Formatação de Histogramas para Cifragem de Imagens Digitais

Juliano B. Lima e Ricardo M. Campello de Souza

***Resumo*— Neste artigo, discute-se a formatação de histogramas de imagens digitais por meio da transformada do cosseno de corpo finito (FFCT). O procedimento proposto consiste em dividir a imagem em blocos e aplicar a FFCT, de maneira recursiva, a cada um desses blocos. Por meio de simulações, verifica-se que o histograma da versão transformada da imagem apresenta aspecto uniforme e que seus pixels possuem baixa correlação com seus vizinhos. A utilidade da técnica proposta no contexto de cifragem de imagens digitais é discutida.**

***Palavras-Chave*— Formatação de histograma, transformada do cosseno, cifragem de imagens.**

***Abstract*— In this paper, histogram shaping of digital images by means of the finite field cosine transform (FFCT) is examined. The approach consists in dividing an image into blocks and applying the FFCT, in a recursive manner, to each one of such blocks. Simulations of the procedure show that the histogram of the transformed image exhibits a uniform shape and their pixels have low correlation with their neighbors. The suitability of the proposed technique in the context of image encryption is discussed.**

***Keywords*— Histogram shaping, cosine transform, image encryption.**

I. Introdução

Nos dias atuais, a partilha de informação multimídia por meio da Internet e de outras redes de comunicação se tornou uma prática simples e bastante utilizada por usuários com diferentes perfis. Em determinados cenários, técnicas voltadas à proteção desse tipo de informação desempenham um importante papel, proporcionando transmissões com certa confidencialidade e assegurando a integridade dos dados recebidos. Esses são alguns dos motivos que têm feito crescer o interesse pelo estudo de esquemas de inserção de marcas d'água, esteganografia e cifragem para imagens, vídeos e áudio digitais [1], [2], [3].

Neste artigo, investiga-se uma técnica para formatar histogramas de imagens digitais. Aqui, entende-se por *formatação* a uniformização do histograma de uma imagem, resultante da aplicação de algum algoritmo ou transformação. A ideia é que um procedimento como este seja parte integrante de sistemas para cifragem de imagens digitais; ele seria incluído com o objetivo de ocultar o histograma original da imagem, o qual poderia ser usado em ataques criptográficos estatísticos.

Juliano B. Lima, Grupo de Pesquisa em Redes e Comunicações, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil, E-mail: juliano.bandeira@poli.br.

Ricardo M. Campello de Souza, Departamento de Eletrônica e Sistemas, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Recife, Brasil, E-mail: ricardo@ufpe.br.

A técnica de formatação proposta neste trabalho é baseada na transformada do cosseno de corpo finito (FFCT, *finite field cosine transform*), cuja aplicação requer operações envolvendo apenas aritmética modular. Isso significa que arredondamentos não são necessários e que os cálculos podem ser realizados de forma eficiente. Adicionalmente, o fato de a imagem com histograma formatado conter apenas números inteiros facilita sua codificação e, consequentemente, sua transmissão ou armazenamento.

Após esta introdução, na Seção II, são revisados alguns aspectos teóricos da FFCT. Na Seção III, é apresentada a técnica para formatação de histogramas proposta. Na Seção IV, são descritas as simulações realizadas e analisados seus resultados; diversos aspectos do procedimento são discutidos e comparações com outras técnicas são feitas. Finalmente, na Seção V, são apresentadas as considerações conclusivas deste trabalho.

II. A Transformada do Cosseno de Corpo Finito

O procedimento proposto neste artigo é baseado na transformada do cosseno de corpo finito (FFCT, *finite field cosine transform*). Esta ferramenta, que foi introduzida em [4], requer conceitos da trigonometria de corpo finito, alguns dos quais são apresentados a seguir.

Definição 1 (Função cosseno de corpo finito): Seja ζ um elemento não nulo de um corpo finito $\mathrm{GF}(p)$, p é um primo ímpar. A função trigonométrica cosseno de corpo finito relacionada a ζ é calculada módulo p por

$$\cos_\zeta(x) := \frac{\zeta^x + \zeta^{-x}}{2}, \tag{1}$$

$x = 0, 1, \ldots, \mathrm{ord}(\zeta)$, em que $\mathrm{ord}(\zeta)$ corresponde à ordem multiplicativa de ζ.

Noutros trabalhos, a definição do cosseno de corpo finito contém alguns detalhes adicionais, como a possibilidade de se tomar ζ pertencente a $\mathrm{GI}(p)$, o conjunto de inteiros Gaussianos módulo p, e a necessidade de se ter $p \equiv 3 (\mathrm{mod}\ 4)$ [4], [5]. No entanto, no presente contexto, esses fatos podem ser negligenciados. Em seguida, apresenta-se a definição da transformada do cosseno de corpo finito.

Definição 2: Se $\zeta \in \mathrm{GF}(p)$ possui ordem multiplicativa $2N$, então a transformada do cosseno de corpo finito do vetor $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]$, $x_i \in \mathrm{GF}(p)$, é o vetor $\mathbf{X} = [X_0, X_1, \ldots, X_{N-1}]$, $X_k \in \mathrm{GF}(p)$, de elementos

$$X_k := \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \beta_k x_i \cos_\zeta \left(k \frac{2i+1}{2} \right), \tag{2}$$

em que

$$\beta_r = \begin{cases} 1/\sqrt{2}, & r = 0, \\ 1, & r = 1, 2, \ldots, N-1. \end{cases} \quad (3)$$

O cálculo da FFCT de um vetor $\mathbf{x}$ pode ser representado pela equação matricial

$$\mathbf{X} = \mathbf{C}\mathbf{x}, \quad (4)$$

em que $\mathbf{C}$ corresponde à matriz de transformação cujos elementos são obtidos diretamente da Equação (2). Consequentemente, a FFCT de uma matriz $\mathbf{m}$ com dimensões $N \times N$, isto é, a versão bidimensional da transformada, pode ser obtida por

$$\mathbf{M} = \mathbf{C}\mathbf{m}\mathbf{C}. \quad (5)$$

Mostra-se que $\mathbf{C}\mathbf{C}^t = \mathbf{C}^t\mathbf{C} = \mathbf{I}$, em que $\mathbf{C}^t$ e $\mathbf{I}$ denotam, respectivamente, a matriz transposta de $\mathbf{C}$ e a matriz identidade. Isso significa que a FFCT inversa é obtida pelo uso da matriz de transformação $\mathbf{C}^t$ [5].

Um aspecto de grande importância para a aplicação descrita nas seções vindouras deste artigo é o que se denomina *período* da matriz da FFCT [6]. Esse parâmetro corresponde à menor potência l (inteira e positiva) que fornece $\mathbf{C}^l = \mathbf{I}$ ou, noutros termos, representa simplesmente a ordem multiplicativa da matriz $\mathbf{C}$ enquanto elemento do grupo $\mathrm{GL}(N, \mathrm{GF}(p))$. Diferentemente do que ocorre com outras transformadas, como, por exemplo, a de Fourier, cuja matriz de transformação sempre possui período igual a 4, a matriz da FFCT definida em (2) não apresenta regularidade em seu período.

Para investigar essa questão, pode-se escrever a matriz $\mathbf{C}$ como $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^t$, em que $\mathbf{U}$ é uma matriz unitária cujas colunas são autovetores de $\mathbf{C}$ e Λ é uma matriz diagonal cujos elementos são os autovalores de $\mathbf{C}$. Uma vez que $\mathbf{U}\mathbf{U}^t = \mathbf{I}$, tem-se $\mathbf{C}^l = \mathbf{U}\Lambda^l\mathbf{U}^t$, em que Λ^l é obtida elevando-se a l cada um dos elementos da diagonal principal de Λ. Então, l será o período de $\mathbf{C}$, se l for o menor múltiplo comum das ordens multiplicativas dos autovalores de $\mathbf{C}$. Conjectura-se que os autovalores da matriz $\mathbf{C}$ são todos distintos, além de poderem se encontrar em corpos de extensão [6]. Com isso, é possível obter matrizes da FFCT com períodos relativamente grandes em comparação aos de outras transformadas.

III. Formatação de Histogramas Via FFCT

Nesta seção, descreve-se o procedimento empregado para formatar o histograma de uma imagem digital pelo uso da transformada do cosseno de corpo finito. A ideia básica consiste em dividir a imagem original em blocos de tamanho fixo e, tomando-os sequencialmente da esquerda para a direita e de cima para baixo, aplicar a FFCT bidimensional um certo número de vezes. A quantidade de vezes que a transformada é recursivamente aplicada a cada bloco depende da FFCT utilizada. Este procedimento é ilustrado na Figura 1. Para recuperar a imagem original, basta realizar o mesmo procedimento, trocando a FFCT por sua inversa.

A descrição dos efeitos que a FFCT tem sobre o histograma da imagem tratada pode ser feita considerando alguns aspectos da aritmética modular. Se cada pixel de uma imagem em escala de cinza (8 bits) for multiplicado módulo p por uma

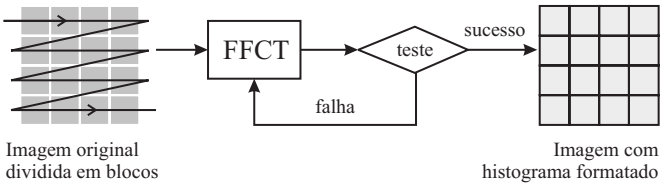


Fig. 1: Diagrama para formatação do histograma de uma imagem.

constante K, como resultado, observa-se um deslocamento das frequências de ocorrência dos símbolos desta imagem (números inteiros de 0 a 255). Se $K = 2$ e $p = 257$, por exemplo, a frequência de ocorrência do símbolo 200 estará, após a multiplicação modular, associada ao símbolo 143.

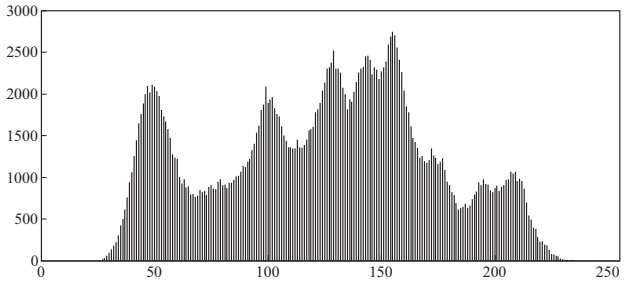
Quando o produto entre cada símbolo e uma constante K é substituído por uma combinação linear que envolve um bloco de símbolos (aplicação de uma transformada), o processo torna-se mais complexo. No entanto, o aspecto observado na situação apresentada como exemplo persiste. Ainda que seja considerável a diferença entre as freqüências de ocorrência dos símbolos de uma imagem, a tendência é que a aplicação da FFCT produza blocos transformados compostos por símbolos uniformemente distribuídos [7].

Neste artigo, são consideradas imagens em escala de cinza e, portanto, é necessário que se empregue uma transformada com $p \geq 257$ para processá-las. No entanto, quanto maior o número primo utilizado, mais bits serão necessários para representar a imagem após a aplicação da transformada. Para evitar isso, nos exemplos desenvolvidos, utiliza-se $p = 257$; uma FFCT com $N = 8$ e $\zeta = 128$, tal que $\mathrm{ord}(\zeta) = 16$, seria, então, calculada pela matriz de transformação

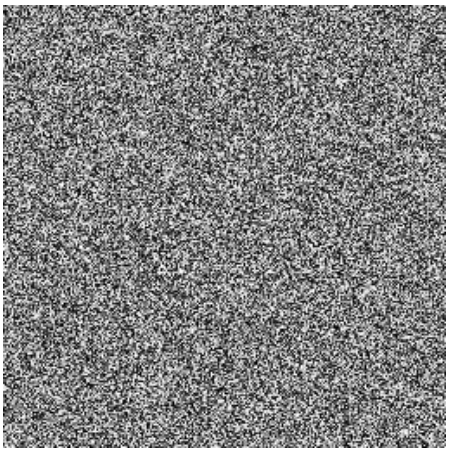
$$\mathbf{C} = \begin{bmatrix} 15 & 15 & 15 & 15 & 15 & 15 & 15 & 15 \\ 137 & 163 & 98 & 106 & 151 & 159 & 94 & 120 \\ 160 & 6 & 251 & 97 & 97 & 251 & 6 & 160 \\ 163 & 151 & 120 & 159 & 98 & 137 & 106 & 94 \\ 242 & 15 & 15 & 242 & 242 & 15 & 15 & 242 \\ 98 & 120 & 106 & 163 & 94 & 151 & 137 & 159 \\ 6 & 97 & 160 & 251 & 251 & 160 & 97 & 6 \\ 106 & 159 & 163 & 120 & 137 & 94 & 98 & 151 \end{bmatrix}. \quad (6)$$

Na Figura 2, é apresentada a imagem *lenna.bmp*, com tamanho 512×512, e o seu histograma original; além disso, apresenta-se a imagem após o processo de formatação ilustrado na Figura 1, com cada bloco tendo sido transformado uma só vez pela matriz da Equação (6), e seu respectivo histograma. Verifica-se que a figura, após a aplicação da transformada, tem toda a sua informação visual comprometida, apresentando um aspecto completamente ruidoso. Em relação ao novo histograma, observa-se que sua distribuição aparenta uniformidade, o que compromete, também, qualquer análise estatística que se deseje realizar sobre a imagem. É importante enfatizar que este resultado foi obtido pela aplicação da FFCT apenas uma vez a cada bloco da imagem, o que indica que, diferentemente de outras técnicas, a que este trabalho propõe não requer um grande número de iterações.

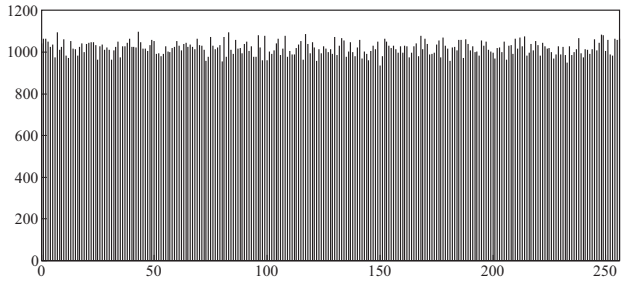
(a)



(b)



(c)



(d)

Fig. 2: Exemplo de formatação de histograma via FFCT. (a) Imagem original *lenna.bmp*; (b) Histograma de *lenna.bmp*; (c) Imagem com histograma formatado; (d) Histograma formatado.

A. Aplicação recursiva da FFCT

Conforme mencionado anteriormente, para formatar o histograma de imagens em escala de cinza, utiliza-se uma FFCT em que $p = 257$, como aquela cuja matriz de transformação é dada por (6). Embora a aplicação dessa FFCT a cada bloco de uma imagem uma única vez produza, do ponto de vista de formataçao do histograma, resultados como aqueles apresentados na Figura 2, ela traz implicações relacionadas à codificação da imagem. Particularmente, a imagem transformada pode conter pixels com valores iguais a 256, o que impede uma codificação em 8 bits.

Para contornar essa restrição, a ideia, já ilustrada na Figura 1, é aplicar a FFCT recursivamente a cada bloco da imagem, até que o resultado não contenha nenhum pixel com valor igual a 256. Naturalmente, no processo de recuperação da imagem original, a FFCT inversa também deve ser aplicada recursivamente, até que se obtenha um bloco que não contenha pixels com valores iguais a 256.

A fim de ilustrar o procedimento proposto, a imagem *lenna.bmp*, com tamanho 512×512, é novamente considerada. Após a aplicação recursiva da FFCT aos 4096 blocos de tamanho 8×8 da imagem, obtêm-se os resultados apresentados na Figura 3. Na parte (a) da referida figura, observa-se a imagem após a aplicação das transformadas e, na parte (b), o seu histograma. Os efeitos visuais são semelhantes àqueles observados na Figura 2 e o histograma original é visivelmente modificado.

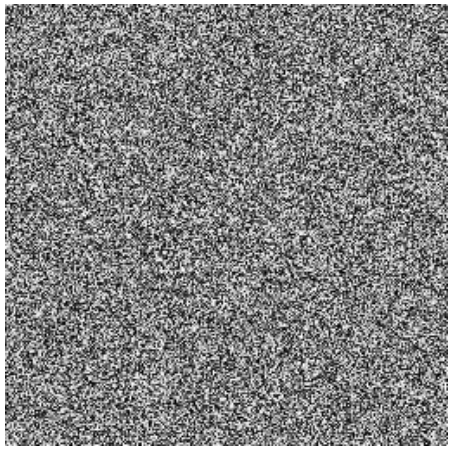
A Tabela I permite a avaliação da quantidade de blocos que precisou ser submetida a determinado número de aplicações da FFCT até que não se encontrassem pixels com valores iguais a 256. Quase 80% dos blocos já respeitava essa condição após uma única aplicação da FFCT; o número máximo de rodadas necessárias a um bloco foi 6, sendo que isso ocorreu no processamento de apenas 1 bloco. Como a matriz de transformação utilizada possui período $l = 16.974.594$ (obtido computacionalmente), não há qualquer risco de que, com o cálculo recursivo da FFCT, um bloco transformado retorne ao bloco original.

TABELA I: Números absolutos e percentuais de blocos da imagem *lenna.bmp* submetidos a aplicações recursivas da FFCT.

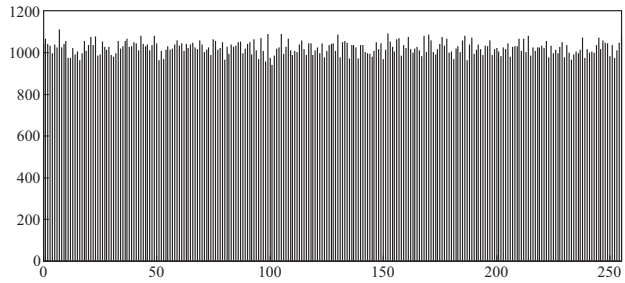
Aplicações recursivas da FFCT	Número de blocos	Percentual correspondente (%)
1	3.155	77,0264
2	715	17,4561
3	163	3,9795
4	53	1,2939
5	9	0,2197
6	1	0,0244

B. Métricas para avaliação da técnica proposta

Para a avaliação quantitativa dos resultados da técnica de formatação proposta, as métricas descritas a seguir são consideradas. A primeira delas é o *grau de diferença de tons de cinza* [8], que requer, inicialmente, o cálculo da diferença



(a)



(b)

Fig. 3: Exemplo de formatação de histograma via FFCT, aplicando a transformada recursivamente. (a) Imagem com histograma formatado; (b) Histograma formatado.

de tons de cinza entre um pixel e seus vizinhos (GN). Este parâmetro é dado por

$$GN = \frac{\sum[G(x,y) - G(x',y')]^2}{4}, \tag{7}$$

em que

$$(x',y') = \begin{cases} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{cases} \tag{8}$$

e $G(x,y)$ denota o valor do pixel na posição (x,y). O valor médio da diferença de tons de cinza entre um pixel e seus vizinhos, considerando todos os pixels de uma imagem $M \times N$ é

$$AN(GN(x,y)) = \frac{\sum_{x=2}^{M-1}\sum_{y=2}^{N-1} GN(x,y)}{(M-2)\times(N-2)}. \tag{9}$$

O grau de diferença de tons de cinza é, então, definido por

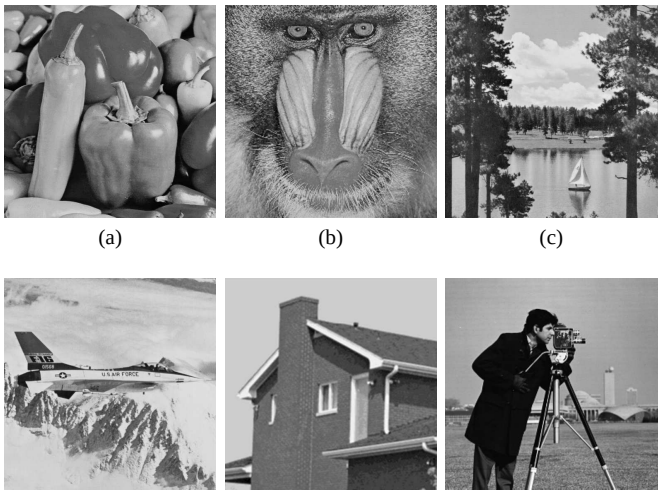
$$GVD = \frac{AN'(GN(x,y)) - AN(GN(x,y))}{AN'(GN(x,y)) + AN(GN(x,y))}, \tag{10}$$

em que AN e AN' denotam, respectivamente, o valor médio da diferença de tons de cinza da imagem original e o da imagem após a formatação do histograma. Quanto mais próximo de 1 for o valor de *GVD*, melhor terá sido o efeito do procedimento de formatação aplicado.

Outra métrica considerada é a *correlação entre dois pixels adjacentes* de uma imagem (a adjacência pode ser horizontal, vertical ou diagonal) [8]. Selecionando ao acaso P pixels da imagem, calcula-se o coeficiente de correlação por

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}}, \tag{11}$$

em que $cov(x,y) = \frac{1}{P}\sum_{i=1}^{P}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{P}\sum_{i=1}^{P}(x_i - E(x))^2$ e $E(x) = \frac{1}{P}\sum_{i=1}^{P} x_i$; x_i é o valor do i-ésimo pixel selecionado e y_i é o valor do pixel adjacente correspondente. Espera-se que uma imagem, anteriormente à formatação, possua coeficiente de correlação próximo de 1; o coeficiente de correlação da imagem com histograma formatado deve ser o mais próximo de 0 possível.



(a) (b) (c)

(d) (e) (f)

Fig. 4: Imagens em escala de cinza utilizadas nas simulações. (a) *peppers.bmp*; (b) *mandril.bmp*; (c) *lake.bmp*; (d) *jetplane.bmp*; (e) *house.bmp*; (f) *camera.bmp*. Todas as imagens têm tamanho 512×512 pixels.

IV. Simulações e Discussão

Nesta seção, são descritas as simulações realizadas para avaliar o método proposto e discutidos os seus resultados. Os experimentos foram executados no Matlab e, para isso, foram desenvolvidos programas para implementar a técnica de formatação de histograma apresentada na Seção III e, particularmente, para calcular as métricas apresentadas na Seção III-B. Além da imagem *lenna.bmp*, que já havia sido considerada anteriormente, foram usadas as imagens da Figura 4. Todas as imagens possuem tamanho 512×512 pixels e estão em escala de cinza. A matriz de transformação empregada é dada por (6).

Na Tabela II, para cada imagem de teste, são apresentados os números percentuais de blocos submetidos a determinado número de aplicações recursivas da FFCT, durante o processo de formatação. Conforme explicado anteriormente, o objetivo deste cálculo recursivo é eliminar pixels com valores iguais a 256, para que a codificação da imagem resultante permaneça a mesma da imagem original (8 bits por pixel). Os resultados obtidos são semelhantes aos da Tabela I. Para todas as imagens, quase 80% dos blocos precisaram ser transformados apenas uma vez. Um percentual muito baixo de blocos

TABELA II: Números percentuais de blocos das imagens de teste submetidos a aplicações recursivas da FFCT.

Aplic.	*mandril*	*camera*	*peppers*	*lake*	*house*	*jetplane*
1	78,2715	78,1738	78,1494	78,3447	83,0322	77,7344
2	17,3096	17,3340	17,5293	16,7236	13,4277	17,2607
3	3,3447	3,6133	3,4180	3,9307	2,7344	3,9795
4	0,7813	0,8057	0,6348	0,8301	0,6104	0,8057
5	0,2197	0,0488	0,1953	0,221	0,1221	0,1465
6	0,0488	0,0244	0,0488	0,0488	0,0488	0,0732
7	0,0244	–	–	–	0,0244	–
8	–	–	–	–	–	–
9	–	–	0,0244	–	–	–

TABELA III: Graus de diferença de tons de cinza (*GVD*) e coeficientes de correlação das imagens de teste originais (r_{xy}) e com histograma formatado ($\tilde{r}_{xy}$).

Métrica	*lena*	*mandril*	*camera*	*peppers*	*lake*	*house*	*jetplane*
GVD	0,9807	0,9519	0,9806	0,9782	0,9628	0,9940	0,9775
$r_{xy}(v)$	0,9726	0,9351	0,9832	0,9804	0,9768	0,9956	0,9734
$\tilde{r}_{xy}(v)$	-0,0061	0,0068	-0,0020	0,0006	0,0030	0,3533	-0,0018
$r_{xy}(h)$	0,9718	0,9328	0,9829	0,9800	0,9771	0,9954	0,9731
$\tilde{r}_{xy}(h)$	-0,0004	0,0052	0,0050	0,0006	-0,0166	0,3618	-0,0007
$r_{xy}(d)$	0,9728	0,9313	0,9831	0,9800	0,9774	0,9956	0,9724
$\tilde{r}_{xy}(d)$	-0,0102	-0,0106	-0,0124	-0,0109	-0,0081	0,3649	-0,0057

precisou ser transformado 3 vezes ou mais. Isso indica que a carga computacional extra, em função da aplicação recursiva da FFCT, é pouco significativa.

Na Tabela III, são apresentados os valores de *GVD* e dos coeficientes de correlação vertical, horizontal e diagonal para cada imagem de teste (utilizou-se $P = 32.768$). Conforme se pode verificar, a aplicação da FFCT às imagens proporciona valores de *GVD* bastante próximos de 1. Os números obtidos são melhores que aqueles alcançados pelo conhecido método de Arnold, que apenas embaralha os pixels e produz *GVD* em torno de 0,9 (este valor pode decair significativamente, em função do número de iterações realizadas) [8], [9]. Também em [8], propõe-se um método de cifragem baseado em sequências caóticas geradas por chaves secretas. Os *GVD* alcançados aqui são comparáveis aos daquele trabalho, no entanto, a aplicação da presente proposta envolve menor complexidade (dependendo do tamanho da imagem, a geração e a aplicação de uma sequência caótica de embaralhamento pode consumir um tempo de processamento significativo).

Para todas as direções e imagens consideradas, os valores de r_{xy} encontrados são bastante próximos de 1 e, com exceção da imagem *house.bmp*, os valores de $\tilde{r}_{xy}$ são bastante próximos de 0. O resultado destoante obtido para *house.bmp* se deve ao fato de esta imagem ter sido gerada com baixa resolução e com um menor número de cores (nas partes da imagem que contêm degradês, como, por exemplo, nas sombras dos telhados, observa-se que as transições entre os tons de cinzas são abruptas). Isso pode ter feito com que, nos blocos da imagem, houvesse pouca variação de intensidade entre os pixels, comprometendo o efeito de formatação pela FFCT, a qual é aplicada bloco a bloco. Ainda assim, os valores de $\tilde{r}_{xy}$, que ficaram em torno de 0,36, são plenamente distinguíveis dos respectivos valores de r_{xy}. De maneira geral, considerando os coeficientes de correlação calculados, o método proposto se comporta melhor que o de Arnold e de modo semelhante àquele descrito em [8].

V. Conclusões

Neste artigo, foi introduzido um procedimento para formatação de histogramas de imagens baseado na transformada do cosseno de corpo finito. Conforme se enfatizou ao longo do texto, o propósito da técnica, que deve ser parte de um sistema maior para cifragem de imagens, é dificultar a implementação de ataques que explorem a frequência de ocorrência dos pixels. Este tipo de ataque é factível, quando são utilizadas técnicas de cifragem baseadas em mudanças nas posições dos pixels ou noutro tipo de transformação que não altere significativamente o histograma da imagem. Foram apresentadas simulações que indicam que o esquema proposto possui potencial aplicabilidade em cenários práticos, produzindo resultados comparáveis aos de outras técnicas com finalidades semelhantes. A união da técnica descrita com outros procedimentos que dependam de chaves, para composição de um esquema de cifragem completo, tem sido investigada.

Agradecimentos

O desenvolvimento deste trabalho foi apoiado financeiramente pela Fundação de Amparo à Ciência e Tecnologia de Pernambuco (FACEPE), com recursos do processo APQ 1196-3.04/10.

Referências

[1] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, March 2010.

[2] M. A. Suhail and M. S. Obaidat, "Digital watermarking-based DCT and JPEG model," *IEEE Trans. on Instrumentation and Measurement*, vol. 52, no. 5, pp. 1640–1647, October 2003.

[3] C.-P. Wu and C.-C. Jay Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, October 2005.

[4] M. M. C. de Souza, H. M. de Oliveira, R. M. Campello de Souza, and M. M. Vasconcelos, "The discrete cosine transform over prime finite fields," in *International Conference on Telecommunications*, J. N. de Souza, P. Dini, and P. Lorenz, Eds., Berlin, 2004, Lecture Notes in Computer Science, pp. 482–487, Springer.

[5] J. B. Lima and R. M. Campello de Souza, "New trigonometric transforms over prime finite fields for image filtering," in *Proceedings of the VI International Telecommunications Symposium, ITS'06*, Fortaleza, Brasil, 2006.

[6] J. B. Lima, R. M. Campello de Souza, and D. Panario, "Blind sequence separation based on the eigenstructure of finite field transforms," in *Anais do XXVI Simpósio Brasileiro de Telecomunicações, SBrT'08*, Rio de Janeiro, Brasil, 2008.

[7] J. B. Lima, H. M. de Oliveira, and R. M. Campello de Souza, "Formatação de distribuições de probabilidade sobre os inteiros," in *Anais do XXV Simpósio Brasileiro de Telecomunicações, SBrT'07*, Recife, Brasil, 2007.

[8] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347–354, April 2010.

[9] G. Ye, X. Huang, and C. Zhu, "Image encryption algorithm of double scrambling based on ASCII code of matrix element," in *Proc. 2007 International Conference on Computational Intelligence and Security*, Harbin, China, 2007, pp. 843–847.