7th Brazilian Conference on Dynamics, Control and Applications
May 07 - 09, 2008
FCT - Unesp at Presidente Prudente, SP, Brazil
DINCON' 2008

AVALIAÇÃO COMPUTACIONAL DA COMPRESSÃO FRACTAL DE IMAGENS UTILIZANDO A TÉCNICA *PARTICIONED ITERATED FUNCTION SYSTEM*

Rodolfo Ranck Junior [1], Marcilei A. Guazzelli [2]

[1] INPE – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, Brasil, rodolforanck@yahoo.com.br
[2] USP – Universidade de São Paulo, São Paulo, Brasil, shila@if.usp.br

Resumo: Neste trabalho, avaliamos o desempenho da compressão fractal de imagens pela técnica PIFS (*Particioned Iterated Function System*) quando os parâmetros de busca e de dimensionamento dos blocos de imagem/domínio são modificados. O fator de compressão é comparado e a qualidade das imagens é avaliada utilizando a métrica RMS. Para validar a técnica, um comparativo com o formato JPEG é realizado.

Palavras-chave: Compressão Fractal de Imagens, *Particioned Iterated Function System*, Análise Computacional.

1. INTRODUÇÃO

A compressão fractal é utilizada para reduzir a quantidade de dados necessários ao representar imagens digitais. Baseia-se na observação de que fragmentos de uma imagem podem representar outras partes desta mesma imagem. Pode-se dizer que a codificação fractal é um processo matemático utilizado para comprimir os pixels que contém as imagens do mundo real, como um grupo de dados que descrevem propriedades fractais da imagem, [1,2].

Usualmente, os fractais são formados por diversas cópias de si mesmo e podem ser construídos matematicamente por um sistema de função iterada conhecido por IFS - *Iterated Function System*, desenvolvido por Hutchinson. O problema inverso a este, constitui uma base para a compressão fractal: dada uma imagem inicial, qual é o seu fractal gerador? O teorema de colagem desenvolvido por Barnsley apresenta alguns caminhos para obter essa resposta, [1,2].

Quando o processo de construção de uma imagem é conhecido, este pode ser armazenado para uma posterior reconstituição. Entretanto, diferentemente do que ocorrem com os fractais, as imagens do mundo real não apresentam um processo de construção iterado, e não são formadas por cópias reduzidas de si mesmas.

Embora as imagens do mundo real não contenham as propriedades fractais geradas pela técnica IFS, uma imagem pode ter fragmentos que, após algumas transformações, podem ser utilizados para substituir outras regiões da mesma imagem. Estas transformações podem ser mapeadas e simplificadas de modo que qualquer imagem possa ter um processo de construção, diminuindo assim, o espaço necessário para armazená-la.

O método de compressão fractal é um método inovador, pois consiste em representar uma imagem através de um conjunto de transformações utilizando o conceito de auto-similaridade. Nesse tipo de codificação, ao invés de armazenar os blocos da imagem como uma coleção de pixels, procura-se somente armazenar coeficientes de transformações, ou seja, seu processo de construção.

A figura 1 mostra o processo de construção do triângulo de Sierpinski e pode ser utilizada para compreender a idéia principal da compressão fractal de imagens. Recursivamente, para cada imagem resultante, divide-se o triângulo branco em três outros menores mantendo o tamanho original da imagem. Essa transformação é aplicada repetidamente até que a imagem não seja mais alterada. Observando este processo, uma questão surge: para que armazenar a imagem deste fractal pixel a pixel se podemos simplesmente armazenar seu processo de construção?

Fig. 1. Processo de construção do triangulo de Sierpinsky

Esse questionamento permite introduzir a técnica IFS, a definição para o processo básico de construção de um fractal, e fundamenta as bases para o desenvolvimento da técnica PIFS, a técnica atualmente estudada e utilizada na compressão fractal de imagens.

1.2. IFS (Iterated Function System)

Michael Barnsley e Allan Sloan nos anos 80 apresentaram idéias sobre teorias fractais utilizando um sistema particular de mapeamento chamado de IFS. O objetivo era representar as transformações iteradas de uma imagem, de modo que essa pudesse ser interpretada como um fractal.

Para entender o principio de construção de fractais por funções iteradas (IFS), pode-se observar na figura 2, o fractal da folha de samambaia que parece ser complicado e intrincado, mas que pode ser reproduzido com um IFS relativamente simples devido à sua auto-similaridade global. Isto é, a imagem inteira é constituída de pequenas cópias de si mesma, [3].

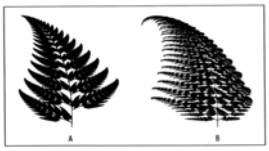

Fig. 2. Folhas de samambaia geradas com um sistema repetitivo de transformações

Fisher criou uma interessante analogia para descrever o funcionamento de um fractal e suas transformações: uma máquina copiadora, ao receber uma determinada imagem, realiza três cópias reduzindo-as juntas para formar a imagem de saída. O que acontece se a imagem de saída for inserida novamente como entrada? O que acontece se este procedimento for repetido uma "grande" quantidade de vezes? A figura 3 ilustra esse procedimento, [4].

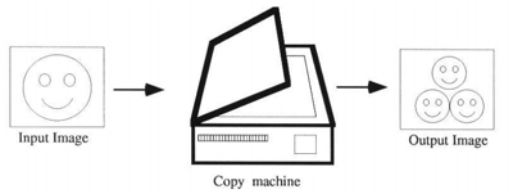

Fig. 3. Exemplo da foto-copiadora de Fisher

Conforme é repetido o processo, a imagem segue um determinado padrão em sua estrutura, que passa a ser inalterável e fixo. Independentemente da imagem de entrada a ser colocada na copiadora, ela convergirá para um atrator, conforme observado na figura 4.

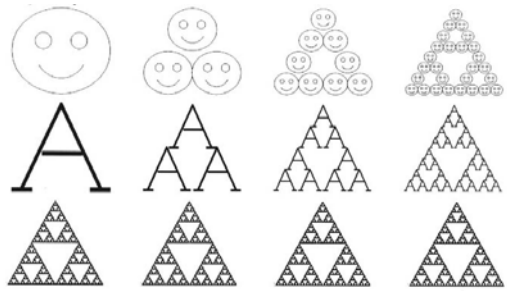

Fig. 4. Primeiras três cópias geradas pela foto-copiadora para três imagens diferentes

Na prática, estas transformações são polinomiais do tipo $f(x)=a.x+b$, tal que $a,b \in \Re$ e permitem gerar uma grande quantidade de atratores. Estas transformações são chamadas de afim e podem deslizar, rotacionar, escalar ou deslocar a imagem de entrada, dependendo dos valores dos coeficientes destas transformações.

A partir dos atratores produzidos por essas transformações, deseja-se representar um determinado fractal (imagem). Este é o principio que trata o teorema da colagem, [1].

A partir de uma forma na imagem, quer-se determinar um sistema dinâmico cujo atrator se pareça com o formato inicial. Isto é, a partir de um fractal inicial (imagem), deseja-se encontrar qual a IFS que o representa. Este é justamente o processo inverso de construção de um fractal.

1.3. RIFS (*Recurrent Iterated Function System*)

Essa técnica generaliza o sistema de função iterada IFS, e permite misturar e encaixar dois ou mais atratores para a construção de uma imagem. Além do processo original que leva a um atrator, há uma segunda IFS que copia o atrator para outros cantos da mesma imagem. A figura 5 apresenta esse processo considerando que os retângulos da imagem são os atratores.

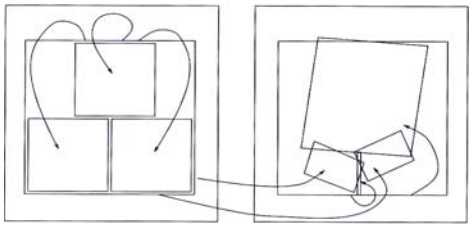

Fig. 5. Processo de mistura das "IFS" para construir uma imagem

Imagine que se deseja construir, conforme figura 6, uma samambaia utilizando o triângulo de Sierpinski para a construção das folhas. Seria necessário desarticular os planos fazendo com que os atratores assumam uma determinada seqüência de posição. Esta técnica permite um arranjo dos atratores em um nível global, [4].

Fig. 6. Samambaia construída com folhas de triângulo de Sierpinski através da técnica RIFS

1.4. PIFS (*Partioned Iterated Function System*)

Imagens do mundo real não possuem o tipo de similaridade apresentado até agora, isto é, não são formadas de cópias reduzidas de seu todo. Para essas, não é possível relacionar o todo com as partes de forma exata conforme sugere a técnica IFS. Entretanto, pode-se pensar que uma imagem é formada por fragmentos (blocos) transformados da mesma imagem, considerando a auto-similaridade nas partes que a formam.

É possível encontrar similaridades espalhadas na própria imagem e, como proposto pela técnica IFS, regras de construção podem ser formuladas, reduzindo a quantidade de dados a serem armazenados.

Dada esta observação, é interessante procurar por um determinado pedaço da imagem, ou bloco, que aplicada as devidas transformações, este possa se encaixar em algum outro local na mesma imagem substituindo o pedaço original. Esse é o mecanismo fundamental da técnica PIFS, [4,5].

Não é esperado que os blocos numa mesma imagem, mesmo após transformações, fiquem idênticos uns aos outros. Embora não seja possível encontrar um bloco que substitua outro perfeitamente, é possível encontrar um que o faça da melhor forma possível. Por este motivo, a compressão fractal de imagens utilizando a técnica PIFS, admite perda de qualidade na imagem, [4,5].

Fig. 7. Personagem comum na literatura da compressão de imagens, onde se verificam auto-similaridades em blocos que constituem a mesma imagem

Nesta técnica, dois tipos de blocos são considerados: os blocos de domínio, e os blocos de imagem. Os blocos de domínio são aqueles disponíveis a transformações e são candidatos a "substituírem" um bloco na mesma imagem. Os blocos de imagem são todos os blocos distintos que constituem a imagem, chamados também de blocos fundamentais.

O primeiro passo desta técnica é definir quais serão os blocos de domínio e qual será o seu tamanho. Definido estes parâmetros fundamentais, procura-se um bloco de domínio que, aplicada às devidas transformações, seja o mais semelhante possível ao bloco de imagem considerado. Ao encontrar este bloco, as coordenadas e transformações que levaram ao seu estado atual são armazenadas. Este processo é repetido até que, para todos os blocos fundamentais da imagem, haja um mapeamento correspondente, isto é, uma referência de como construir cada um dos blocos que constituem a imagem.

Este processo faz com que todas as informações para se obter uma reconstituição da imagem original a partir dos blocos de domínio e de um conjunto de regras de construção, sejam armazenadas. Espera-se que o método simplifique a imagem a ponto de render uma boa compactação de dados.

Como são procurados blocos que formem o melhor "encaixe" dentro de um critério estabelecido de semelhança, considera-se que na maioria das vezes será encontrada uma aproximação e não um bloco idêntico. Diante de tal consideração, sempre para um bloco a será possível encontrar outro bloco b na imagem tal que, após um conjunto de transformações W, $a \cong b$ cuja métrica $d(a, b)$ seja pequena o bastante, [4].

Conforme Fisher, blocos de domínio maiores que os blocos de imagem fornecem uma compressão de dados mais eficiente. As Transformações geométricas consideradas para os blocos candidatos da imagem são: rotação de 90, 180 e 270 graus; inversão do tipo vertical, horizontal, diagonal principal e diagonal secundária. Dentro da técnica PIFS as transformações geométricas são ainda combinadas com as transformações de brilho (o) e contraste (s), [4].

Considerando o mapeamento das transformações a ser realizado no processo de codificação, deseja-se encontrar a coleção de transformações $\omega_1, \omega_2, ..., \omega_n$ de acordo com (1):

$$\omega_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}, i = 1,2,...,n \quad (1)$$

Onde $\{a_i, b_i, c_i, d_i, e_i, f_i\}$ definem as transformações geométricas; s_i determina o contraste e o_i o brilho. Todos esses componentes compõem a transformação afim.

Dado que a_i e b_i representam os i pixels do bloco de imagem e do bloco de domínio respectivamente sendo comparados, para cada transformação geométrica serão procurados os valores para o e s que minimizam o erro médio quadrático entre os dois blocos candidatos utilizando a função (2).

$$R = \sum_{i=1}^{n} ((s \cdot a_i + o) - b_i)^2 \quad (2)$$

Para encontrar o mínimo da função em (2) dado a_i e b_i, é necessário fazer com que as derivadas parciais em o e s valham zero, obtendo as relações (3) e (4):

$$s = \frac{\left[n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i \right]}{\left[n\sum_{i=1}^{n} a_i{}^2 - \left(\sum_{i=1}^{n} a_i \right)^2 \right]} \quad (3)$$

$$o = \frac{1}{2}\left[\sum_{i=1}^{n} b_i - s\sum_{i=1}^{n} a_i \right] \quad (4)$$

Para cada dois blocos comparados na imagem é calculado em (2) o erro RMS respectivo através dos valores ótimos de o e s obtidos em (3) e (4). De forma equivalente, é possível obter o valor do erro R, resolvendo (5). O erro RMS é dado pela raiz quadrada de R.

$$R = \frac{\sum_{i=1}^{n} b_i{}^2 + no - 2\sum_{i=1}^{n} bi + s\left(s\sum_{i=1}^{n} a_i{}^2 - 2\sum_{i=1}^{n} a_i b_i + 2o\sum_{i=1}^{n} a_i \right)}{n} \quad (5)$$

A decodificação fractal para a técnica PIFS, assim como pelas outras técnicas apresentadas, é simples. Uma vez que os dados foram corretamente codificados, basta que o mapeamento codificado seja aplicado na imagem a ser descompactada, [4].

Uma das características da compressão fractal, é que a qualidade depende pouco da resolução final. No caso da técnica PIFS, a imagem é reconstituída por partes da imagem original independentemente da resolução final. Tal característica permite que imagens sejam decodificadas em tamanhos maiores que o original com muito mais realidade, [4].

1.4. JPEG (*Joint Photographic Experts Group*)

O formato padrão de compressão de imagens JPEG (Joint Photographic Experts Group), foi originalmente desenvolvido por Eric Hamilton da C-Cube Microsystems. Esse modelo é o padrão para aplicações gráficas, graças a sua qualidade na compressão, rapidez e padronização. Superá-lo é o principal desafio para outros modelos de compressão, principalmente para a compressão fractal de imagens, [6].

Um arquivo JPEG é um arquivo bitmap com unidades de dado de 8x8 pixels e 24 bits de cor. Nesse formato, após a subdivisão da imagem, existem cinco etapas até a compressão total da imagem:

1º etapa: Transformação da imagem no espaço RGB para o espaço YCbCr, para auxiliar na compressão de imagens coloridas;
2º etapa: Consiste no chamado *downsampling*, onde a parte da informação de cores é descartada;
3º etapa: Aplicação da transformada discreta de co-senos;
4º etapa: Quantização dos coeficientes da etapa anterior;
5º etapa: Codificação dos dados, [6].

A técnica de *downsampling* trabalha a redundância psicovisual da imagem e é baseada no fato de que o olho humano é muito mais sensível às informações de luminância do que as de cores. Deste modo, parte das informações de cores pode ser descartada, [6].

A tolerância do quantizador define a amostra dos dados e, portanto, vai definir a qualidade da imagem resultante após o processo de codificação, [6].

Na codificação dos dados muitos dos valores quantizados são bem pequenos, e são truncados para o valor zero. Uma grande quantidade de zeros passa então a ser esperada na imagem, principalmente para o caso de compressões mais altas. Por esse motivo, o comitê JPEG decidiu codificar esses bits usando o RLC (*Run length coding*), já que se torna conveniente armazenar apenas as corridas das seqüências desses zeros. Somente após a aplicação do RLC, é aplicada a codificação de Huffman de comprimento de código variável, fechando assim o esquema de compressão, [6,7].

Pode-se observar que o modelo de compressão JPEG é híbrido, pois utiliza vários algoritmos de compressão em etapas para constituir um único formato.

2. OBJETIVOS

A PIFS é uma importante técnica para a compressão fractal de imagens por ter se mostrado eficiente para codificar também as imagens do mundo real e não apenas imagens provenientes de fractais matemáticos. Entretanto, a literatura não apresenta um estudo de desempenho desta técnica quando seus parâmetros de tamanho de bloco de domínio e imagem são variados. Frente a isso, uma avaliação computacional desta técnica é proposta neste trabalho.

Nos testes, procurou-se mostrar como é alterada a qualidade e o tempo computacional em função dos parâmetros de dimensionamento dos blocos de domínio e de imagem. Foram verificadas as mesmas alterações quando o raio de busca computacional do algoritmo é modificado.

Um comparativo com o formato JPEG também é apresentado para validar a técnica e os testes utilizados.

3. METODOLOGIA

O software de compressão fractal, escrito em linguagem C, utiliza o principio da técnica PIFS descrito na sessão 1.4 e é baseado em Pulcini. O particionamento da imagem utilizado foi o retangular homogêneo, [8].

O funcionamento do algoritmo pode ser descrito em seis passos:

- Passo 1 - Particionar a imagem de acordo com o tamanho de bloco desejado, em pixels;
- Passo 2 - Escolher um valor *n* para o raio de busca;
- Passo 3 - Colocar todos os blocos que compõe a imagem como blocos do tipo imagem e os marcar como não codificados;

Os passos 4, 5 e 6 são repetidos para todos os blocos de imagem não codificados:

- Passo 4 - Encontrar em um raio de *n* blocos, o bloco de domínio que minimize a função de erro apresentada em (5) utilizando todas as transformações apresentadas em (1);
- Passo 5 - Armazenar o conjunto de transformações utilizadas para minimizar a função (5);
- Passo 6 - Marcar o bloco como codificado.

Os testes computacionais foram feitos com a imagem: “Ski” descrita na tabela 1. As imagens apresentadas em todos os testes foram codificadas utilizando o parâmetro, raio de busca = 10 blocos. As imagens podem ser observadas nas figuras 8, 9 e 10.

Para facilitar a visualização das imagens geradas em 4.1, 4.2 e 4.3, foi considerado um fragmento da imagem em particular, destacado na figura 8. A figura 9 corresponde ao fragmento utilizado para apresentar o resultado visual da imagem original.

Nos testes computacionais, baseado na observação de Fischer e seguido de ajustes empíricos, utilizou-se um bloco de domínio duas vezes maior que o bloco de imagem, [4].

As medidas de erro utilizadas, não são medidas de percepção de qualidade, para tal, a análise visual das figuras de saída dos testes é adequada.

No comparativo realizado entre o método de compressão fractal e o formato JPEG, foi utilizada a imagem “Papagaio” conforme tabela 1.

Tabela 1. Descrição das Imagens utilizadas nos testes.

Nome	Resolução (pixels)	Tamanho (Kb)
“Ski”	800x532	1247
“Papagaio”	698x523	1065

Fig. 8. Imagem "Ski" original de tamanho reduzido indicando a área da figura que será realçada nos testes

Fig. 9. Área realçada da imagem "Ski" original

Foram realizados testes de tempo computacional, erro e fator de compressão. Levou-se em conta o raio de alcance dos blocos considerados na busca, a resolução da imagem e o tamanho dos blocos de particionamento. Os testes foram divididos em três etapas de acordo com o tamanho dos blocos de imagem e domínio utilizados.

Os testes computacionais foram realizados em um computador de processador Intel Centrino Duo T2400, 1,83 Ghz operando com 1GB de memória RAM do tipo DDR2 à 987 Mhz;

4. RESULTADOS

4.1. Codificação com blocos de imagem=4 pixels e blocos de domínio=8 pixels

Neste primeiro teste optou-se por um tamanho de bloco pequeno com o objetivo de manter a qualidade da imagem original. Entretanto, tal configuração aumenta o número de comparações necessárias, aumentando também o tempo codificação.

Verificou-se (figura 10) que mesmo com raios de busca pequenos, a imagem é decodificada com boa qualidade. Novamente, o fato dos blocos serem pequenos incorre em mais blocos de domínio e mais mapeamentos a serem armazenados, reduzindo a taxa de compressão resultante.

Tabela 2. Resultados da codificação fractal para a imagem Ski, utilizando os blocos imagem=4 pixels e domínio=8 pixels.

Raio de busca (blocos)	Tempo (s)	Erro *RMS* (bits/pixel)	Taxa de compressão
1	1,04	5,04	1:7
2	1,19	4,7	1:7
3	1,5	4,41	1:7
4	1,97	4,21	1:7
5	3,02	4,09	1:7
10	9,04	3,71	1:7
20	29,95	3,3	1:7
30	66,1	3,11	1:7
40	116,1	3	1:7

Fig. 10. Área realçada da imagem "Ski" codificada com os blocos imagem=4 pixels, domínio=8 pixels e com o raio de busca = 10 blocos

4.2. Codificação com blocos de imagem=8 pixels e blocos de domínio=16 pixels

Neste segundo teste, os blocos foram duplicados de tamanho. O tempo de codificação foi reduzido e a taxa de compressão teve um grande aumento.

A imagem da figura 11 apresenta uma qualidade razoável com relação à imagem original. Os detalhes mais delicados da imagem começam a se perder.

Tabela 3. Resultados da codificação fractal para a imagem Ski, utilizando os blocos imagem=8 pixels e domínio=16 pixels.

Raio de busca (blocos)	Tempo (s)	Erro *RMS* (bits/pixel)	Taxa de compressão
1	0,98	12,48	1:26
2	1,1	12,12	1:26
3	1,3	11,81	1:26
4	1,5	11,57	1:26
5	1,79	11,45	1:26
10	3,51	11,15	1:26
20	12,05	10,93	1:26
30	26,02	10,73	1:26
40	52,69	10,64	1:26

Fig. 11. Área realçada da imagem "Ski" codificada com os blocos imagem=8 pixels, domínio=16 pixels e com o raio de busca = 10 blocos

4.3. Codificação com blocos de imagem=16 pixels e blocos de domínio=32 pixels

Nesta última avaliação, os blocos utilizados no teste anterior foram duplicados. Novamente o tempo de codificação foi reduzido e a taxa de compressão atingiu um valor muito alto.

A imagem da figura 12 apresenta uma qualidade baixa com relação à imagem original. Os detalhes da imagem, exceto regiões homogêneas como a do céu, são distorcidos devido ao tamanho dos blocos de comparação na imagem. Considerando o enorme fator de compressão, (1:104) a imagem, embora tenha perdido muita qualidade, apresenta ainda uma aparência natural.

Tabela 4. Resultados da codificação fractal para a imagem Ski, utilizando os blocos imagem=16 pixels e domínio=32 pixels.

Raio de busca (blocos)	Tempo (s)	Erro *RMS* (bits/pixel)	Taxa de compressão
1	0,95	14,48	1:104
2	1,1	14,1	1:104
3	1,2	13,77	1:104
4	1,4	13,49	1:104
5	1,7	13,33	1:104
10	3,1	12,9	1:104
20	8,07	12,59	1:104
30	18,9	12,4	1:104
40	31,85	12,26	1:104

Fig. 12. Área realçada da imagem "Ski" codificada com os blocos imagem=16 pixels, domínio=32 pixels e com raio de busca = 10 blocos

4.4. Análise dos dados gerados em 4.1, 4.2 e 4.3

Na figura 13 é apresentado um gráfico comparando o tempo computacional gasto em segundos em função do raio de busca utilizado para os testes realizados. Percebe-se que o tempo cresce exponencialmente nos três casos, entretanto de forma menos assintótica quando os blocos utilizados são maiores.

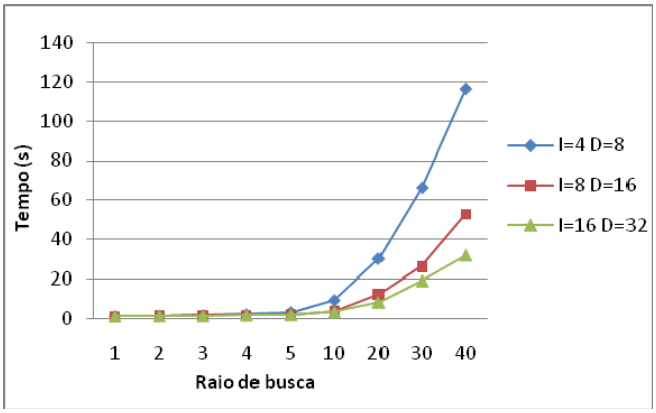

Fig. 13. Gráfico do tempo computacional em função do raio de busca utilizado para os três testes realizados

Na figura 14 é apresentado o aumento da qualidade em função da distância em que está sendo procurada uma similaridade na imagem. Os resultados mostram que existe uma maior probabilidade de encontrar uma similaridade logo nas proximidades do bloco de imagem sendo codificado.

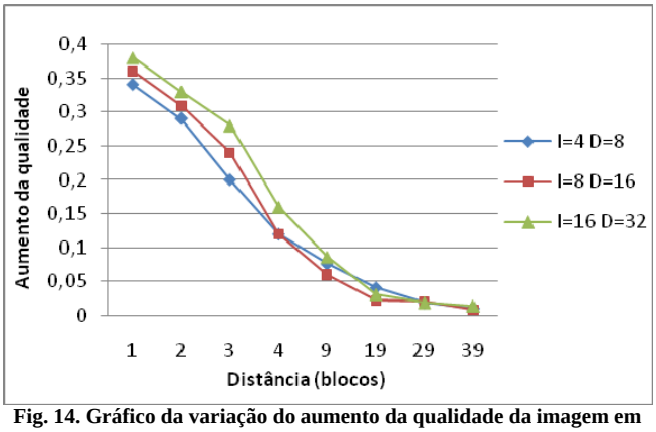

Fig. 14. Gráfico da variação do aumento da qualidade da imagem em função da distância do bloco de imagem codificado para os três testes realizados

4.4. Comparativo entre o método de compressão fractal e o formato JPEG

É interessante realizar um comparativo verificando a qualidade das imagens produzidas por estes dois codificadores, dado que o principal concorrente no mercado de compressão de imagens é o formato JPEG.

Consideramos neste teste uma taxa de compressão elevada, onde a compressão fractal produziu resultados significantemente superiores àqueles obtidas pelo formato JPEG nos quesitos qualidade e compressão de imagem. Em nossos testes, quando considerado taxas pequenas e médias de compressão (até aproximadamente 1:30) a compressão fractal apresentou resultados muito similares aos obtidos pelo formato JPEG, entretanto com um tempo computacional sempre maior.

A imagem "papagaio" original foi codificada pelo método fractal utilizando as mesmas configurações aplicadas no teste 3.3 e também foi codificada utilizando um software comercial de compressão JPEG. Após a codificação, as imagens foram decodificadas e seus resultados foram comparados.

A imagem de qualidade original é apresentada na figura 15. Os resultados visuais do comparativo entre os codificadores JPEG e fractal são apresentados respectivamente nas figuras 16 e 17. Os detalhes do comparativo são apresentados na tabela 5.

Não se julgou necessário realçar parte da imagem utilizada neste teste comparativo, dado que a diferença de qualidade apresentada ficou clara para o tamanho de imagem utilizado.

Tabela 5. Resultados da codificação fractal e JPEG para a imagem papagaio, utilizando os blocos imagem=16 pixels e domínio=32 pixels.

Codificadores	Tempo (s)	Erro *RMS* (bits/pixel)	Taxa de compressão
JPEG	-	16,49	1:107
Fractal	2,6	12,36	1:107

Fig. 15. Imagem "Papagaio" original

Fig. 16. Imagem "Papagaio" codificada pelo formato JPEG com um fator de compressão = 1:107

Fig. 17. Imagem "Papagaio" codificada pelo método Fractal com um fator de compressão = 1:107

Neste último teste, observou-se que pela técnica de compressão fractal a imagem decodificada apresentou maior fidelidade com relação à imagem original, enquanto que pelo formato JPEG, a imagem perdeu boa parte das informações de cores devido ao processo de quantização empregado neste método.

5. CONCLUSÃO

Os resultados obtidos permitem formular parâmetros do codificador fractal de imagens pela técnica PIFS procurando os critérios que melhor atendem qualidade e tempo computacional.

O trabalho contribui para que o método de compressão fractal receba a mesma quantidade de esforços que a DCT (*Discrete cossine transform*), principal técnica utilizada pelo JPEG, e possa ser utilizado em aplicações específicas no processamento de imagens.

Com o poder dos processadores atuais, a aplicação da compressão fractal de imagens passa a se tornar muito mais viável que há alguns anos atrás, quando iniciaram os seus primeiros estudos. Entretanto, devido à natureza intrinsecamente combinatória dessa técnica de compressão, novos métodos para diminuir a complexidade desta técnica devem ser utilizados. Em um trabalho futuro pretende-se considerar métodos mais inteligentes no sistema de busca, como uma pré-classificação dos blocos de domínio. Pretende-se considerar também um método de particionamento sensível a imagem, onde regiões que apresentem uma quantidade maior de detalhes possam ser particionadas com blocos menores, enquanto que regiões mais homogêneas possam utilizar blocos maiores a fim de poupar esforço computacional e melhorar a qualidade da imagem decodificada.

6. BIBLIOGRAFIA

[1] HUTCHINSON, J. E. Fractal and Self-Similarity. [S.l.]: Indiana University Mathematics Journal, Vol. 35, No. 5, 1981.

[2] BARNSLEY, M. F. Fractals Everywhere. 2.ed. San Diego, USA: Ed. Morgan Kaufmann, Academic Press, 1993.

[3] BARNSLEY, M. F. A. Sloan. A Better Way to Compress Images. [S.l.]: Byte, Jan, 1988.

[4] FISHER, Y, Fractal Image Compression: Theory and Application. New York, USA: Ed. Springer-Verlag, 1995.

[5] JACQUIN, A. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations – *IEEE Transactions on Image Processing*, vol. 01, no. 01, pp. 18-30, 1992.

[6] BLELLOCH, Guy E. Introduction to Data Compression. Carnegie Mellon University, 2001.

[7] HOFFMANN, Gernot. JPEG compression. f.5. 2006. Artigo. [S.l.]

[8] PULCINI, Giovambattista. IFSAF dlg lite. 1995. Software. Disponível em: http://www.verrando.com/pulcini/gp-ifsl.html. Acesso em: 15 de out. 2006.