento de la Información y Control, San Nicolás, Octubre de 2003.
- [13] Shonkwiler. R. An Algorithm for Computing the Hausdorff Distance Efficiently. *Information Processing Letters*, 30:87–89, 1989.
- [14] R. Shonkwiler, F. Mendivil, and A. Deliu. Genetic Algorithms for the 1-D Fractal Inverse Problem. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 495–501, Los Altos, CA, 1995. Morgan Kaufmann Publishers.