

# Compressão Fractal de Imagens Baseada em Tabelas de Dispersão

Rodolfo Barriviera

Departamento de Informática - Universidade Federal do Paraná  
Curitiba-PR, Brasil, 81531-990

William Robson Schwartz

Department of Computer Science - University of Maryland Institute for Advanced Computer Studies  
College Park, MD, USA, 20742-327

Hélio Pedrini

Instituto de Computação - Universidade Estadual de Campinas  
Campinas, SP, Brasil, 13084-971

## Resumo

*Os sistemas computacionais, que proporcionam meios de armazenamento transmissão de dados por conexões remotas, têm evoluído significativamente nos últimos anos. Com isso, a demanda por métodos eficientes, especificamente para compressão de dados, tem sido alta. Este artigo propõe a utilização de uma tabela de dispersão como forma de aperfeiçoar algoritmos de compressão fractal baseados em busca exaustiva e por quadtree. Para avaliar a eficiência da técnica, várias imagens foram comprimidas utilizando-se diferentes medidas de similaridade e tamanhos de blocos. Uma análise estatística detalhada é realizada levando-se em conta o tempo de compressão, a taxa de compressão e uma medida de qualidade.*

## 1. Introdução

A evolução dos sistemas de armazenamento de dados, a velocidade de transmissão e o volume de informações transmitidas pelos possíveis meios de comunicação têm sido crescente nos últimos anos. Com isso, a demanda por métodos mais eficientes, especialmente para compressão de dados, tem sido alta.

As técnicas de compactação de arquivos, entre elas a compressão de imagens, contribuem para a eficiência dos sistemas computacionais utilizados em diversas áreas para o armazenamento e a transmissão de imagens, tais como medicina, sensoriamento remoto, automação industrial, educação a distância, entre muitas outras.

Técnicas de compressão de imagens são normalmente classificadas como sem perdas ou com perdas [9]. A compressão de imagens com perdas é um método irreversível,

garantindo, em geral, maiores taxas de compressão. Por outro lado, as técnicas de compressão de imagens sem perdas mantêm a qualidade da imagem original com taxas menores de compressão.

Métodos de compressão de imagens cada vez mais eficientes têm sido propostos por pesquisadores, de modo a resultar em altas taxas de compressão, redução de esforço computacional e reconstrução de imagens de boa qualidade. Uma técnica que visa atender a esses requisitos é a compressão fractal de imagens [7], que explora a similaridade em diferentes escalas da imagem para obter a compressão.

O principal objetivo deste trabalho é aprimorar os métodos de compressão fractal de imagens baseados em busca exaustiva e particionamento por *quadtree* a partir do uso da estrutura de dados denominada tabela de dispersão ou espalhamento (*hash table*), de modo a melhorar o desempenho dos algoritmos com relação ao tempo de compressão. Uma análise comparativa é realizada para avaliar os métodos desenvolvidos aplicados sobre um grande volume de imagens de diferentes naturezas, tais como fotografias, dados de terreno, cenas naturais e imagens médicas.

O texto está estruturado como segue. A seção 2 apresenta os principais conceitos relacionados à compressão fractal de imagens. A seção 3 descreve dois codificadores fractais clássicos. A seção 4 descreve a metodologia proposta neste trabalho. Os resultados obtidos a partir da metodologia são apresentados na seção 5. As conclusões do trabalho e propostas para trabalhos futuros são apresentadas na seção 6.

## 2. Compressão Fractal de Imagens

Três características principais são buscadas nos métodos de compressão de imagens: alta taxa de compressão, redução do esforço computacional e alta qualidade das imagens

codificadas. A compressão fractal procura atender satisfatoriamente a esses aspectos.

Os fractais são estruturas que apresentam irregularidades e fragmentação em uma faixa de escala muito ampla. Em síntese, fractal é uma forma que contém partes que são réplicas do todo sob algum aspecto [13].

Alguns conceitos importantes sobre compressão fractal de imagens são discutidos nas próximas seções.

## 2.1. Auto-similaridade

A auto-similaridade é uma propriedade das figuras geométricas tradicionais e também dos fractais, a qual consiste em se ter réplicas do fractal em diferentes escalas. Desse modo, entende-se que uma imagem pode ter porções similares que, de certa forma, contribuam para o processo de compressão e descompressão fractal.

## 2.2. Teorema do Ponto Fixo

Algumas funções utilizadas nos modelos fractais são denominadas contrativas. Essas funções possuem um valor denominado atrator da função, que é chamado de ponto fixo [12]. O teorema do ponto fixo diz que: *toda transformação contrativa, ao ser aplicada várias vezes com qualquer valor inicial, converge para um ponto fixo* [14].

Seja  $X$  uma métrica no espaço e  $W : X \rightarrow X$  contrativa, então  $W$  tem um ponto fixo  $|W|$ . A função é contrativa, quando se aplica o resultado dela a ela mesma, o que faz tender a um determinado valor, que é o atrator da função.

Uma contração pode ser representada da seguinte forma

$$f(x) = sx + o \quad (1)$$

em que  $s$  é o fator de contração com  $0 < s < 1$  e  $o$  é uma constante de deslocamento. Observa-se que  $x$  pode tender a qualquer valor inicial na função, pois irá tender ao ponto fixo da função contrativa.

## 2.3. Teorema da Colagem

O teorema da colagem utiliza como base a auto-similaridade (seção 2.1) dos fractais, ou seja, um fractal pode ser representado por transformações sobre o todo. Essas transformações são funções contrativas e aplicadas infinitas vezes sobre o conjunto que representa o fractal.

O teorema da colagem diz que: *se a transformação  $f : X \rightarrow X$  é um mapeamento contrativo em determinado espaço métrico completo  $(X, d)$  com um ponto fixo  $x_f$  e fator de contração  $s$ , então* [11]

$$d(x, x_f) \leq \frac{1}{1-s} d(x, f(x)) \quad \forall x \in X \quad (2)$$

Nota-se que a distância entre um ponto qualquer do espaço e seu atrator é menor que da sua transformação. Desse modo, quanto menor a distância  $d(x, f(x))$ , mais próximo se está do atrator. Assim, ao encontrar a função  $f(x)$  que faz  $d(x, f(x))$  tender a zero, descobre-se a função associada ao atrator  $x_f$ . As transformações buscam encontrar a menor distância entre os pontos para que se possa atingir o atrator da função e realizar de modo mais eficiente a transformação do fractal [3].

## 2.4. Transformação Afim

Uma transformação afim é importante nos fractais devido à possibilidade de se fazer combinações de rotação, escala, distorção e translação no espaço  $n$  dimensional [12]. A transformação afim é uma aplicação de  $T$  de  $R^n \in R^n$ , sendo que no plano  $R^2$  é representada da seguinte maneira

$$T(x, y) = (ax + by + e, cx + dy + f) \quad (3)$$

Os parâmetros lineares  $a, b, c, d$  determinam a rotação, escala e distorção. Os parâmetros que determinam a translação são  $e$  e  $f$ . A utilização de transformações afins é importante para a compressão fractal de imagens devido à possibilidade de se manipular parâmetros lineares que determinam formas de comportamento dos fractais.

## 2.5. Sistemas de Funções Iteradas

Segundo Ribas [10], um conjunto de transformações afins contrativas em um espaço métrico completo denomina-se Sistemas de Funções Iteradas (SFI). O SFI é um processo de iterações sobre um ponto (seção 2.2) que gera um fractal através de transformações afins contrativas (seção 2.4), tendendo sempre a um ponto fixo.

Considere  $A \subset \mathbb{R}^2$  um conjunto inicial e  $N$  contrações  $\omega_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  com contração  $0 < \alpha < 1 (i = 1, \dots, N)$ . Assim, define-se  $W : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , sendo

$$W(A) = \bigcup_{i=1}^N \omega_i(A) \quad (4)$$

em que  $\omega_i(A) = \{\omega_i(x) | x \in A\}$  é a imagem da transformação  $\omega_i$  ao ser aplicada aos vetores de  $A$ .  $W$  pode ser uma contração, ao se obter o fator  $\alpha \equiv \max\{\alpha_i, i = 1, \dots, N\}$ .

Um fractal gerado por SFI está em um conjunto  $X \subset \mathbb{R}^2$ , definido por

$$X = \lim_{n \rightarrow +\infty} W^{(n)}(A) \quad (5)$$

De acordo com a equação 5, para gerar um fractal é necessário aplicar, de forma iterativa, o conjunto das contrações  $\omega_i$  a um conjunto qualquer  $A$ . Se o fractal é formado por cópias individuais reduzidas da sua própria imagem, então a contração de  $W : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  é possível nesse espaço

para todas as imagens. Logo, o SFI é um conjunto de iterações que necessita das contrações que tendem para um único ponto (seção 2.2).

## 2.6. Sistemas de Funções Iteradas Particionadas

Os Sistemas de Funções Iteradas Particionadas (SFIP) são uma extensão do SFI, porém, com a vantagem de aplicar as iterações em partições menores da imagem [6]. Essas partições são selecionadas de acordo com as características de similaridade (seção 2.1).

Segundo Fisher [4], o SFIP utiliza as transformações afins  $\omega_i$  na forma matricial, como apresentado na equação 6.

$$\omega_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (6)$$

em que  $s_i$  define o contraste e  $o_i$  o brilho na transformação das imagens.

A imagem é particionada em regiões denominadas blocos moldes, sendo que cada um desses blocos é associado a um bloco domínio por meio de transformações afins, segundo uma determinada medida de similaridade. O processo de codificação envolve a busca do bloco de domínio que melhor se aproxima das características do bloco molde, tal que para cada bloco molde existirá um conjunto de parâmetros de transformações para representar uma região específica da imagem [10].

## 3. Codificadores Fractais

Um codificador fractal tem o objetivo de realizar a compressão da imagem recebida na entrada e gerar uma nova imagem de boa qualidade, quando reconstruída.

As transformações executadas no codificador fractal diferem de outros compressores de imagens, pois este utiliza características como auto-similaridade e ponto contrator da imagem para sua compressão.

A abordagem utilizada na codificação fractal da imagem é baseada em modelos locais (domínios e sub-domínios), que busca regiões similares na imagem para permitir a compressão. Dessa forma, há também um ganho computacional relevante durante as transformações da compressão [3].

Os algoritmos de compressão por fractal buscam a característica da auto-similaridade entre as partes maiores (blocos de domínio) e menores (blocos moldes) da imagem. Para isso, as imagens devem ser uma colagem das partes auto-similares que podem ser mapeadas entre elas [12].

Duas abordagens clássicas para a busca dos melhores pares entre blocos de domínio e blocos moldes são conhecidas como codificação baseada em busca exaustiva e codificação baseada em particionamento *quadtree*. Um trabalho recente

que apresenta várias abordagens para compressão fractal de imagens é descrito por Koli e Ali [7].

### 3.1. Codificação Baseada em Busca Exaustiva

A codificação fractal baseada na busca exaustiva para associação dos blocos moldes e blocos domínios foi originalmente proposta por Barnsley e Hurd [1]. Os blocos moldes da imagem são de dimensão  $n \times n$  pixels e os blocos de domínio têm dimensão  $2n \times 2n$ .

Este codificador é composto por três etapas principais: segmentação da imagem, obtenção dos blocos de domínio e geração dos códigos fractais, descritas a seguir:

1. *segmentação da imagem*: a imagem  $N \times N$  original é segmentada em blocos quadrados (blocos moldes) de tamanho fixo  $n \times n$ . Assim, uma imagem com dimensão  $512 \times 512$  pixels e blocos de  $4 \times 4$  pixels seria composta por uma matriz de 128 linhas e 128 colunas, ou seja, 16384 blocos moldes. Cada bloco molde irá gerar uma função, resultando um total de 16384 funções [12]. Com a divisão da imagem original em blocos moldes, permite-se um melhor desempenho computacional e facilidade de aplicação das transformações.
2. *obtenção dos blocos de domínio*: esta etapa tem o objetivo de fazer referências às transformações que serão executadas de acordo com as comparações de auto-similaridade e colagem entre os pixels dos blocos molde e de domínio, através da definição dos blocos de domínio. A obtenção dos blocos de domínio consiste em encontrar blocos que possuem o dobro de tamanho dos blocos molde, ou seja, são  $2n \times 2n$ . Diferentemente da etapa de segmentação, a definição dos blocos de domínio pode usar sobreposição [15], conforme ilustrado na figura 1.

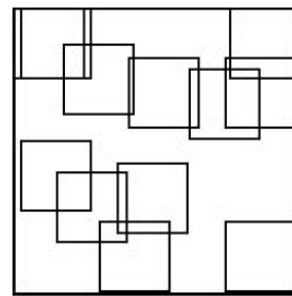


Figura 1. Blocos de domínio.

3. *geração dos códigos fractais*: com os blocos moldes e domínios já definidos, inicia-se o processo de comparações e transformações para gerar a nova imagem

comprimida. Primeiramente, os blocos de domínio são reduzidos ao tamanho dos blocos moldes, cuja mudança de escala é realizada por meio de uma transformação afim. Após isso, faz-se a busca do bloco molde que melhor se aproxima do bloco de domínio, segundo a métrica de similaridade adotada. Para cada bloco molde da imagem, ajusta-se o fator de deslocamento de intensidade ( $o$ ), que se obtém pela diferença entre as médias de intensidade dos blocos de domínio reduzidos e dos blocos moldes. Então, calcula-se a distância entre esse bloco molde e os blocos de domínio reduzidos segundo as seguintes simetrias dadas pelas transformações geométricas: identidade, rotação de  $90^0$ , rotação de  $180^0$ , rotação de  $270^0$ , reflexão horizontal, reflexão horizontal da imagem com rotação de  $90^0$ , reflexão vertical ou reflexão na diagonal. Nesse processo são selecionadas e armazenadas a posição  $(Dx, Dy)$ , a simetria  $(m_i, i \in \{0, 1, 2, \dots, 7\})$  e o fator de deslocamento de intensidade ( $o_i$ ) referente ao bloco de domínio que possui a menor dessas distâncias. Para cada bloco molde referenciado, armazenam-se as tuplas  $(Dx, Dy, m_i, o_i)$ . Esse conjunto de tuplas forma o fractal da imagem de saída, ou seja, o registro que será gravado no arquivo codificado da imagem [12]. Esses passos são executados para todos os blocos moldes.

Este procedimento de codificação é conhecido como busca exaustiva devido à comparação sequencial para encontrar o melhor par entre cada bloco de domínio e bloco molde. A descompressão (decodificação) da imagem tem uma sequência simples de iterações, que inicia com uma imagem  $f_0$  qualquer, e aplica-se iterativamente as transformações utilizando as tuplas definidas  $(Dx, Dy, m_i, o_i)$  até se obter uma convergência desejada, normalmente após um número baixo de iterações (menor que 10).

### 3.2. Codificação Baseada em Quadtree

Uma técnica para reduzir o custo computacional do codificador baseado em busca exaustiva utiliza o particionamento *quadtree*, proposto por Fisher [5], o qual permite a divisão da imagem em blocos de tamanhos diferentes. Dessa forma, diferentes níveis de detalhes presentes na imagem podem ser explorados, tal que regiões homogêneas são cobertas por blocos maiores e regiões heterogêneas são cobertas com blocos menores. O processo de subdivisão é realizado até que um tamanho mínimo de bloco seja obtido.

No particionamento *quadtree*, esses blocos são comparados durante as iterações a fim de encontrar semelhança para as transformações. Caso a similaridade esteja fora de uma tolerância, o bloco é subdividido em quatro novos blocos para continuar a busca. Esse processo é realizado até o término da codificação.

Nesse processo de codificação é comum se definir um método de classificação das regiões mais próximas aquele bloco em uso. Assim, é possível diminuir o custo computacional das iterações. Neste caso, pode-se computar valores proporcionais para cada quadrante segundo a média e variância dos valores de intensidade para classificação.

A decodificação consiste em executar iterações  $W$  para qualquer imagem inicial. O particionamento *quadtree* determina as regiões da imagem a serem divididas. Cada uma dessas regiões é representada por  $R_i$ , seu domínio por  $D_i$  mapeado para blocos com tamanho mínimo de  $2 \times 2$  pixels, sem sobreposição. Os pixels referentes ao domínio são multiplicados por  $s_i$ , adicionado ao  $o_i$  e colocado na região conforme sua informação e orientação. Assim, finaliza-se uma iteração da decodificação. Essas iterações são realizadas até se encontrar o ponto fixo da função (seção 2.2) por meio do contrator da função, pois, quando se chega ao ponto fixo da função, a imagem não é mais alterada devido ao seu contrator e tem-se a imagem final.

## 4. Metodologia Proposta

A técnica proposta para acelerar o processo de comparação entre blocos de domínio e blocos moldes utiliza a estrutura de dados denominada tabela de dispersão ou de espalhamento (*hash table*) [2, 8].

A distribuição dos elementos na tabela é realizada por meio de uma função de dispersão (*hashing*). A função utiliza certa informação do conjunto de dados para calcular a posição da tabela em que o dado será alocado. O resultado dessa função pode ser único (função de dispersão perfeita, em que cada elemento do conjunto é distribuído em uma posição específica única da tabela) ou então pode ter colisões (em que uma ou mais posições apresentam mais de um elemento do conjunto).

A função de dispersão definida no algoritmo para compressão fractal de imagens baseada em *hash table* é definida como

$$p = \sum f_{mod} \left( \left\lfloor \frac{\chi[a, b] - \bar{\chi}}{\lambda \sigma} + B \right\rfloor, C \right) \quad (7)$$

em que  $p$  é a posição resultante da função de espalhamento,  $f_{mod}$  é a função que retorna o resto da divisão entre as partes, em que  $\chi[a, b] - \bar{\chi}$  é o valor do pixel do bloco menos a média do bloco,  $B$  é uma constante com valor 75000,  $\lambda$  tem valor 0,4,  $\sigma$  é o valor de desvio padrão do bloco e a constante  $C$  tem o valor  $\frac{\sqrt{5}-1}{2}$ . Os valores para as constantes foram empiricamente atribuídas com base em dados experimentais.

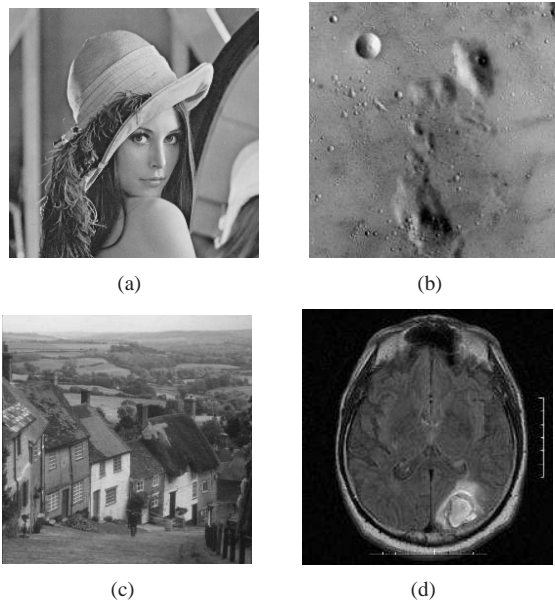
O processo de compressão da imagem com essa estrutura ocorre em duas etapas: distribuição dos blocos de domínio na tabela de dispersão e associação dos blocos de domínio aos blocos moldes por meio de pesquisa na tabela.

Para distribuir os blocos de domínio na tabela, utiliza-se o valor de desvio padrão de cada bloco de domínio e os pixels do bloco. As informações do bloco são enviadas para a função *hashing* que gera um valor indicando a posição na tabela. O resultado é a distribuição dos blocos, tal que valores mapeados para a mesma posição são encadeados por meio de uma lista ligada.

Após a distribuição dos blocos de domínio, as mesmas propriedades dos blocos moldes são utilizadas para gerar cada posição na tabela. Com a posição calculada, realiza-se a pesquisa pelos blocos de domínio com valores dentro de uma certa tolerância, visto que dificilmente o bloco molde possuirá exatamente o mesmo valor de chave do bloco de domínio. Assim, todos os blocos moldes ao serem comparados com os blocos de domínio são associados aqueles com maior semelhança, finalizando o processo de compressão.

## 5. Resultados Obtidos

Para avaliar os resultados obtidos a partir da metodologia proposta e compará-los com os resultados gerados pelos codificadores fractais descritos na seção 3, um conjunto de diferentes imagens foi selecionado contendo fotografias, dados de terreno, cenas naturais e imagens médicas. As imagens possuem dimensões de  $512 \times 512$ ,  $256 \times 256$  e  $128 \times 128$ . As imagens utilizadas são ilustradas na figura 2.



**Figura 2. Imagens utilizadas nos testes de compressão fractal de imagens. (a) fotografia; (b) terreno; (c) cena natural; (d) imagem médica.**

O computador utilizado nos testes foi um Intel Xeon 3 GHz com 2GB de memória RAM. Os algoritmos, implementados no pacote Matlab, para compressão de imagens fractal são baseados na busca exaustiva, no particionamento *quadtree* e na tabela de dispersão (*hash table*).

Em cada algoritmo, as medidas de similaridade aplicadas entre blocos são a média da intensidade, o desvio padrão, a entropia e essas três medidas combinadas. De acordo com os experimentos realizados, a média se mostrou como a melhor medida de similaridade entre blocos, sendo a única medida utilizada nos experimentos mostrados neste trabalho. Diferentes tamanhos de blocos também foram testados, sendo que, devido à limitação de espaço no texto, apenas os resultados obtidos para blocos com dimensões  $8 \times 8$  pixels são apresentados para a busca exaustiva. No caso do particionamento *quadtree*, os blocos podem ter múltiplos tamanhos, uma vez que é permitido a divisão da imagem conforme um limiar pré-estabelecido.

A métrica utilizada para medir a qualidade das imagens após a compressão é a relação sinal-ruído de pico (PSNR, *peak signal-to-noise ratio*). Quanto maior o valor resultante, melhor é a qualidade da imagem. Essa métrica retorna um valor em decibéis (dB), de acordo com a equação 8.

$$\text{PSNR} = 20 \log_{10} \left\{ \frac{2^x - 1}{\text{RMSE}} \right\} \quad (8)$$

em que  $(2^x - 1)$  é o maior valor que o pixel pode assumir na imagem. A raiz do erro quadrático médio (RMSE, *root mean square error*), que também permite a comparação de duas imagens  $A$  e  $B$  com as mesmas dimensões, é dada pela equação 9.

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [A(x, y) - B(x, y)]^2} \quad (9)$$

em que  $M$  e  $N$  são as dimensões da imagem em pixel, enquanto  $A(x, y)$  e  $B(x, y)$  são os valores dos pixels das respectivas imagens  $A$  e  $B$ .

Para medir a razão de compressão [9] entre a imagem original e a imagem comprimida, utiliza-se a equação 10.

$$R = \frac{TO}{TC} \quad (10)$$

em que  $R$  é a razão de compressão,  $TO$  é a quantidade de *bytes* utilizada para armazenar a imagem original e  $TC$  a quantidade de *bytes* utilizada para armazenar a imagem comprimida.

A tabela 1 apresenta uma análise comparativa entre as diferentes técnicas de compressão fractal de imagens avaliadas. Os valores correspondentes à taxa e ao tempo de compressão são mostrados fixando-se um valor para a medida de qualidade. Pode-se observar que o uso da tabela de dispersão melhora significativamente os tempos de compressão,

Imagens	Taxa de Compressão			Tempo de Compressão (s)			PSNR (dB)
	<i>busca exaustiva</i>	<i>quadtree</i>	<i>quadtree+hash</i>	<i>busca exaustiva</i>	<i>quadtree</i>	<i>quadtree+hash</i>	
imagem médica							
512 × 512	24,89	22,19	24,16	15933,80	4858,60	36,28	28,6
256 × 256	14,65	14,24	15,11	2504,40	728,90	6,68	25,0
128 × 128	10,87	10,48	10,55	218,68	65,16	2,10	22,9
fotografia							
512 × 512	16,21	15,71	16,78	36656,24	9749,20	27,17	26,8
256 × 256	10,67	9,68	10,40	3965,12	971,90	7,84	24,5
128 × 128	7,10	7,05	7,05	357,62	90,13	2,70	22,8
terreno							
512 × 512	39,77	34,87	40,81	13424,01	3872,90	22,55	28,7
256 × 256	25,13	23,94	25,25	1672,19	458,65	6,02	27,3
128 × 128	19,21	18,74	19,18	130,72	38,48	1,90	26,8
cena natural							
512 × 512	12,78	12,00	12,67	48696,52	14539,00	30,26	26,5
256 × 256	9,33	8,80	9,26	4578,72	1258,50	8,12	25,1
128 × 128	7,21	7,18	7,18	370,72	105,27	2,69	24,3

**Tabela 1. Análise comparativa entre as diferentes técnicas de compressão avaliadas.**

enquanto mantém taxas de compressão compatíveis com os demais métodos.

## 6. Conclusões

A compressão fractal de imagens tem o objetivo de obter altas taxas de compressão, redução do esforço computacional e reconstrução de imagens de boa qualidade. Este trabalho apresentou o uso de tabela de dispersão para melhorar o desempenho dos métodos clássicos de compressão fractal baseados em busca exaustiva e em particionamento *quadtree*.

Para avaliar os resultados obtidos, utilizou-se o tempo de compressão, a taxa de compressão e a medida de qualidade PSNR. Embora tenham sido testadas várias medidas de similaridade, apenas a média de intensidade foi utilizada na comparação apresentada neste trabalho.

Propostas para trabalhos futuros incluem a comparação do método com outras abordagens de compressão com perdas, bem como a investigação da compressão fractal em vídeos.

## Referências

- [1] M. F. Barnsley and L. P. Hurd. *Fractal Image Compression*. A. K. Peters, Ltda., Natick, MA, USA, 1993.
- [2] T. Cormen, C. Rivest, H. Ronald, and C. Stein. *Algoritmos: Teoria e Prática*. Campus, Rio de Janeiro, 2 edition, 2002.
- [3] M. Ferraz. Codificação de Imagens. Dissertação de Mestrado, Instituto de Matemática Pura e Aplicada, Rio de Janeiro-RJ, 1998.
- [4] Y. Fisher. *Fractal Image Compression*. SIGGRAPH - Course Notes, San Diego, 1992.
- [5] Y. Fisher. *Fractal Image Compression - Theory and Application*. Springer-Verlag, New York, 1995.
- [6] A. Jaccquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. *IEEE Transactions on Image Processing*, 1:18–30, Janeiro 1992.
- [7] N. Koli and M. Ali. A Survey on Fractal Image Compression Key Issues. *Information Technology Journal*, 7(8):1085–1095, 2008.
- [8] K. Koroutchev and J. R. Dorronsoro. Hash-Like Fractal Image Compression with Linear Execution Time. In *Lecture Notes in Computer Science*, volume 2652, pages 395–402. 2003.
- [9] H. Pedrini and W. R. Schwartz. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Editora Thomson Learning, 2007.
- [10] J. P. I. F. Ribas and G. A. Carrijo. Um Modelo de Compressão de Imagens Digitais Baseado em Codificação Fractal e Quantização Vetorial (VQ). *XXX - Congresso Nacional de Matemática Aplicada e Computacional*, 2007.
- [11] A. L. M. Silva. *Procedimentos para Método Híbrido de Compressão de Imagens Digitais Utilizando Transformadas Wavelet e Codificação Fractal*. Tese de Doutorado, Universidade Estadual de Campinas, Campinas-SP, Maio 2005.
- [12] J. Silva and E. Flores. Compressão de Imagens Médicas usando Fractais. In *VI Workshop de Informática Médica*, Vila Velha-ES, Maio 2006.
- [13] L. Soberano. The Mathematical Foundation of Image Compression. *Honors Program in the Department of Mathematic and Statistic*, 2000. The University of North Carolina at Wilmington, Wilmington, North Carolina.
- [14] Texas Instruments Europe. An Introduction to Fractal Image Compression, Outubro 1997. Literature Number: BPRA065 - TI Europe.
- [15] N. V. Thakur and D. O. G. Kakde. Color Image Compression with Modified Fractal Coding on Spiral Architecture. *Journal of Multimedia*, 2(4), Agosto 2007.