

Reconhecimento de texturas por LZW e Co-Ocorrência

1. Introdução

O objetivo deste trabalho é tentar estabelecer um conjunto de características que possibilite um discernimento entre vários tipos de texturas, levando em conta a existência da técnica de reconhecimento de texturas através do **algoritmo LZW**.

O reconhecimento de texturas em imagens é uma atividade importante, que faz parte de uma vasta gama de aplicações em diversas áreas. Dentre elas podemos citar o diagnóstico médico, a automação industrial, o reconhecimento biométrico, sensores remotos e outras.

A idéia inicial do reconhecimento de texturas, assim como outras atividades da área de análise de imagens, consiste em extrair da imagem algumas características que permitam realizar posteriormente um discernimento, uma tomada de decisão ou uma classificação.

No caso da técnica que utiliza o algoritmo LZW, a partir de um conjunto inicial de texturas de exemplo, iremos compor uma base de características identificadoras para cada um dos tipos de textura a serem analisados futuramente, realizando, desta forma, um “treinamento”, antes de efetivamente iniciar o processo de classificação das texturas.

2. Pré-processamento da imagem

Após a aquisição das imagens, percebemos a necessidade de realizar um pequeno ajuste, de forma a compatibilizar algumas distorções que podem ocorrer durante a etapa de aquisição das imagens, principalmente por parte de diferenças de iluminação de ambiente e dos dispositivos de captura, o que poderia comprometer futuramente o processo de reconhecimento das texturas durante as etapas de reconhecimento e comparação das características das imagens.

O pré-processamento escolhido neste caso foi a realização de uma **equalização global do histograma da imagem**, o que deve ser aplicado tanto para as imagens utilizadas na etapa de treinamento quanto em qualquer imagem a ser classificada posteriormente pela técnica.

2.1 Equalização global do histograma

A equalização global do histograma consiste em executar um processo de reorganização dos tons de cinza presentes na imagem, redistribuindo-os de forma que estes fiquem o mais homoganeamente distribuídos possível por toda

a faixa de tons de cinza disponível (mais próximo de um histograma ideal com igual numero de pixels para cada tom).

Inicialmente é calculado o histograma da imagem e em seguida um outro histograma, que é o histograma ideal ou esperado para a imagem. Ambos os histogramas são colocados na forma de histograma acumulado (as quantidades de tons de cinza vão sempre crescendo até que no final do histograma você tem a quantidade igual ao total de pixels da imagem).

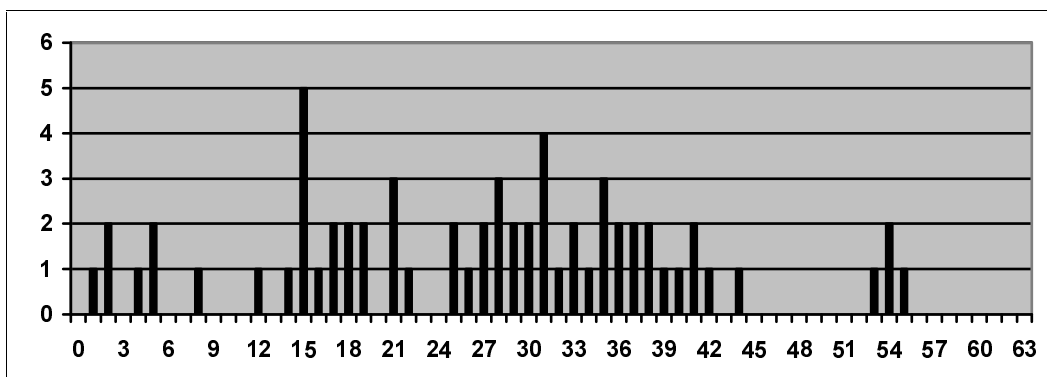
Em seguida, os valores quantitativos de cada tom de cinza presentes no histograma acumulado original são então comparados com os do histograma acumulado ideal, de forma a gerar um terceiro histograma que é o histograma equalizado da imagem, e que possui os tons de cinza melhor distribuídos pela faixa de tons de cinza representado na resolução de cores da imagem (2,4,8,16,32,64,128,256,512...).

Este processo é realizado tanto para as texturas de treinamento quanto para as desconhecidas, de forma a facilitar um futuro discernimento dentre as classes de texturas existentes e evitar problemas ocasionados por variações de luz ambiente no momento da captura das texturas.

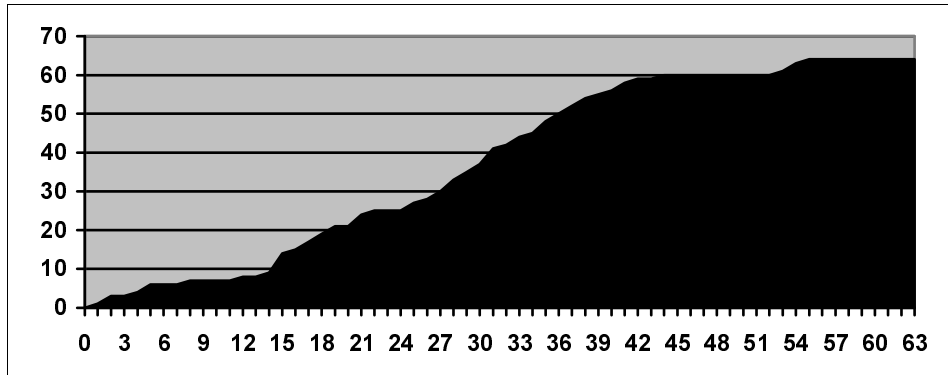
Abaixo segue um exemplo de equalização de histograma. Considere que a imagem original seja a imagem abaixo de 64 pixels (8x8), sendo cada um dos números a intensidade de cinza presente naquele pixel da imagem (que varia de 0 a 63) e que estejamos querendo gerar o histograma equalizado desta imagem:

1	28	32	33	18	54	37	37
29	38	35	14	2	25	27	39
41	53	34	31	31	19	38	15
31	27	17	29	19	15	42	15
21	33	22	18	28	5	41	15
28	21	54	55	4	35	40	15
5	30	21	17	16	35	36	12
44	8	30	31	36	26	25	2

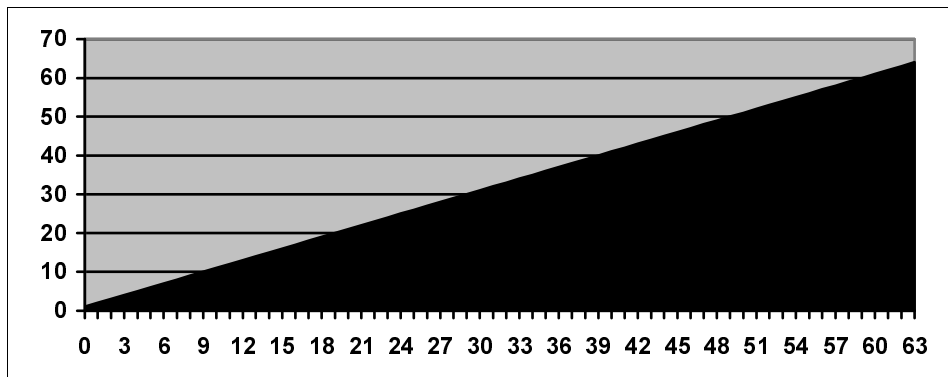
imagem (8x8 = 64 pixels) : cada pixel com uma intensidade de cinza



histograma da imagem



histograma acumulado da imagem

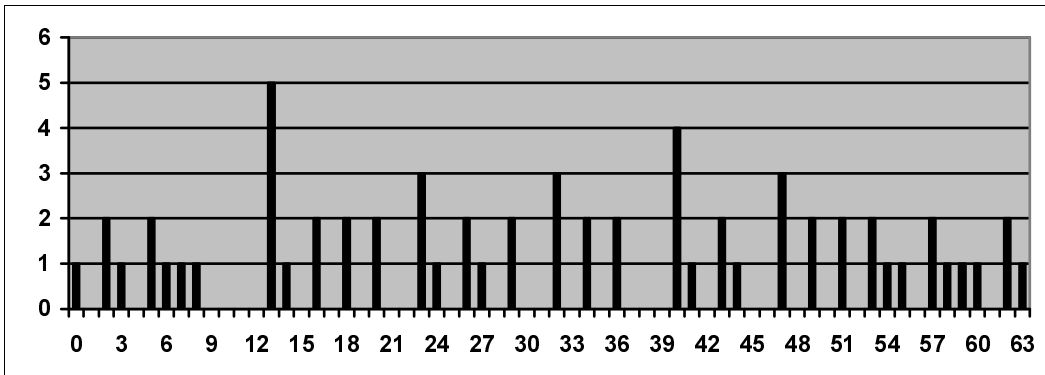


histograma acumulado desejado

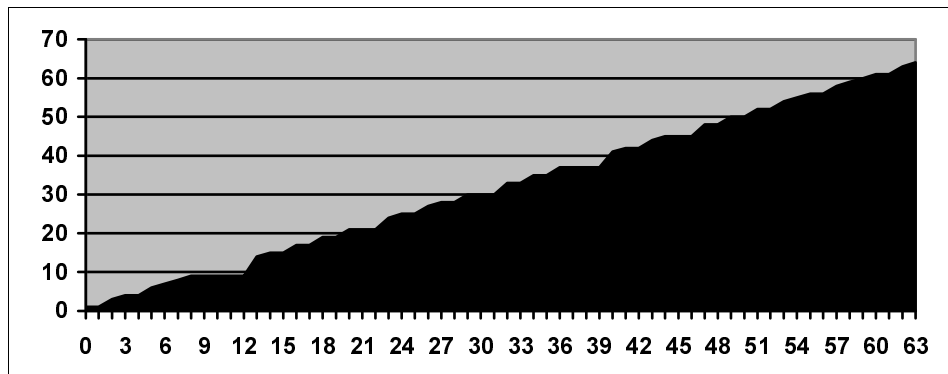
Tabela de Equalização

Tom Original	Qtd.Real	Qtd. Desejada	Tom Resultante
0	0	1	0
1	1	2	0
2	3	3	2
3	3	4	2
4	4	5	3
5	6	6	5
6	6	7	5
7	6	8	5
8	7	9	6
9	7	10	6
10	7	11	6
11	7	12	6
12	8	13	7
13	8	14	7
14	9	15	8
15	14	16	13
16	15	17	14
17	17	18	16
18	19	19	18
19	21	20	20
20	21	21	20
21	24	22	23

22	25	23	24
23	25	24	24
24	25	25	24
25	27	26	26
26	28	27	27
27	30	28	29
28	33	29	32
29	35	30	34
30	37	31	36
31	41	32	40
32	42	33	41
33	44	34	43
34	45	35	44
35	48	36	47
36	50	37	49
37	52	38	51
38	54	39	53
39	55	40	54
40	56	41	55
41	58	42	57
42	59	43	58
43	59	44	58
44	60	45	59
45	60	46	59
46	60	47	59
47	60	48	59
48	60	49	59
49	60	50	59
50	60	51	59
51	60	52	59
52	60	53	59
53	61	54	60
54	63	55	62
55	64	56	63
56	64	57	63
57	64	58	63
58	64	59	63
59	64	60	63
60	64	61	63
61	64	62	63
62	64	63	63
63	64	64	63



histograma da imagem após a equalização



histograma acumulado da imagem após a equalização

3. Vetor de características

Após a realização de algumas pesquisas relacionadas ao reconhecimento de texturas, avaliamos que, para obter uma classificação eficiente das texturas, se faz necessária a extração de algumas características importantes da imagem original, de forma a possibilitar a correta distinção e classificação das texturas.

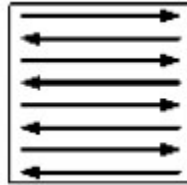
Para este caso em especial, selecionamos algumas características que poderiam se mostrar interessantes na classificação de texturas. Na realidade, há uma gama muito maior de características que pode ser analisada. Estas características abaixo são apenas alguns exemplos de parâmetros para auxiliar na classificação das texturas. São elas:

3.1 Dicionários de codificação das texturas exemplo

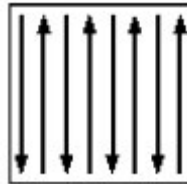
Estes dicionários são construídos durante o processamento de cada uma das texturas exemplo pelo algoritmo LZW, compondo uma listagem de palavras que definirá uma determinada classe de textura ao final do processamento. A princípio, os dicionários podem ser subdivididos em 2 tipos:

Horizontal: O dicionário de palavras horizontal é construído aplicando-se o algoritmo LZW sobre a textura exemplo, que percorre a imagem linha a linha,

da esquerda para a direita, indo e voltando, até a realizar a leitura completa dos dados e construir o dicionário. Podemos utilizar o símbolo H_n para definir um dicionário horizontal de uma textura exemplo n.



Vertical: O dicionário de palavras vertical é construído aplicando-se o algoritmo LZW sobre a textura exemplo, que percorre a imagem coluna a coluna, de cima para baixo, indo e voltando, até a realizar a leitura completa dos dados e construir o dicionário. Podemos utilizar o símbolo V_n para definir um dicionário vertical de uma textura exemplo n.



3.2 Taxa de codificação (code rating) do algoritmo LZW para cada textura processada

A taxa de codificação ou code rating do algoritmo para uma determinada textura é a quantidade de bits que o algoritmo vai precisar gastar para armazenar cada uma das palavras que compõem o dicionário de palavras, seja este horizontal ou vertical. E quanto maior for essa taxa de codificação, maior será a quantidade de palavras que precisaram ser criadas no dicionário de forma a poder armazenar de forma codificada toda a informação referente a uma textura.

Partindo de um dicionário composto somente de 4 palavras, por exemplo, precisaríamos apenas de 2 bits para armazenar cada uma destas palavras do dicionário:

- 00 - sapo
- 01 - cobra
- 10 - casa
- 11 - cachorro

Ou seja, após codificar os dados de um texto qualquer a partir deste dicionário, teríamos como resultado final vários números de 2 bits um após o outro, simbolizando cada uma das 4 palavras presentes no texto e que foram encontradas no dicionário, possibilitando então a codificação destas.

3.2.1 O algoritmo LZW

O algoritmo LZW é um algoritmo de codificação de palavras cuja idéia principal é ir construindo um dicionário de símbolos ou palavras conforme o texto ou a informação vai sendo processada pelo algoritmo. No início da codificação de um texto (ou imagem), o dicionário LZW possui inicialmente como palavras apenas os 256 caracteres da tabela ASCII (ou, no caso de uma imagem, a quantidade de tons que compõe a paleta da imagem).

Durante o processo de análise do texto, o dicionário vai criando novas palavras a partir de combinações de pelo menos 2 caracteres (ou tons) encontrados no texto (ou na imagem). A cada nova combinação encontrada, é criada uma nova entrada no dicionário que será utilizada para codificações futuras. Ao final do processo de codificação, teremos um dicionário de combinações de palavras ou tons que poderá ser utilizado para codificar o texto ou a imagem, possivelmente reduzindo a quantidade de espaço gasto para seu armazenamento.

O algoritmo LWZ funciona da seguinte forma:

```
p = nulo;
enquanto ( leia um caracter c )
{
    se pc existir no dicionário
        p = pc;
    senão
        adicione pc ao dicionário;
        exiba o código para p;
        p = c;
}
```

Onde p é a palavra ou combinação de tons a ser codificada e c o caracter ou tom do pixel atual analisado.

Exemplificando, se quiséssemos codificar a seqüência de caracteres "**^WED^WE^WEE^WEB^WET**", por exemplo, teríamos ao final da execução do algoritmo o seguinte dicionário de palavras:

p	c	saída	índice	símbolo
nulo	^			
^	W	^	256	^W
W	E	W	257	WE
E	D	E	258	ED
D	^	D	259	D^
^	W			
^W	E	256	260	^WE
E	^	E	261	E^
^	W			
^W	E			
^WE	E	260	262	^WEE
E	^			
E^	W	261	263	E^W
W	E			
WE	B	257	264	WEB
B	^	B	265	B^

^	W			
^W	E			
^WE	T	260	266	^WET
T	EOF	T		

Ou seja, após a execução do algoritmo, teríamos um dicionário composto dos símbolos de 0 a 255 da tabela de caracteres ASCII e mais algumas entradas correspondendo a combinações de caracteres encontradas no texto, indo de 256 (correspondendo à combinação W) até a entrada 266 (correspondendo à combinação WET). Desta forma, ao utilizar este novo dicionário para codificar o texto, estaríamos reduzindo uma entrada de 17 caracteres (o texto) para uma entrada de 7 códigos, sendo que cada código precisaria apenas de 1 bit a mais (9 bits) para ser armazenado, ao invés de 8 bits. Também seria necessário algum espaço para armazenar as novas entradas criadas no dicionário.

3.3 Entropia da imagem

A entropia da imagem pode ser definida como um número quantificador da randomicidade da imagem, ou seja, quanto maior for este número, mais irregular, atípica ou despadronizada será a imagem analisada. O cálculo da entropia E de uma imagem x pode ser realizado através da seguinte fórmula:

$$E_x = \sum_{i=0}^{M-1} (p_i \log_2 \left(\frac{1}{p_i} \right))$$

Onde M é o número total tons diferentes presentes na imagem e p_i é a probabilidade do i -ésimo tom ser utilizado novamente.

Para calcular a entropia através da ferramenta Matlab, utilizamos os seguintes comandos:

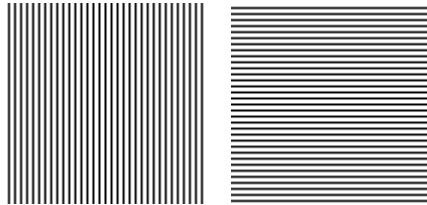
```
I = imread('c:\listras_h.gif');
E = entropy(I)
```

Para uma imagem contendo um único tom de cinza em toda a sua extensão (uma imagem contendo um único tom de cinza, por exemplo), temos que sua entropia é igual a 0 (zero):



Imagem Cinza - único tom: Entropia = 0

Para outras imagens com um padrão de repetição e mais de um tom de cinza, temos uma entropia baixa:



Listras horizontais ou verticais pretas e brancas – 2 tons: Entropia = 0.9149

E imagens sem um padrão de repetição apresentam uma entropia maior:

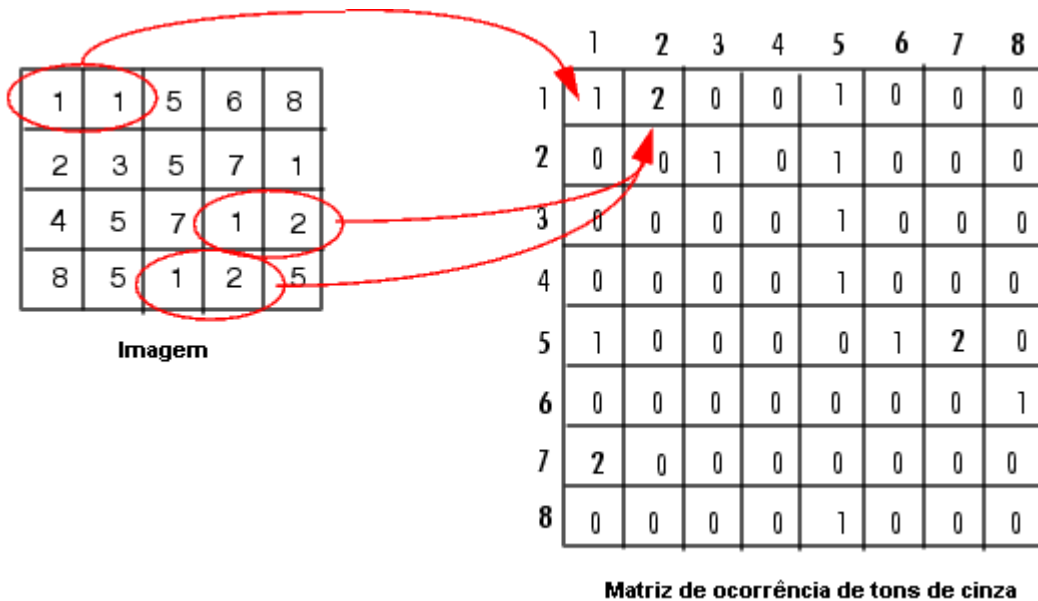


Mapa colorido – vários tons: Entropia = 2.6300

3.4 Matriz de co-ocorrência de tons de cinza

É uma matriz bidimensional, gerada a partir da contagem das ocorrências das combinações entre os tons de cinza do pixel analisado de um pixel vizinho.

Para cada uma das combinações possíveis entre o pixel e o pixel vizinho (normalmente o pixel à direita, por default), será criado um elemento na matriz de co-ocorrência de tons de cinza, de forma que o tamanho da matriz de co-ocorrência será de $t \times t$, onde t é o número de tons presentes na imagem original. A figura abaixo exemplifica melhor como ocorre o processo de construção da matriz de co-ocorrência de tons de cinza:



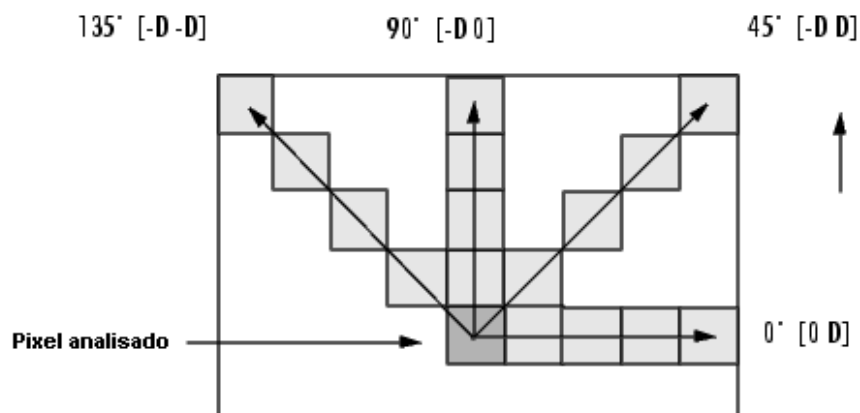
Repare que ao elemento (1,1) da matriz de co-ocorrência de tons de cinza foi associado um valor 1. Este valor 1 está representando a quantidade de ocorrências da combinação [1 1] existentes na imagem original. Como essa combinação [1 1] ocorre apenas uma única vez em toda a imagem, fica armazenado no elemento (1,1) o valor 1. Já para o elemento (1,2), foi colocado o valor 2, que simboliza a existência de 2 ocorrências da combinação [1 2] na imagem original.

A partir da matriz de co-ocorrência de tons de cinza, algumas características interessantes podem ser obtidas. São elas:

Característica	Descrição	Fórmula Matemática
Contraste	Retorna uma medida do contraste entre as intensidades de um pixel analisado e do pixel vizinho. A comparação é realizada em todos os pixels da imagem. Para uma imagem constante (mesmo tom de cinza em toda a extensão), o contraste é 0 (zero).	$\sum_{i,j} i - j ^2 p(i, j)$
Correlação	Retorna uma medida de quão correlacionado está um pixel com o seu vizinho. A comparação é realizada em todos os pixels da imagem. Faixa de valores possíveis: -1 a 1 A correlação é 1 para uma imagem totalmente correlacionada ou -1 para uma completamente descorrelacionada.	$\sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) p(i, j)}{\sigma_i \sigma_j}$

Energia	<p>Retorna a soma dos elementos elevados ao quadrado dentro da matriz de co-ocorrência de tons de cinza.</p> <p>Faixa de valores possíveis: 0 a 1</p> <p>A energia possui valor 1 para uma imagem constante (mesmo tom de cinza em toda a sua extensão).</p>	$\sum_{i,j} p(i,j)^2$
Homogeneidade	<p>Retorna um valor que representa a proximidade da distribuição dos elementos em relação à diagonal da matriz de co-ocorrência dos tons de cinza.</p> <p>Faixa de valores possíveis: 0 a 1</p> <p>Um valor de Homogeneidade 1 representa uma matriz diagonal de co-ocorrência de tons de cinza.</p>	$\sum_{i,j} \frac{p(i,j)}{1+ i-j }$

O pixel vizinho a ser comparado com o pixel analisado, normalmente é o pixel a direita (default), mas podemos utilizar qualquer pixel vizinho, inclusive alguns pixels com distância de vizinhança maior que 1. Também podemos trabalhar com mais de um pixel vizinho, criando uma matriz de vizinhança. A figura abaixo explicará melhor como essa matriz de vizinhança pode ser construída.



Com base na figura acima, poderíamos então, gerar a seguinte matriz de vizinhança:

```
matriz_vizinhanca = [ 0 1; 0 2; 0 3; 0 4;...
                    -1 1; -2 2; -3 3; -4 4;...
                    -1 0; -2 0; -3 0; -4 0;...
                    -1 -1; -2 -2; -3 -3; -4 -4];
```


Para a geração do gráfico acima, foi utilizada a matriz de vizinhança [0 1; 0 2; 0 3; 0 4], que corresponde a uma vizinhança de variando entre 1 e 4 pixels à direita (eixo horizontal). Repare que para uma vizinhança de apenas 1 pixel, podemos visualizar um valor de correlação em torno de 0,5. Este valor vai decaindo à medida em que a distância entre o pixel e o seu vizinho aumenta, resultando num valor próximo de zero quando a distância entre o pixel analisado e o pixel vizinho é de 4 pixels. O gráfico mostra então que há uma certa correlação entre os pixels mais próximos da imagem, mas que a imagem como um todo não é homogênea (vizinhanças mais distantes possuem correlação próxima de zero).

Para gerar o gráfico no Matlab, foram necessários os seguintes comandos:

```
I = imread('c:\matriz.gif');  
  
vizinhanca = [0 1; 0 2; 0 3; 0 4];  
  
GLCM = graycomatrix(I,'Offset',vizinhanca);  
stats = graycoprops(GLCM,'Correlation');  
figure, plot([stats.Correlation]);  
title('Correlação x Vizinhança');  
ylabel('Correlação');  
xlabel('Vizinhança');
```

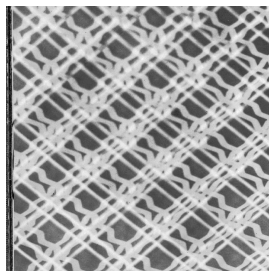
4. O processo

O processo de reconhecimento das texturas está subdividido em 2 etapas, que são o treinamento e a classificação:

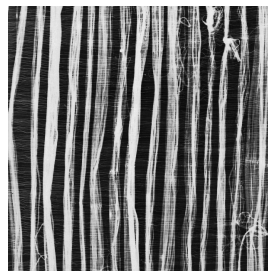
4.1 Treinamento

O treinamento é uma etapa preliminar, onde algumas texturas já conhecidas serão submetidas ao algoritmo de compressão LZW, produzindo como resultado 2 dicionários de palavras (horizontal e vertical) e em seguida será calculada também a entropia da imagem e sua matriz de co-ocorrência dos tons de cinza. Todos estes dados serão armazenados para possibilitar definirmos cada uma das classes de textura possíveis durante o processo de classificação das texturas.

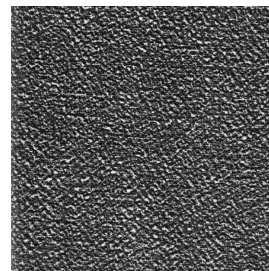
Como exemplo de classificação, iremos utilizar as texturas abaixo:



Textura 1



Textura 2



Textura 3

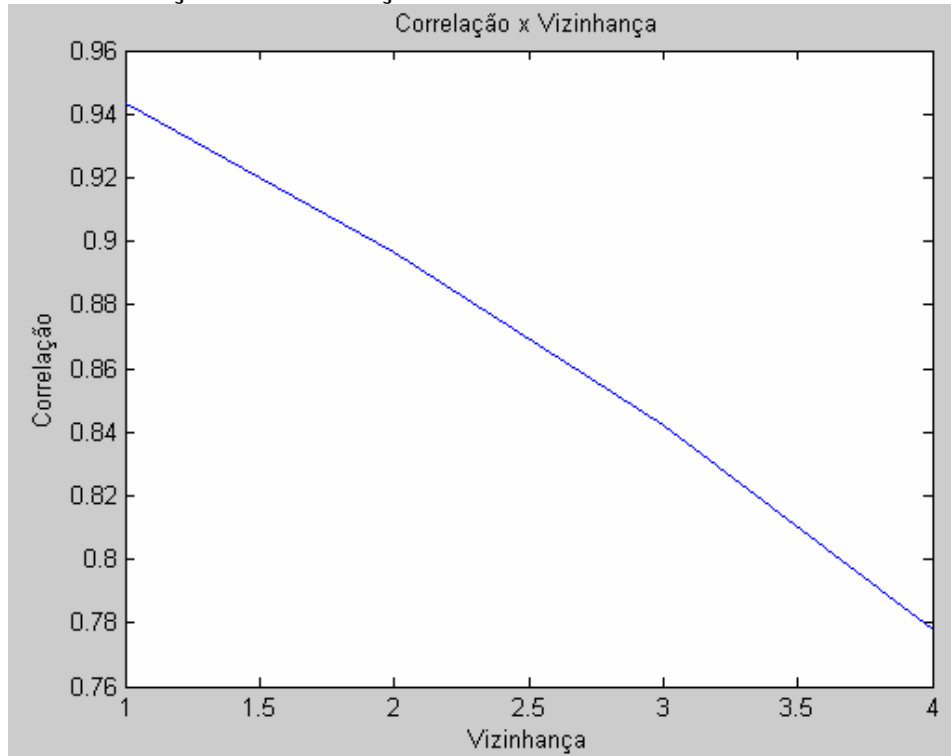
Entropia:

Textura1 = 5.8766

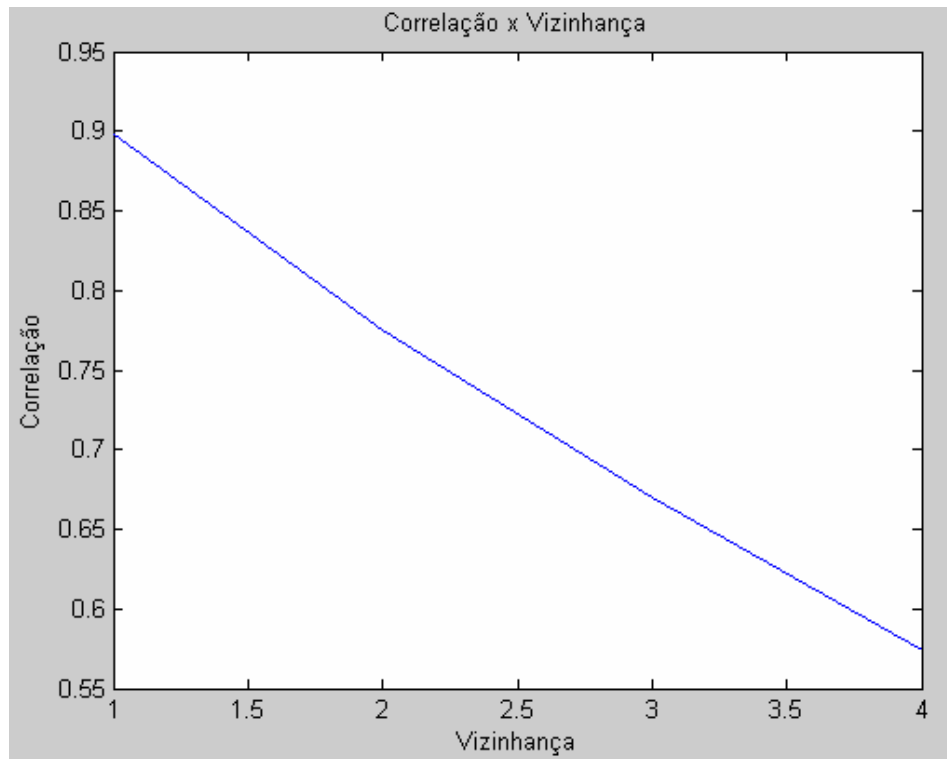
Textura2 = 5.9851

Textura3 = 6.2731

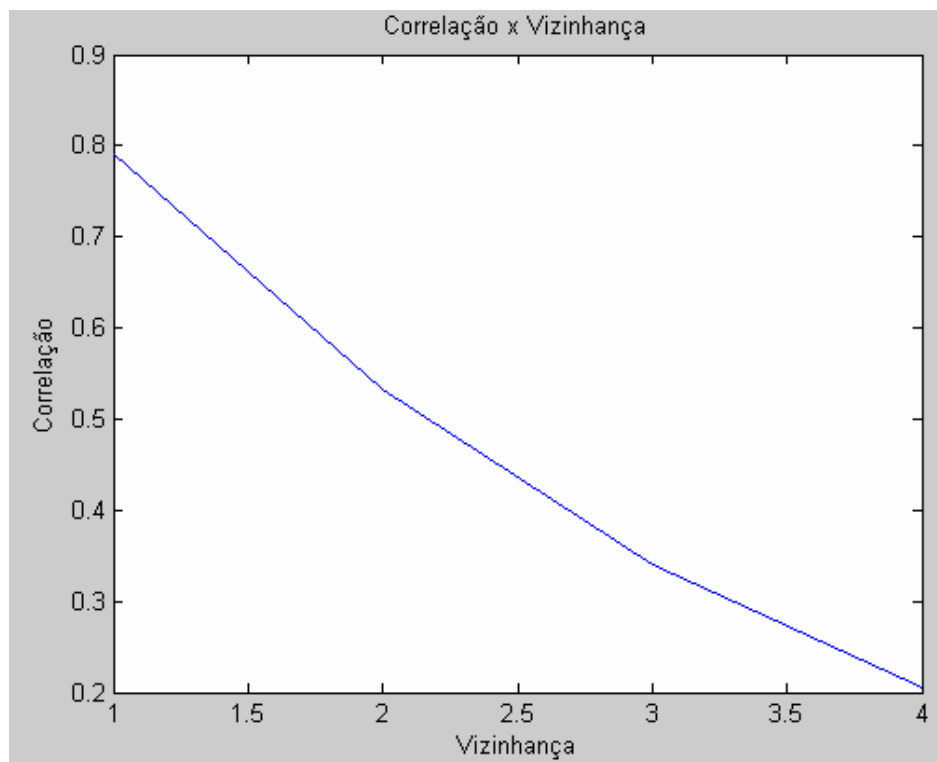
Gráfico de Correlação x Vizinhaça:



Textura 1 - Correlação x Vizinhaça



Textura 2 - Correlação x Vizinhaça



Textura 3 - Correlação x Vizinhaça

4.2 Classificação de texturas

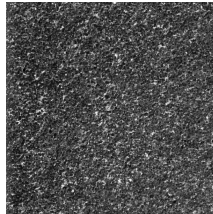
A etapa de classificação de texturas ocorre no momento seguinte, onde uma textura desconhecida é submetida para ser analisada e classificada em uma das classes de textura estabelecidas anteriormente na etapa de treinamento.

Nesta etapa, são realizadas várias codificações da textura a ser analisada utilizando em cada codificação os dicionários LZW horizontal e vertical correspondentes à cada uma das classes de texturas que foram definidas na etapa de treinamento. Em seguida é calculada a taxa de codificação (r_i) a partir dos dicionários horizontal (h_i) e vertical (v_i), para cada uma dessas codificações a partir da seguinte fórmula:

$$r_i = \frac{h_i + v_i}{2}$$

Por último, calcularemos a entropia da textura e geraremos o seu gráfico de Correlação x vizinhança, comparando estas características com as existentes para cada uma das classes de textura e definindo a classe da textura analisada por proximidade.

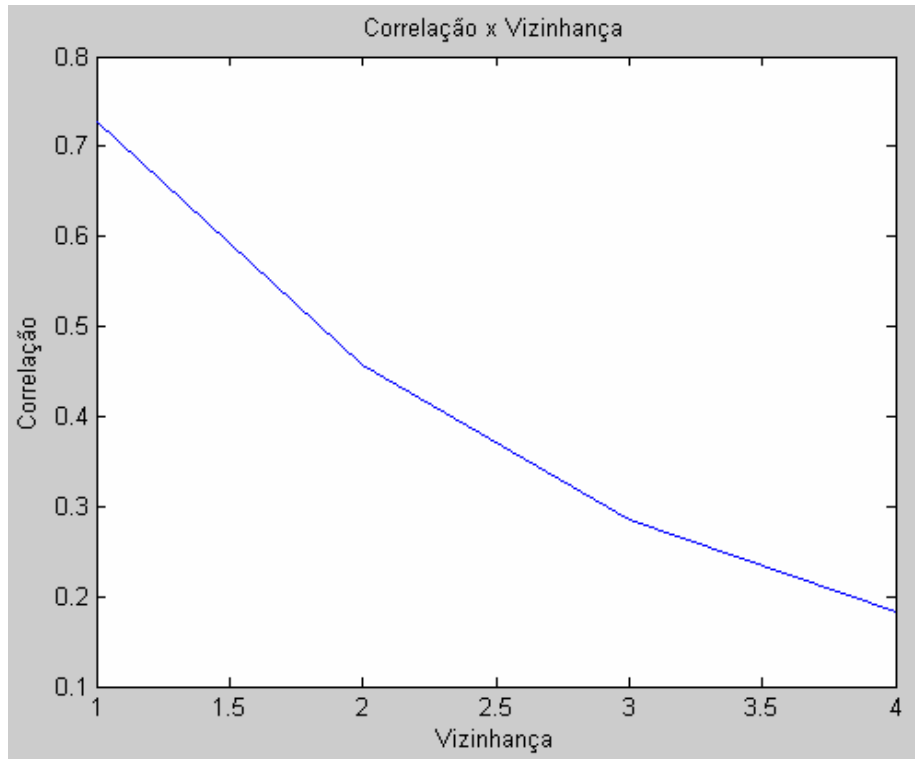
Para tal, tomaremos como exemplo as texturas abaixo, de classes desconhecidas:



Textura desconhecida 1

Entropia da textura: 6.0881

Gráfico Correlação x Vizinhança:



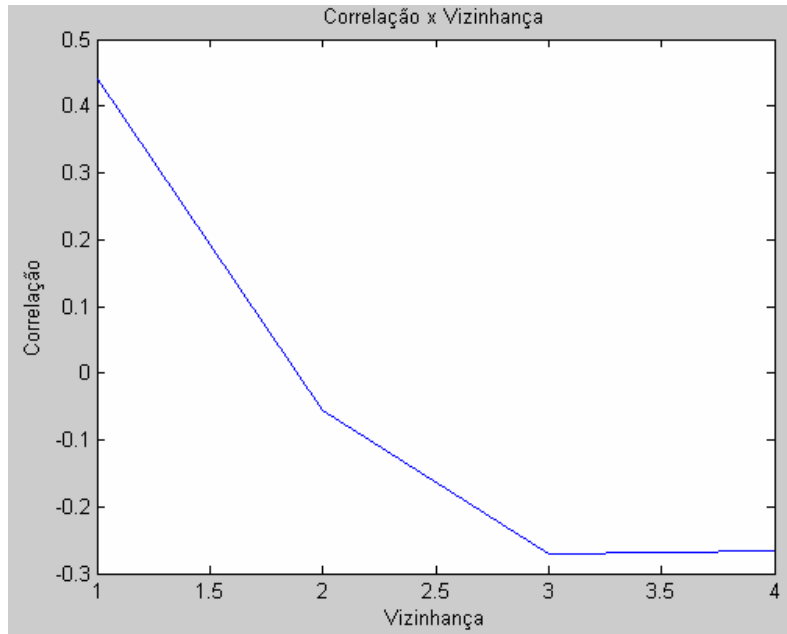
Textura desconhecida 1 – Correlação x Vizinhança



Textura desconhecida 2

Entropia da textura: 5.1305

Gráfico Correlação x Vizinhança:

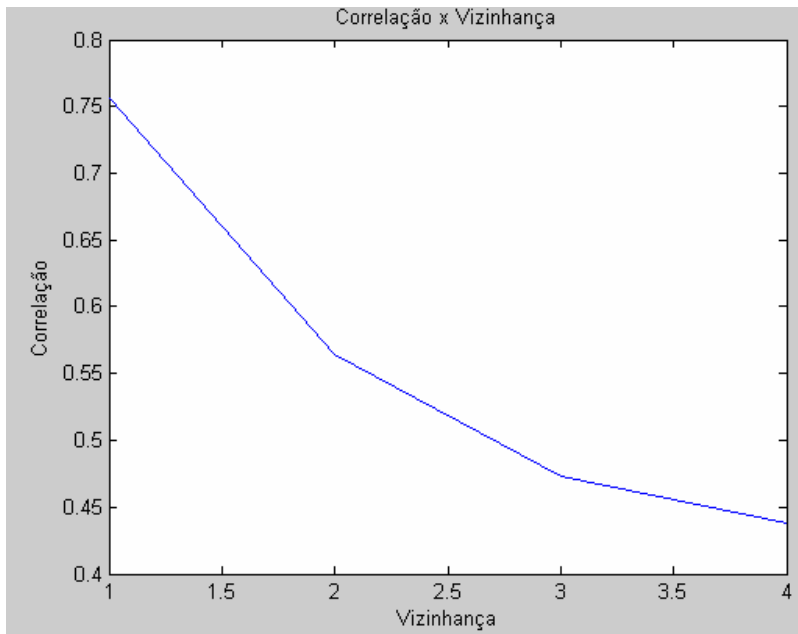


Textura desconhecida 2 – Correlação x Vizinhaça



Textura desconhecida 3

Entropia da textura: 6.1882



Textura desconhecida 3 – Correlação x Vizinhança

5. Conclusões

Após calcular a entropia das texturas na etapa de treinamento, percebemos que os valores encontrados (5.8766, 5.9851 e 6.2731) foram muito próximos, não possibilitando uma diferenciação nítida entre as classes de texturas. Desta forma, avaliamos que seriam necessárias outras características para definir melhor as classes de texturas, e que a característica entropia, se considerada isoladamente, não se mostra como uma boa opção para realizar as distinções entre as classes de texturas.

Ao gerar os gráficos de correlação x vizinhança, notamos uma diferença razoável entre os gráficos gerados para cada uma das texturas, o que nos fornece alguns subsídios para tentar realizar uma melhor distinção entre as classes de texturas. Mesmo assim, percebemos ainda uma insuficiência de parâmetros para discernir completamente a classe de algumas texturas (a textura desconhecida 3, por exemplo).

Acredito que após a utilização do algoritmo LWZ e da taxa de codificação para classificar as texturas, provavelmente este problema não estaria mais ocorrendo

Referências:

- Leonardo Vidal Batista, Moab Mariz Meira e Nicomedes L. Cavalcanti Júnior, "Texture Classification using Local and Global Histogram Equalization and the Lempel-Ziv-Welch Algorithm", pp. 297-302, Fifth International Conference on Hybrid Intelligent Systems (HIS'05), 2005.
- <http://homepages.inf.ed.ac.uk/rbf/HIPR2/> (tutorial com tudo sobre análise de imagens)

- <http://www.cs.cf.ac.uk/Dave/Multimedia/node214.html> (algoritmo de compressão LZW)
- <http://www.dogma.net/markn/articles/lzw/lzw.htm> (compressão LZW)
- <http://www.ux.his.no/~tranden/brodatz.html> (texturas brodatz)
- <http://www2.imm.dtu.dk/courses/02503/F03/Lectures/Gabor.pdf> (método Gabor para reconhecimento de texturas)
- <http://research.microsoft.com/~viola/Pubs/MIT/debonetNonParametricCVPR98.pdf> (reconhecimento de texturas através de modelo estatístico multi-escalar e não paramétrico)
- <http://www.cs.ucsd.edu/~sjb/ICPR94-texture.pdf> (reconhecimento de texturas invariantes a rotação – método da pirâmide)
- <http://debut.cis.nctu.edu.tw/pages/TextureStudy/classification.htm> (classificação de texturas)
- <http://www.cs.bris.ac.uk/Publications/Papers/1000023.pdf> (classificação de texturas usando transformada de fourier multiresolução)
- http://www.mff.cuni.cz/veda/konference/wds/contents/pdf05/WDS05_009_i2_Vacha.pdf (similaridade em texturas)
- http://www.teses.usp.br/teses/disponiveis/18/18133/tde-26042001-125441/publico/Capitulo_7.pdf (métodos para classificação de texturas)
- <http://cnx.rice.edu/content/m11088/latest/> (entropia da imagem)
- <http://www.mathworks.com/access/helpdesk/help/toolbox/images/enhanc15.html#33707> (matriz de co-ocorrência de níveis de cinza)
- <http://www.mathworks.com/access/helpdesk/help/toolbox/images/graycoprops.html> (características derivadas da matriz de co-ocorrência de níveis de cinza)