

REFERÊNCIAS

- [1] Mather, Paul M. *Computer Processing of Remotely-Sensed Images. An Introduction*, Second Edition, School of Geography, The University of Nottingham, UK, 1999.
- [2] Leão, Ruth, *GTD – Geração, Transmissão e Distribuição de Energia Elétrica* [Apostila]. Universidade Federal do Ceará Centro de Tecnologia, Departamento de Engenharia Elétrica. 2009.
- [3] Gonzalez, R.C. and R. E. Woods- *Digital Image processing*, Addison Wesley, Pub. Co. 1993 – ISBN 0-201-60078-1.
- [4] Jahne, Bernd. *Digital Image Processing*. 5th revised and extended edition. Springer. University of Heidelberg, Interdisciplinary Center for Scientific Computing. Germany. 2007.
- [5] McAndrew, Alasdair. *An Introduction to Digital Image Processing with Matlab: Notes for SCM2511 Image Processing 1*. School of Computer Science and Mathematics. Victoria University of Technology. Semester 1, 2004.
- [6] MINISTÉRIO DA CIÊNCIA E TECNOLOGIA. Instituto Nacional de Pesquisas Espaciais. [S.I. s.n. 200—]. Disponível em: <<http://www.inpe.br>>. Acesso em fevereiro de 2001.
- [7] Elvidge, Christopher D.; Lunetta, Ross S. *Remote sensing Change Detection*. Environmental Monitoring Methods and Applications, Ed. Arbor Press, 1998.
- [8] Furnas Centrais Elétricas S.A. [S.I.: s.n., 200—]. Disponível em: <http://www.furnas.com.br/hotsites/sistemafurnas/usina_hidr_serramesa.asp>. Acesso em 16 de agosto de 2009
- [9] Bastos, Vanessa Parodes. *Técnicas de Segmentação de Imagens para Recuperação de Informações Visuais*. Universidade Católica de Pelotas[S.I.: s.n., 200—]. 9 p. Disponível em: <[http:// paginas.ucpel.tche.br/~vbastos/index.htm](http://paginas.ucpel.tche.br/~vbastos/index.htm)>. Acesso em: 18 de out 2008.
- [10] Smith, Stephen M.; Brady, J. Michael. *SUSAN—A New Approach to Low Level Image Processing*. International Journal of Computer Vision 23(1), 45–78 (1997)
© 1997 Kluwer Academic Publishers. Manufactured in The Netherlands.

- [11] Viapiana, Evandro Luís; EngeL, P. M. *Um estudo comparativo de aplicação de técnicas de segmentação em imagens reais*. Semana Acadêmica UFRGS.[1998]. Disponível em: <www.inf.ufrgs.br/pos/SemanaAcademica/Semana98/evandro.html> Acesso em: 12 maio 2008.
- [12] Wangenheim, Aldo Von. *Introdução a Visão Computacional*. The Cyclops Project PPGCC - INE – UFSC. 2004.[S.I.: s.n., 200—]. Disponível em: <<http://www.inf.ufsc.br/~visao/morfologia.pdf>>. Acesso em julho de 2008.
- [13] Merlet, Nicloas; Zerubia, Josiane. *New prospect in Line Detection by Dynamic Programming*;; IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL 18, NO. 4, APRIL 1996
- [14] Bueno, Marcelo Lemes. *Detecção de Bordas através de Algoritmo Canny*. [S.I.: s.n.] Disponível em: <<http://www.inf.ufsc.br/~awangem>>. Acesso em 27 de agosto de 2008.
- [15] Canny, J. *A computational approach to edge detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), pp 679 – 698 (1986).
- [16] Nevatia, R; Babu, K. R., *Linear feature extraction and description*. CGIP, vol. 13, pp. 257-269, 1980.
- [17] Burns, J.B.; Hanson, A.R.; Riseman, E. M., *Extracting straight lines*. Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 425-455, 1986.
- [18] Russ, J. C. *The Image Processing Handbook*. CRC .1992.
- [19] Castleman, K.R. *Digital Image Processing*. Prentice Hall. 1970.
- [20] Wangenhain, Aldo Von; Rahal, Nelson Abu Samra Junior. *Reconhecimento de Elementos Geométricos em Imagens Computacionais*. [S.I.: s.n., 1999?]. Disponível em: <<http://www.inf.ufsc.br/~visao/1999/abu/index.html>>. Acesso em nov. 2008.
- [21] Kubiça, Stefano; Lethelier, Edouard; *Processamento de imagens de documentos - Parte III*. [S.I.: s.n., 1999?]. Disponível em: <<http://www.pr.gov.br/batebyte/edicoes/1999/bb90/imagens.html>>. Acesso em 13 mar. de 2008.
- [22] Jamundá, Teobaldo, *Reconhecimento de Formas: A Transformada de Hough*, Seminário Visão Computacional - CPGCC/UFSC - [S.I.: s.n., 2000]. Disponível em: <<http://www.inf.ufsc.br/~visao/2000/Hough/index.html>>. Acesso em: 13 mar. 2008.
- [23] Blanchet, Gérard; Charbit, Maurice. *Digital Signal and Image Processing using MATLAB*. UK. ISTE. 2006.
- [24] Aggarwal, Nitin. *Line Detection in Images Through Regularized Hough Transform*, Student Member, IEEE, and William Clem Karl, Senior Member, IEEE. Ieee Transactions On Image Processing, VOL. 15, NO. 3, MARCH 2006.

- [25] Deans, Stanley R. *Hough Transform from the Radon Transform*, IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL. PAMI-3, NO. 2, MARCH 1981
- [26] Helgason, Sigurdur. *Radon Transform*. Second Edition. [Downloadable - Hardcover edition also available, published by Birkhauser] Boston. [S.I.: s.n.,1999] 196p. Disponível em: <www.math.mit.edu/~helgason/Radonbook.pdf>. Acesso em: 21 set. 2009.
- [27] Wood, John C. and Daniel T. Barry. *Radon Transformation of Time-Frequency, Distributions for Analysis of Multicomponent Signals. Member, IEEE*. Ieee Transactions On Signal Processing. VOL. 42. NO. 11, NOVEMBER 1994
- [28] Matus, František; Flusser Jan. *Image Representations via a Finite Radon Transform*, Ieee Transactions On Pattern Analysis And Machine Intelligence, VOL. 15, NO. 10, OCTOBER 1993
- [29] Shapiro, Vladimir A; Ivano, Valenth H. *Real-Time Hough/Radon Transform: Algorithm And Architectures*. [ICIP-94. IEEE. Image Processing International Conference](#) Publication. Date: 13-16 Nov 1994. Volume: 03, On page(s): 630-634.
- [30] Coifman, R.R., et al. *A notion of Radon Transform for Data in a Cartesian Grid which is Rapidly Computible, Algebraically Exact, Geometrically Faithful and Invertible*. Averbuch. [S.I.: s.n., 2001?]. [2001?]. 40p. Disponível em: <www.stat.stanford.edu/~donoho/Reports/2001/FastSlantStack.pdf>. Acesso em: 12 nov. 2008.
- [31] Jain. Anil K., *Fundamental of Digital Image processing*, UNIVERSITY OF CALIFORNIA, Prentice Hall, 1989.
- [32] Ginkel, M. van; Hendriks, C.L. Luengo; Vliet, L.J. Van; *A short introduction to the Radon and Hough transforms and how they relate to each other*. Number QI-2004-01. Imaging Science & Technology Department Faculty of Applied Science. Delft University of Technology. 2004.
- [33] Silva, Erivaldo Antônio da; Leonardi, Fabrício; Alberti, Giovana Araújo. *Aplicação de técnicas de Morfologia Matemática e PDI na extração de drenagem em imagens digitais*. FCT/UNESP. Depto de Cartografia, 7º Brazillian Conference on Dynamics, Control na Applications, SP, maio de 2008.
- [34] Simões, Filipe Carmona. *Introdução à Morfologia Matemática com Aplicações em Processamento de Imagens*. XII Congresso Interno de Iniciação Científica da Unicamp. IMECC, UNICAMP. 2004

- [35] Soille, P.; Pesaresi, M. *Advances in Mathematical Morphology Applied to Geoscience and Remote Sensing*. Geoscience and Remote Sensing, IEEE Transactions on Volume 40, Issue 9, Sep 2002 Page(s): 2042 – 2055.
- [36] Guangjian Yan, et al. *Automatic Extraction of Power Lines from Aerial Images*. IEEE Geoscience And Remote Sensing Letters, VOL. 4 NO. 3, july 2007.
- [37] Dell'Acqua, Fabio, *Member, IEEE*; Gamba, Paolo, *Senior Member, IEEE*. *Detection of Urban Structures in SAR Images by Robust Fuzzy Clustering Algorithms: The Example of Street Tracking*. IEEE Transactions On Geoscience And Remote Sensing, VOL. 39, NO.10, OCTOBER 2001.
- [38] Rangel, Rodrigo Kuntz; Kienitz, Karl Heinz; Brandão, Mauricio Pazini. *Sistema de Inspeção de Linhas de Transmissão de Energia Elétrica utilizando veículos aéreos não tripulados*. Brazilian Symposium on Aerospace Eng. & Applications. 2009
- [39] Claro, R. T.; et al. *Desenvolvimento de Sistema de Segurança Patrimonial Aplicado a Linhas de Transmissão de Energia Elétrica*. Disponível em: 23 out. 2009. <http://www.cteep.com.br/port/institucional/artigos/Artigo_0068_001_2006.pdf>. Acesso em: 23 ago. 2009.
- [40] Monteiro, Sergio Paulo Lemos. *Monitoramento de Linhas de Transmissão via satélites de alta resolução*. DEPL. CHESF.[2005?]. Disponível em: <<http://www5.chesf.gov.br/Aplic/pedweb2005.nsf/0/3647def31b0305c503256fcd00713180?OpenDocument>>. Acesso em: 23 ago. 2009.
- [41] Olivieira, Francisco Henrique de; Wosny, Guilherme Clasen; Dal Santo, Mariane Alves. *Análise e monitoramento das Linhas de Transmissão e Faixas de Servidão, utilizando SIG*. UDESC. Disponível em: <www.geolab.faed.udesc.br/publicacoes/.../SIGSUL_2005.pdf>. Acesso em: 23-08-2009.
- [42] Sistema de Cadastro Georreferenciado com Sensoriamento Remoto de Linhas de Transmissão. CPqD. Disponível em: <<http://www.cpqd.com.br/pad-e-inovacao/4211-sistema-de-cadastro-georreferenciado-com-sensoriamento-remoto-de-linhas-de-transmissao.html?tmpl=component&print=1&page=>>>. Acesso em: 23 ago. 2009.
- [43] Katartzis, A; Pizurica, V; Sahli, H. *Application of Mathematical Morphology and Markov Random Field Theory to the Automatic Extraction do Linear Features in Airborne Images*. Proc. of International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing V (ISMM-2000), Palo Alto, California, USA, June 26-28, pp. 405-414, 2000.

- [44] Mohammadzadeh, Ali; Tavakoli, Ahad; Joej, Mohammad J. Valadan. *Road Extraction Based on Fuzzy Logic and Mathematical Morphology from Pan-Sharpended Ikonos Images*. The Photogrammetric Record 21(113): 44-60 (March 2006).
- [45] Thyagarajan, K. S. *Digital Image Processing with Application to Digital Cinema*. UK. Focal Press. 2006.
- [46] Serra, J. *Image Analysis and Mathematical Morphology*. London. Academic Press Inc. (LONDON) LTD. 1982
- [47] Pratt, William K. *Digital image processing: PIKS Scientific inside / William K. Pratt.*—, 4th ed. Los Altos, California. WILEY-INTERSCIENCE. 2007

APÊNDICE A – Algoritmos

A.1 – Escolhendo o detector de Bordas

```

f=imread('imagem.tif');
choice=0;
while (choice~=5)
choice=input('1:  Prewitt\n2:  Sobel\n3:  Laplacian  of  a
Guassian(LoG)\n4:  Canny\n5:  Exit\n Enter your choice : ');
switch choice
    case 1
        PF=edge(f(:,:,1),'prewitt');
        PF=edge(f(:,:,2),'prewitt');
        PF=edge(f(:,:,3),'prewitt');
        figure, imshow(f),title('Original
Image'),figure,imshow(PF),title('Prewitt Filter');
    case 2
        RF=edge(f(:,:,1),'sobel');
        RF=edge(f(:,:,2),'sobel');
        RF=edge(f(:,:,3),'sobel');
        figure, imshow(f),title('Original
Image'),figure,imshow(RF),title('Sobel Filter');
    case 3
        LF=edge(f(:,:,1),'log');
        LF=edge(f(:,:,2),'log');
        LF=edge(f(:,:,3),'log');
        figure, imshow(f),title('Original
Image'),figure,imshow(LF),title('Laplacian of Gaussian (LoG)
Filter');

```

```

case 4
    CF=edge(f(:,:,1),'canny');
    CF=edge(f(:,:,2),'canny');
    CF=edge(f(:,:,3),'canny');
    figure, imshow(f),title('Original
Image'),figure,imshow(CF),title('Canny Filter');
case 5
    display('Program Exited');
otherwise
    display('\nWrong Choice\n');
end
end
end

```

A.2 – Complemento da Imagem

```

% Aplica o complemento da imagem
I=imread('imagem.tif');
IM2 = IMCOMPLEMENT(I);
imshow (IM2);

```

A.3 – Detecção de bordas utilizando filtro Canny

```

I=imread ('imagem.tif');
BW = edge(I(:,:,1),'canny');
BW = edge(I(:,:,2),'canny');
BW = edge(I(:,:,3),'canny');
figure(1), imshow(I)
figure(2), imshow(BW)

```

A.4- Extração de Linhas através da TH utilizando a TR

A extração de linhas foi baseada nos algoritmos abaixo:

1º algoritmo

```
I = imread('imagem.tif');
```

```
%or threshold its edges
```

```
figure(1),imshow(I);
```

```
E = edge(I(:,:,1),'canny');
```

```
E = edge(I(:,:,2),'canny');
```

```
E = edge(I(:,:,3),'canny');
```

```
figure(2),imshow(E);
```

```
title ('Gradient Image')
```

```
imwrite(E,'HoughlineE.jpg','jpg','Quality',70)
```

```
%Uso da função radon para computar a TH:
```

```
theta = (0:179)';
```

```
[R,xp] = radon(E,theta);
```

```
figure(2)
```

```
imagesc(theta,xp,R), colorbar;
```

```
xlabel ('theta (deg)'), ylabel ('rho (pixels from center)')
```

```
title('Line Space');
```

```
plt=1;
```

```
for peaks =50:10:110
```

```
    i = find(R>peaks);
```

```
    [foo,ind] = sort(-R(i));
```

```
    k = i(ind(1:size(i)));
```

```
    [y,x] = ind2sub(size(R),k);
```

```
        t = -theta(x)*pi/180;
```

```
        lines = [cos(t) sin(t) -r];
```



```

cx = size(I,2)/2-1;
cy = size(I,1)/2-1;
lines(:,3) = lines(:,3) - lines(:,1)*cx - lines(:,2)*cy;

figure(3+plt)
text1='Gradient image with lines - Accumulator > ';
text2=num2str(peaks);
text=strcat(text1,text2);
imshow(I); title(text);
draw_lines(lines);
plt=plt+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [XX,YY] = draw_lines(L, X, Y, lt)
if nargin < 4
    lt = '-';
end
if nargin < 3
    if nargin < 2
        lt = '-';
    else
        lt = X;
    end
end
V = axis;
X = V(1:2); Y = V(3:4);

hold on

end

```

```

i = find(abs(L(:,1)) < abs(L(:,2)));
j = find(abs(L(:,1)) >= abs(L(:,2)));

YY1 = -L(i,1)./L(i,2)*X + -L(i,3)./L(i,2)*ones(size(X));
XX1 = ones(length(i),1)*X;

XX2 = -L(j,2)./L(j,1)*Y + -L(j,3)./L(j,1)*ones(size(Y));
YY2 = ones(length(j),1)*Y;

XX = [XX1' XX2'];
YY = [YY1' YY2'];

[foo,k] = sort([i; j]);
plot(XX(:,k),YY(:,k),lt)

if nargin < 3
hold off
end

2° algoritmo
I = imread('image.tif');
I=E
title ('Gradient Image')
imwrite(E,'HoughlineE.jpg','jpg','Quality',70)
theta = (0:179)';
[R, xp] = radon(E,theta);
figure(2)
imagesc(theta,xp,R), colorbar;
xlabel ('theta (deg)'), ylabel ('rho (pixels from center)')
title('Line Space');

plt=1;
for peaks =100:5:145
    i = find(R>peaks);

```

```

[foo,ind] = sort(-R(i));
k = i(ind(1:size(i)));
[y,x] = ind2sub(size(R),k);
t = -theta(x)*pi/180;
r = xp(y);
lines = [cos(t) sin(t) -r];
cx = size(I,2)/2-1;
cy = size(I,1)/2-1;
lines(:,3) = lines(:,3) - lines(:,1)*cx - lines(:,2)*cy;
figure(3+plt)
text1='Gradient image with lines - Accumulator > ';
text2=num2str(peaks);
text=strcat(text1,text2);
imshow(I); title(text);
draw_lines(lines);
plt=plt+1;
end

```

A.5 - Extração de linhas utilizando TH

```

I = imread('imagem.tif');

BW = edge(I(:,:,1),'canny');
BW = edge(I(:,:,2),'canny');
BW = edge(I(:,:,3),'canny');

[H,theta,rho] =
hough(BW,'ThetaResolution',0.45,'RhoResolution',0.7);

imshow(H,[],'XData',theta,'YData',rho,...
'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');

```

```

axis on, axis normal, hold on;
P =
houghpeaks(H,12, 'threshold',ceil(0.755*max(H(:)))) , 'NHoodSize'
,[3 3];

x = theta(P(:,2));
y = rho(P(:,1));
plot(x,y, 's', 'color', 'white');

lines = houghlines(BW,theta,rho,P, 'FillGap',15, 'MinLength',5);

figure, imshow(I), hold on
max_len = 0;
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2), 'LineWidth',2, 'Color', 'green');

    plot(xy(1,1),xy(1,2), 'x', 'LineWidth',2, 'Color', 'yellow');
    plot(xy(2,1),xy(2,2), 'x', 'LineWidth',2, 'Color', 'red');

    len = norm(lines(k).point1 - lines(k).point2);
    if ( len > max_len)
        max_len = len;
        xy_long = xy;
    end
end

plot(xy_long(:,1),xy_long(:,2), 'LineWidth',2, 'Color', 'cyan');

```

A.6- Filtragem Morfológica

1º) A imagem de entrada é a imagem complemento, ver A.2

```
% Primeiro teste
I=imread('comple.jpg');
figure(1),imshow(I);

se = strel('disk',3);
J = imsubtract(imadd(I,imtophat(I,se)), imbothat(I,se));
figure(2), imshow(J)
```

A.7- Detector de Bordas Externas através Morfologia Matemática

→A imagem resultante de A.7, será a imagem de entrada neste algoritmo

```
IG= J(:,:,1);
sq=ones(3,3);
%bordas externas para IG>150
r150=IG>150;
rd150=imdilate(r150,sq);
r_ext150=rd150&~r150;
figure(1),imshow (r_ext150)
%bordas externas para IG>210
r210=IG>210;
rd210=imdilate(r210,sq);
r_ext210=rd210&~r210;
figure(2),imshow (r_ext210)
B= r210-[(r210 - r_ext150)+ (r210 - r_ext210)];figure(3),
imshow(B);
```

A.8- SUSAN algoritmo

O algoritmo utilizado nesta dissertação foi uma adaptação do algoritmo descrito abaixo:

```
% =====
% File name      : edgexample.m
% File Type     : m-file (script file for Matlab)
% Requirements: Matlab Image Processing Toolbox
% Begin        : 2006-07-07
% Last Update  : 2007-03-22
% Author       : Nicola Asuni
% Description   : example script for edgemap.m and edgedir.m
functions.
% Copyright    : Tecnick.com S.r.l.
%              : Via Ugo Foscolo n.19
%              : 09045 Quartu Sant'Elena (CA)
%              : ITALY
%              : http://www.tecnick.com
%              : info@tecnick.com
% License      : GNU GENERAL PUBLIC LICENSE v.2
%              : http://www.gnu.org/copyleft/gpl.html
% =====
```

DESCRIÇÃO

```
function [] = edgexample()
% =====
% File name      : edgexample.m
% File Type     : m-file (script file for Matlab)
% Requirements: Matlab Image Processing Toolbox
% Begin        : 2006-07-07
% Last Update  : 2007-03-22
% Author       : Nicola Asuni
```

```

% Description : example script for edgemap.m and edgedir.m
functions.
% Copyright   : Tecnick.com S.r.l.
%             Via Ugo Foscolo n.19
%             09045 Quartu Sant'Elena (CA)
%             ITALY
%             http://www.tecnick.com
%             info@tecnick.com
% License     : GNU GENERAL PUBLIC LICENSE v.2
%             http://www.gnu.org/copyleft/gpl.html
% =====
% --- parâmetros de entrada ---
TR = 3;
KR = 9;
NR = 255;
OP = false;
IMG = imread('IMAGEM');

figure(1), imshow(IMG);
t1 = cputime;
EDG = edgemap(IMG, TR, KR, NR, OP);

fprintf('\nProcessing time: %8.3f sec\n', cputime-t1);

figure(2), imshow(EDG);
figure(3),
surface(EDG, 'FaceColor', 'texturemap', 'EdgeColor', 'none', 'CData
Mapping', 'scaled');
view(135,80);

ANG = edgedir(EDG, KR, OP);
fprintf('\nProcessing time: %8.3f sec\n', cputime-t1);

```

```

    figure(4), imshow(ANG);
    figure(5),
surface(ANG, 'FaceColor', 'texturemap', 'EdgeColor', 'none', 'CData
Mapping', 'scaled');
    view(135,80)

% === EOF =====

function [EDG] = edgemap(IM, TR, KR, NR, OP)
% =====
% File name      : edgemap.m
% File Type      : m-file (script file for Matlab)
% Requirements:  Matlab Image Processing Toolbox
% Begin         : 2006-07-07
% Last Update   : 2007-05-31
% Author        : Nicola Asuni
% Description    : This function detects image edges using an
improved
%                SUSAN technique.
% Copyright     : Tecnick.com S.r.l.
%                Via Ugo Foscolo n.19
%                09045 Quartu Sant'Elena (CA)
%                ITALY
%                http://www.tecnick.com
%                info@tecnick.com
% License       : GNU GENERAL PUBLIC LICENSE v.2
%                http://www.gnu.org/copyleft/gpl.html
% Version       : 1.1.000

%
=====

```



```

    if (nargin < 1)
disp('edgemap function by Nicola Asuni.');
```

disp('This function returns an edge map.');

disp('Usage:');

disp('[EDG] = edgemap(IM)');

disp('[EDG] = edgemap(IM, TR)');

disp('[EDG] = edgemap(IM, TR, KR)');

disp('[EDG] = edgemap(IM, TR, KR, NR)');

disp('[EDG] = edgemap(IM, TR, KR, NR, OP)');

disp('Where:');

disp('IM : source image (RGB or grayscale)');

disp('TR : Brightness Threshold (default = 20)');

disp('KR : USAN Kernel Radius (nucleus excluded) (default = 3)');

disp('NR : the EDG matrix will be normalized to this range of integers (default = 0 = not normalize)');

disp('OP : if true removes from USAN the pixels that are not directly connected with the nucleus (default = false)');

disp('EDG : edge strength image');

EDG = [];

return;

end

% assign default values

if (nargin > 5)

error('Too many arguments');

end

if (nargin < 5)

OP = false;

end

if (nargin < 4)

NR = 255;

```

end
if (nargin < 3)
    KR = 9;
end
if (nargin < 2)
    TR = 5;
end

% -----
N_COLORS = ndims(IM);
if (N_COLORS == 3)
% RGB image
    IMG = double(rgb2gray(IM));
elseif (N_COLORS == 2)
    IMG = double(IM);
else
    error('Unrecognized image type, please use RGB or greyscale
images');
end
IMG = IMG + 255;

[m,n] = size(IMG);

KW = (2 * KR) + 1;

KM = ones(KW,KW);
for i = -KR:KR
    for j = -KR:KR
        if (round(sqrt((i.^2) + (j.^2))) > KR)
            KM(i+KR+1,j+KR+1) = 0;
        end
    end
end
end

```

```

end

KAREA = nnz(KM);
GT = 3 * KAREA / 4;

IMG = [ repmat(IMG(1,:),KR,1);IMG;repmat(IMG(m,:),KR,1) ];
IMG = [ repmat(IMG(:,1),1,KR),IMG,repmat(IMG(:,n),1,KR) ];
EDG = zeros(m,n);

for i = KR+1:m+KR
    for j = KR+1:n+KR
        USAN = KM .* exp(-((IMG(i-KR:i+KR, j-KR:j+KR) - IMG(i,j))
/ TR).^6);

        if (OP)

            USAN_BINARY = ceil(USAN);
            USAN = bwselect(USAN_BINARY,KR+1,KR+1,8) .* USAN;
        end
    end

% USAN area
    USAN_AREA = sum(sum(USAN));

    if (USAN_AREA < GT)
        EDG(i-KR,j-KR) = GT - USAN_AREA;
    end
end
end

if (NR > 0)
    NSCALE = NR / max(max(EDG));

```

```

EDG = round(NSCALE .* EDG);
if (NR > (2^32))
    EDG = uint64(EDG);
elseif (NR > (2^16))
    EDG = uint32(EDG);
elseif (NR > (2^8))
    EDG = uint16(EDG);
else
    EDG = uint8(EDG);
end
end

% === EOF

function [ANG] = edgedir(EDG, KR, OP)
% =====
% File name      : edgedir.m
% File Type     : m-file (script file for Matlab)
% Requirements:  Matlab Image Processing Toolbox
% Begin        : 2006-07-07
% Last Update  : 2007-05-31
% Author       : Nicola Asuni
% Description  : This function returns the orientation of image
edges
%              using a modified SUSAN technique.
% Copyright   : Tecnick.com S.r.l.
%              Via Ugo Foscolo n.19
%              09045 Quartu Sant'Elena (CA)
%              ITALY
%              http://www.tecnick.com
%              info@tecnick.com
% License     : GNU GENERAL PUBLIC LICENSE v.2
%              http://www.gnu.org/copyleft/gpl.html

```

```
% Version      : 1.1.000

if (nargin < 1)
    disp('edgedir function by Nicola Asuni.');
```

disp('This function detects edges orientation using the SUSAN criteria.');

```
disp('Usage:');
disp('[ANG] = edgedir(EDG)');
disp('[ANG] = edgedir(EDG, KR)');
disp('[ANG] = edgedir(EDG, KR, OP)');
disp('Where:');
disp('EDG : edge strength image');
disp('KR  : USAN Kernel Radius (nucleus excluded) (default
= 3)');
disp('OP  : if true removes from USAN the pixels that are
not directly connected with the nucleus (default = false)');
disp('ANG : edge directions in radians normalized in the
interval ]0,pi]');
```

```
ANG = [];
return;
end

if (nargin > 3)
    error('Too many arguments');
end

if (nargin < 3)
    OP = false;
end

if (nargin < 2)
    KR = 3;
end
```

```

NCOLORS = length(size(EDG));
if (NCOLORS == 3)
    EDG = rgb2gray(EDG);
end
MAXVAL = double(max(max(EDG)));
if (MAXVAL > 1)
    EDG = double(EDG) / MAXVAL;
end
EDG = EDG + 255;
[m,n] = size(EDG);
KW = (2*KR)+1; % kernel width
KM = ones(KW,KW);
for i = -KR:KR
    for j = -KR:KR
        if (round(sqrt((i.^2) + (j.^2))) > KR)
            KM(i+KR+1,j+KR+1) = 0;
        end
    end
end

KAREA = nnz(KM);
EDG' =
[zeros(KR,n+(2*KR));zeros(m,KR),EDG,zeros(m,KR);zeros(KR,n+(2*
KR))];
ANG = zeros(m,n);
D = -KR:KR;
RX = repmat(D,(2*KR)+1,1);
RY = RX';
DSQX = RX .^ 2;
DSQY = DSQX';
for i = KR+1:m+KR
    for j = KR+1:n+KR

```

```

if (EDG(i,j) > 0)
    USAN = KM .* double(EDG(i-KR:i+KR,j-KR:j+KR));
    if (OP)
        USAN_BINARY = ceil(USAN);
        if (nnz(USAN_BINARY) < (KAREA - KR))
            USAN = bwselect(USAN_BINARY,KR+1,KR+1,8) .* USAN;
        end
    end
    USAN_AREA = sum(sum(USAN));
    CDY = -(double(sum(sum(RY .* USAN))) / USAN_AREA);
    CDX = (double(sum(sum(RX .* USAN))) / USAN_AREA);

    if (sqrt((abs(CDX).^2) + (abs(CDY).^2)) < 1)
        DX = sum(sum(DSQX .* USAN));
        if (DX > 0)
            DY = - sum(sum(DSQY .* USAN));
            DXY = sum(sum(RX .* RY .* USAN));
            EANG = sign(DXY) * atan(DY/DX);
        else
            EANG = pi/2;
        end
    else
        if (abs(CDX) > 0)
            EANG = (pi/2) + atan(CDY/CDX);
        else
            EANG = pi;
        end
    end
    EANG = normalizeangle(EANG);
    ANG(i-KR,j-KR) = EANG;
end

```

```
    end  
  end  
% === EOF
```


APÊNDICE B - Revisão de Teoria dos Conjuntos

Teoria dos Conjuntos

Conjunto como próprio nome diz é uma coleção de objetos sem repetição. Se um conjunto não é muito grande, este pode ser descrito listando-se os seus elementos em qualquer ordem, por exemplo:

$$A = \{1, 2, 3, 4\}$$

Se um conjunto é muito grande ou infinito, este pode ser descrito através de propriedades requeridas de seus membros:

$$X = \{x \mid x \text{ é um } n^{\circ} \text{ inteiro, positivo e par}\}$$

$x \in X$, se x está em X , caso contrário $x \notin X$.

Combinação de Conjuntos

Sejam dois conjuntos X e Y :

UNIÃO

A união de dois conjuntos X e Y é o conjunto de elementos que pertencem ou X ou Y ou ambos:

$$X \cup Y = \{x \mid (x \in X) \text{ ou } (x \in Y)\}$$

INTERSEÇÃO

A interseção de dois conjuntos X e Y é o conjunto de elementos pertencentes a ambos X e Y :

$$X \cap Y = \{x \mid (x \in X) \text{ e } (x \in Y)\}$$

DIFERENÇA

A diferença entre os conjuntos X e Y é o conjunto de elementos que pertencem a X , mas não pertencem a Y :

$$X - Y = \{x \mid (x \in X) \text{ e } (x \notin Y)\}$$

COMPLEMENTO

O complemento do conjunto X é o conjunto dos elementos não pertencentes ao conjunto X . Se nos pensarmos em imagem este poderia corresponder aos pixels brancos na figura ou na linguagem da teoria dos conjuntos:

DISJUNÇÃO

São disjuntos quando não existem elementos comuns entre eles.

$X \cap Y = \emptyset$ então X e Y são disjuntos.

TRANSLAÇÃO

A translação do conjunto X pelo ponto a é definida, em notação de conjuntos, como:

$$X' = \{y \mid y = a + x, x \in X\}$$

REFLEXÃO

A reflexão do conjunto X é definida como:

$$X' = \{y \mid y = -x, x \in X\}$$