UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

REVISTA UD Y LA GEOMÁTICA

http://revistas.udistrital.edu.co/ojs/index.php/UDGeo/index
DOI:http://dx.doi.org/10.14483/udistrital.jour.udgeo.2015.10.a06

UD y la GEOMÁTICA

INVESTIGACIÓN

Resultados da implementação de detectores de bordas no software CARTOMORPH

Implementation of edge detectors in CARTOMORPH software

Erivaldo Antônio da Silva[1], Amanda Fructuozo dos Santos[2], Carolina Dias Chaves[3]

Para citar este artículo: da Silva, E.A., dos Santos, A.F. & Dias-Chaves C. (2015). Resultados da implementação de detectores de bordas no software CARTOMORPH. *UD y la Geomática*, 10, 45-52

Fecha de recepción: 22 de abril de 2015 **Fecha de aceptación:** 02 de diciembre de 2015

RESUMO:

A utilização da Morfologia Matemática na área de Processamento Digital de Imagem é apresentada como uma abordagem não linear de análise de estruturas espaciais resultando, na maioria das vezes em resultados melhores que os obtidos com a abordagem linear para extração de estruturas geométrica de objetos. Na FCT/UNESP está sendo desenvolvido o software de domínio público, CARTOMORPH, em que foram implementados detectores de bordas morfológicos e convencionais. O operador Gblur foi implementado no software utilizando detectores morfológicos, já os operadores Sobel e Prewitt foram implementados na forma convencional, utilizando filtros espaciais como base em suas aplicações. De acordo com a análise dos resultados obtidos depois da aplicação dos detectores de borda, foi concluído que o detector Gblur possuiu o melhor resultado, confirmando a hipótese inicial de que operadores não lineares são mais adequados em relação aos operadores lineares.

Palavras-chaves: Detector de Borda, Morfologia Matemática, PDI, Sensoriamento Remoto.

ABSTRACT:

Using Mathematical Morphology in the area of Digital Image Processing is presented as a non-linear approach to analyzing spatial structures, giving better results than the linear approach to extracting geometric structures of objects. In FCT / UNESP the software is being developed for public CARTOMORPH field, which is the implementation of morphological and conventional edge detectors. The Gblur operator was implemented in software using morphological detectors, since the Sobel and Prewitt operators were implemented in the conventional manner using spatial filters based on their applications. According to the analysis of the results obtained after the application of edge detectors, it was concluded that the detector Gblur possessed the best result, confirming the initial hypothesis that nonlinear operators are more appropriate in relation to linear operators.

Keywords: Edge Detector, Mathematical Morphology, PDI , Remote Sensing .

1 Faculdade de Ciências e Tecnologia – FCT/UNESP – Depto de Cartografia. Rua Roberto Simonsen, 305 – Campus Universitário–19060-900 – Presidente Prudente-SP. silva.erivaldo@gmail.com

2 Faculdade de Ciências e Tecnologia – FCT/UNESP – Depto de Cartografia. Rua Roberto Simonsen, 305 – Campus Universitário–19060-900 – Presidente Prudente-SP. amandafructuozodossantos@gmail.com

3 Faculdade de Ciências e Tecnologia – FCT/UNESP – Depto de Cartografia. Rua Roberto Simonsen, 305 – Campus Universitário–19060-900 – Presidente Prudente-SP.carolina.dias.chaves@gmail.com

Introdução

O Sensoriamento Remoto utiliza o Processamento Digital de Imagens (PDI), pois envolve a manipulação e interpretação de imagens digitais. Dentre as diversas técnicas de PDI, a escolhida para a elaboração deste trabalho foi a Morfologia Matemática (MM). Esta pode ser definida como uma análise de estruturas espaciais (Soille, 1999), fundamentada a partir de uma malha perfeitamente definida e conhecida, chamada elemento estruturante. Foi utilizada para melhor tratamento das imagens e detecção das bordas, assim foi possível obter os resultados esperados. A detecção de bordas pode ser definida como uma operação de identificação de mudanças locais na imagem e também uma das ferramentas de PDI. A aplicação da Morfologia Matemática no processo de detecção de bordas para a ciência de Sensoriamento Remoto é de grande ajuda, pois apresenta uma solução para a desatualização de produtos cartográficos. Neste trabalho foram utilizados os seguintes detectores de bordas: Gblur, Sobel e Prewitt.

Metodologia

A metodologia do trabalho apresenta a fundamentação teórica utilizada para a realização do mesmo, a implementação dos detectores de bordas estudados, os resultados, a análise e a conclusão do trabalho.

Fundamentação Teórica

O presente trabalho foi realizado com base nas fundamentações teóricas de Morfologia Matemática; Processamento Digital de Imagens e Detectores de Bordas. Cada uma delas são apresentadas como subseções à seguir.

O estudo das respectivas áreas foi de grande importância para a implementação dos detectores de bordas e comparação de todos para encontrar o que melhor se encaixa na extração de rodovias.

Morfologia Matemática

A Morfologia Matemática teve origem no período de 1964 e 1968 na França, juntamente na época em que foi criado o Centro de Morfologia Matemática na Escola de Minas de Paris localizada em Fontainebleau (França).

Segundo Soille(1999) a Morfologia Matemática (MM) pode ser definida como uma teoria para a análise de estruturas espaciais. É chamada de morfologia porque visa analisar a forma dos objetos. É matemática no sentido que a análise está baseada em teoria ajustada, na geometria e na álgebra. Porém, a MM não é apenas uma teoria, mas é também uma poderosa técnica de análise de imagens.

O método de análise de imagens por MM é fundamentado na teoria de conjuntos e tem como objetivo principal analisar a estrutura geométrica das imagens a partir de uma malha retangular perfeitamente definida e conhecida, chamada de elemento estruturante. Sendo assim, a Morfologia Matemática age sobre as imagens digitais a partir desses elementos.

A Morfologia Matemática é constituída a partir de dois operadores básicos: erosão e dilatação.

A erosão de um conjunto X por um elemento estruturante B é denotada por $\varepsilon_B(X)$ e é definido como a posição dos pontos, x, tal que B está incluído em X quando sua origem está localizada em x (Soille, 1999):

$$\varepsilon_B(X) = \{x \mid B_x \subseteq X\} \quad (1)$$

O elemento estruturante Bx está relacionado ao elemento estruturante B centrado no pixel x. De acordo com a equação (1), o elemento estruturante B passa pela imagem X, comparando cada pixel com a vizinhança de x. São mantidos os pixels onde as vizinhanças coincidem, ou seja, se o pixel de B corresponde a mesma posição na vizinhança de x.

A Figura 1 apresenta um exemplo da aplicação do operador morfológico erosão.

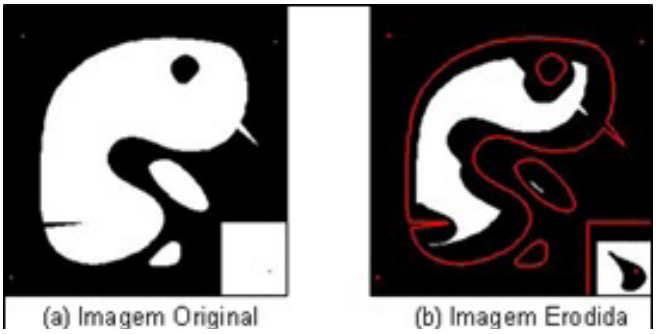


Figura 1–Aplicação do operador erosão.

Fonte: SDC Information Systems.

A dilatação de um conjunto X pelo elemento estruturante B é denotada por $\delta_B(X)$ e é definida como a posição dos pontos x tal que B toca X quando sua origem coincide com x (Soille, 1999):

$$\delta_B(X) = \{x \mid B_x \cap X \neq \varnothing\} \quad (2)$$

A Figura 2 apresenta o resultado da aplicação do operador morfológico dilatação.

Os operadores de erosão e dilatação são chamados de operadores elementares de Morfologia Matemática (SDC Information Systems, 2000), pois são fundamentais para a construção de outros operadores morfológicos.

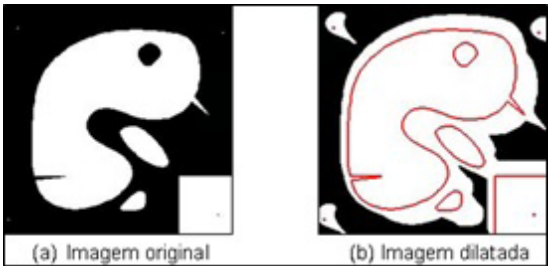


Figura 2–Aplicação do operador erosão.

Fonte: SDC Information Systems.

Processamento Digital de Imagem

Gonzalez define a palavra "imagem" como uma função bidirecional, *f* (*x,y*), em que *x* e *y* são coordenadas *espaciais* (plano), e a amplitude de *f* em qualquer par de coordenadas é chamada *intensidade* ou *nível de cinza* do ponto da imagem.

O ato de fornecer uma imagem de entrada para o computador e o mesmo te retorna uma imagem que foi gerada à partir da original, é chamado de Processamento Digital de Imagem (PDI). Existem três tipos de processamento, baixo nível, médio nível e alto nível.

Para a redução de ruídos da imagem, realce do contraste e afinamento da imagem é utilizado o processamento de baixo nível. Caso o objetivo seja detectar bordas, contorno ou objetos individuais o mais indicado é o de médio nível. O processamento de alto nível é usado para conectar conjuntos de objetos reconhecidos.

Como o objetivo principal do trabalho foi encontrar o melhor detector de borda, o processamento que melhor se encaixa na implementação é o de médio nível.

Detectores de Borda

De acordo com Gonzalez (1993), as bordas dos elementos presentes em uma imagem são fundamentais no processo de análise de imagens. Isto ocorre porque as bordas definem o contorno dos objetos presentes na imagem.

Durante o desenvolvimento do trabalho foram estudados e implementados os detectores de bordas convencionais Sobel e Prewitt e o detector Gblur.

Detector Sobel

O detector de borda Sobel apresenta uma característica marcante que é a ponderação dos *pixels*. Nele quanto mais próximo do *pixel* central, maior o valor do peso atribuído ao ponto. Por esse motivo, a máscara de *Sobel* obtém as bordas mais destacadas em relação aos demais.

Na prática o operador é capaz de calcular o gradiente de intensidade da matriz (Figura 3) de cada campo, com direção em função da distância e quantidade, permitindo que haja uma interpretação da variação ponto a ponto, se ela é suave ou abrupta, por exemplo.

Sobel	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

Figura 3–Operador Sobel.

Fonte: Adaptado de Maturana (2010)

Detector Prewitt

O operador *Prewitt* é capaz de diferenciar as direções vertical e horizontal. Ele encontra as bordas utilizando uma aproximação da derivada, retornando bordas que possuem gradiente máximo.

As matrizes aplicadas no processamento que usa o operador *Prewitt* são:

Prewitt	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Figura 4–Operador Prewitt.

Fonte: Adaptado de Maturana (2010)

Detector Gblur

É um detector de borda que opera sobre a imagem original borrada, transformando uma borda ideal numa rampa (Silva, 1989).

Na aplicação do operador Gblur é utilizada uma máscara (Figura 5) para borrar a imagem original, esta possui dimensão 3x3, resultando na imagem borrada.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1. & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figura 5–Máscara do detector Gblur.

O processamento uma imagem e a extração de sua borda pode ser realizada encontrando o valor mínimo das diferenças da imagem original e da erodida, com a dilatação

menos a original. Sendo assim chamado processamento utilizando operador Gblur.

Implementação MatLab

O algoritmo empregado para os detectores Sobel e Prewitt foi o mesmo, o que os diferencia são os elementos estruturantes utilizados. Abaixo encontra-se o algoritmo.

```
for i = 2 : size(B,1)–1
for j = 2 : size(B,2)–1

GX(i,j) = (abs(hor(1,1) * double(B(i-1,j-1)) + hor(1,2) * double(B(i-1,j)) + hor(1,3) * double(B(i-1,j+1)) + hor(2,1) * double(B(i,j-1)) + hor(2,2) * double(B(i,j)) + hor(2,3) * double(B(i,j+1)) + hor(3,1) * double(-B(i+1,j-1)) + hor(3,2) * double(B(i+1,j)) + hor(3,3) * double(B(i+1,j+1))));
GY(i,j) = (abs(ver(1,1) * double(B(i-1,j-1)) + ver(1,2) * double(B(i-1,j)) + ver(1,3) * double(B(i-1,j+1)) + ver(2,1) * double(B(i,j-1)) + ver(2,2) * double(B(i,j)) + ver(2,3) * double(B(i,j+1)) + ver(3,1) * double(-B(i+1,j-1)) + ver(3,2) * double(B(i+1,j)) + ver(3,3) * double(B(i+1,j+1))));

G(i,j) = (abs((double(GX(i,j)) + (double(GY(i,j))))));
if(G(i,j)>254)
H(i,j) = (1);
else
H(i,j) = (0);
end
end
```

Onde B é a imagem original em tons de cinza, GX e GY são os gradientes horizontal e vertical respectivamente, hor e ver são os elementos estruturantes do detector em questão. G(i,j) tem a soma dos gradientes e se for maior que 254 (é borda), o pixel H(i,j) irá receber o valor 1 (branco). A imagem H contém o resultado da aplicação do detector, ou seja, a borda.

Detector Gblur

- Para o detector Gblur um bom desempenho teórico proposto por Lee et al (1987) consiste na seguinte expressão:

$$\text{Gblur} = \min\{I_1 - \textit{erosão}\ (\mathrm{I}_1),\ \textit{dilatação}\ (\mathrm{I}_1) - I_1\}$$

No caso I_1 representa a imagem de entrada borrada.

Detector Sobel

Os elementos estruturantes hor e ver, criados para o detector Sobel serão apresentados a seguir.

$$\text{hor} = [\text{-1 -2 -1;}\ 0\ 0\ 0;\ 1\ 2\ 1]$$
$$\text{ver} = [\text{-1 0 1;}\ \text{-2 0 2;}\ \text{-1 0 1}]$$

Detector Prewitt

Os elementos estruturantes hor e ver, criados para o detector Prewitt serão apresentados a seguir.

$$\text{hor} = [1\ 0\ \text{-1;}\ 1\ 0\ \text{-1;}\ 1\ 0\ \text{-1}]$$
$$\text{ver} = [\text{-1 -1 -1;}\ 0\ 0\ 0;\ 1\ 1\ 1]$$

Implementação CARTOMORPH

Detector Gblur

O detector pode ser definido como o resultado da imagem com os valores mínimos entre a imagem dilatada e sua a erodida definido da seguinte forma:

```
cmImage * cartomorph::cmGBlur(cmImage * img, cmStructureElement * se) {
    if (img == NULL) {
        cout << endl << "CARTOMORPH ERROR na funcao cmGblur: nao pode ser executada pois a imagem é nula.";
        return NULL;
    }
    if (se == NULL) {
        cout << endl << "CARTOMORPH ERROR na funcao cmGblur: nao pode ser executada pois o elemento estruturante é nulo.";
        return NULL;
    }

    cmImage * imgBlur = cartomorph::cmFilterAVG(img, se);
    return cmMin(cartomorph::cmSubtractImages(imgBlur, cartomorph::cmErode(imgBlur, se)),
            cartomorph::cmSubtractImages(cartomorph::cmDilate(imgBlur, se), imgBlur));
```

Figura 6 – Implementação Detector Gblur.

Detector Sobel

O detector tem como característica a ponderação dos pixels, assim quanto mais próximo do *pixel* central, maior valor do peso para o ponto. Portanto a máscara de *Sobel* obtém bordas mais destacadas em relação aos demais, é codificado da seguinte forma:

```
for (j = 0; j < img->cmGetWidth(); j++) {
    Gx = (-1 * img->cmGetRColor(i-1, j-1) + 1 *  img->cmGetRColor(i+1, j-1)+
          -2 * img->cmGetRColor(i-1, j)   + 2 *  img->cmGetRColor(i+1, j)  +
          -1 * img->cmGetRColor(i-1, j+1) + 1 *  img->cmGetRColor(i+1, j+1))/4 ;
    Gy = (-1 * img->cmGetRColor(i-1, j-1) -2 * img->cmGetRColor(i, j-1)
          -1 * img->cmGetRColor(i+1, j-1) +1 * img->cmGetRColor(i-1, j+1)
          +2 * img->cmGetRColor(i, j+1)   +1 * img->cmGetRColor(i+1, j+1))/4;
    imgResult->cmSetRColor(i,j,sqrt(Gx*Gx + Gy*Gy));
           if ( (Gx ==0) && (Gy != 0))
                {
                  if (Gy > 0)  Dir[i][j] = 90; else Dir[i][j] = 270;
           }
           if ( (Gx !=0) && (Gy == 0))
                {
                  if (Gx > 0)  Dir[i][j] =  0; else Dir[i][j] = 180;
           }
           if ( (Gx !=0) && (Gy != 0))
                {
                  Dir[i][j] = (int)(atan2(Gy,Gx)*180/3.1415);
                  if (Dir[i][j] < 0) Dir[i][j] =  Dir[i][j] + 360;
           }
                Dir[i][j]+=90;
                if (Dir[i][j] > 360)    Dir[i][j]-=180;
           if ( (Gx ==0) && (Gy == 0))   Dir[i][j] = -1;
}

return imgResult;
```

Figura 7 – Implementação Detector Sobel.

A codificação do detector de borda morfológico *Sobel* foi desenvolvida de maneira que, os *pixels* da imagem fossem multiplicados pela matriz da Figura 3–Operadores utilizados na estimação da amplitude do gradiente através de uma borda. E então conforme os gradientes *Gx* e *Gy*, a matriz *Dir* terá o valor do cálculo das direções em relação à intensidade da matriz em cada ponto, em função da distância e quantidade. Resultando então na imagem com o detector aplicado.

Detector Prewitt

O detector tem como característica diferenciar das direções vertical e horizontal da imagem, assim as bordas são encontras por uma aproximação da derivada. A máscara *Prewitt* é codificada da seguinte forma:

```
for (i = 1; i < height-1; i++) {

    for ( j = 1; j < width-1; j++) {

       Px = ( (img->cmGetRColor(i-1, j+1) + img->cmGetRColor(i, j+1) + img->cmGetRColor(i+1, j+1))
            - (img->cmGetRColor(i-1, j-1) + img->cmGetRColor(i, j-1) + img->cmGetRColor(i+1, j-1)) )/3 ;
       Py = ( ( img->cmGetRColor(i-1, j-1) + img->cmGetRColor(i-1, j) + img->cmGetRColor(i-1, j+1))
            - ( img->cmGetRColor(i+1, j-1) + img->cmGetRColor(i+1, j) + img->cmGetRColor(i+1, j+1)) )/3;

       imgResult->cmSetRColor(i,j,sqrt(Px*Px + Py*Py));


    }
}
return imgResult;
```

Figura 8 – Implementação Detector Prewitt.

Para o desenvolvimento do algoritmo foi utilizada a representação de uma imagem pela função *b(i,j)*, assim o operador realizará o calculo do *pixel* atual que será atualizado por meio das variáveis *Pi* e *Pj*.

As variáveis *Pi* e *Pj* são calculadas com base na matriz da figura 4, as equações matemáticas utilizadas forma:

$$Pi = [b(i\text{ -1 , }j\text{+1}) + b(i\text{ , }j\text{+1}) + b(i\text{ + 1 , }j\text{ + 1})] - [b(i\text{ -1 , }j\text{–1}) + b(i\text{ , }j\text{–1}) + b(i\text{ +1 , }j\text{ -1})] \quad (1)$$

$$Pj = [b(i\text{–1 , }j\text{ -1}) + b(i\text{ -1 , }j) + b(i\text{ -1 , }j\text{ + 1})] - [b(i\text{ +1 , }j\text{ -1 }) + b(i\text{ +1 , }j\text{ }) + b(i\text{ + 1 , }j\text{ +1})] \quad (2)$$

E o pixel atualizada será a raiz quadrada da soma dos quadrados das variáveis *Pi* e *Pj*, obtido por meio da equação (3).

$$P(\ i\ ,\ j\ ) = \sqrt{(Pi^2 + Pj^2)} \quad (3)$$

Materiais

MATLAB

O software MATLAB é um ambiente de fácil manuseio e interpretação e possui uma linguagem de programação própria (Linguagem MATLAB) para a realização de cálculos matemáticos. Em um ambiente simples, atende a muitos requisitos no processo de tratamento de imagens digitais e na detecção de feições.

NetBeans

A plataforma NeatBeans é uma gabe para aplicações desenvolvidas em diversas linguagens, uma delas é a C++, utilizada no desenvolvimento da biblioteca CARTOMORPH.

Linguagem C++

"C é uma linguagem de programação de finalidade geral que permite economia de expressão, modernos fluxos de controle e estruturas de dados e um rico conjunto de operadores. Sua falta de restrições e sua generalidade tornam-na mais conveniente e eficaz para muitas tarefas do que linguagens supostamente mais poderosas." (Ritchie e Kernighan, 1989)

A linguagem C é considerada de médio nível, pois possibilita o acesso e a manipulação direta da memória. Tal fator contribui diretamente para um alto desempenho dos programas desenvolvidos. Sabendo que aplicações de processamento digital de imagens necessitam de alto desempenho, a linguagem C foi a escolhida para o desenvolvimento da biblioteca de morfologia matemática com o intuito de facilitar a extração de feições de interesse presentes na imagem digital.

Resultados

Nesta seção estão apresentados os resultados obtidos aplicando os detectores de bordas implementados nos dois ambientes, MATLAB e CARTOMORPH.

Na Figura 9 está representado um trecho do Rodoanel Mario Covas – SP, imagem do satélite QuickBird de 2010, que foi utilizada para os testes dos detectores de bordas.

Figura 9–Trecho Rodoanel Mario Covas – SP.

A Figura 10 apresenta o resultado obtido da aplicação do detector de bordas Sobel na imagem original.

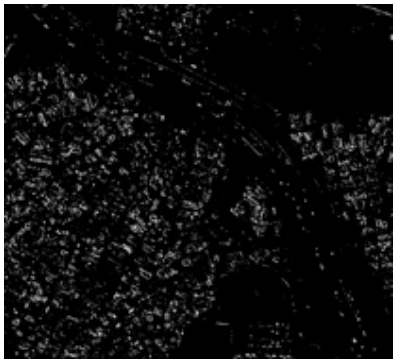


Figura 10–Aplicação Sobel no MATLAB.

Na Figura 11 contém o resultado da aplicação do detector de bordas Prewitt na imagem original.

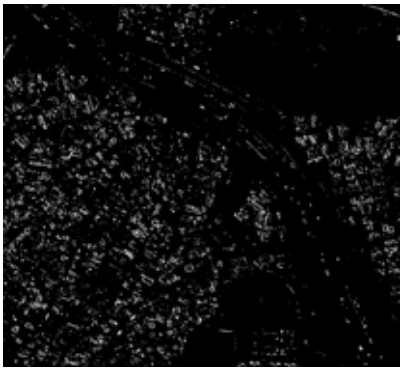


Figura 11–Aplicação Prewitt no MATLAB.

Na Figura 12 tem-se o resultado da aplicação do detector de bordas Gblur na imagem original.

Figura 12–Aplicação Gblur no MATLAB.

A Figura 13 apresenta o resultado obtido da aplicação do detector de bordas Sobel na imagem original.

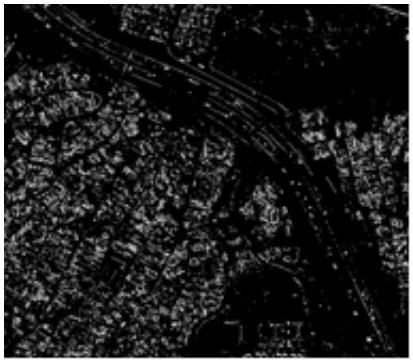


Figura 13–Aplicação Sobel no CARTOMORPH.

Na Figura 14 visualiza-se o resultado da aplicação do detector de bordas Prewitt na imagem original.

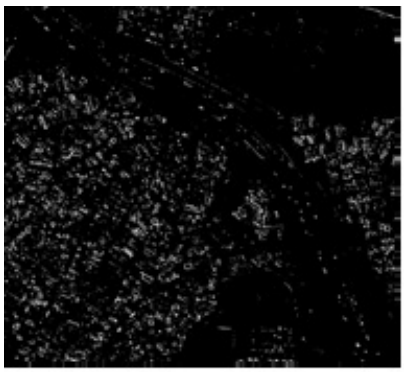


Figura 14–Aplicação Prewitt no MATLAB.

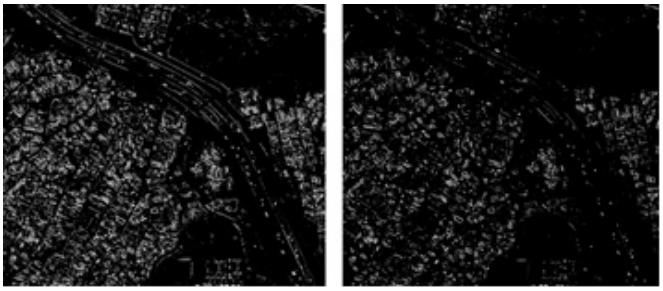
Na Figura 15 está ilustrado o resultado da aplicação do detector de bordas Gblur na imagem original.

Figura 15–Aplicação Gblur no CARTOMORPH.

Discussão de resultados

No decorrer do trabalho, depois dos estudos e desenvolvimentos e/ou implementação dos algoritmos mencionados, concluiu-se a partir da análise dos resultados obtidos com a aplicação dos operadores, que o operador de Sobel apresentou melhor resultado do que o operador Prewitt. Na sequencia o resultado obtido pelo operador de Sobel foi comparado com o obtido pelo operador Gblur. Desta comparação conclui-se que o melhor resultado obtido foi com a aplicação do operador Gblur.

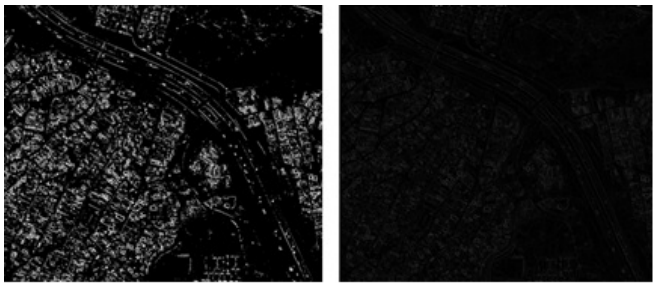
As imagens resultantes das aplicações dos operadors Sobel e Prewitt estão apresentadas na Figura 16.



(a) Sobel (b) Prewitt

Figura 16 (a,b)–Comparação Sobel e Prewitt.

Observa-se, visualmente, que o resultado de Sobel apresentou melhor resultado. A partir disso, o resultado obtido com o operador de Sobel foi comparado com o resultado do Gblur e o resultado está apresentado na Figura 17.



(a) Sobel (b) Gblur

Figura 17 (a,b)–Comparação Sobel e Gblur.

Pode-se verificar, visualmente, que a borda detectada pelo operador Gblur é melhor que a detectado pelo Sobel.

A partir da análise das comparações realizadas anteriormente, podemos concluir que dentre os detectores convencionais (Sobel e Prewitt), o Sobel obteve melhor resultado. Depois de comparar o Sobel com o Gblur, percebe-se que o detector que utiliza a Morfologia Matemática apresentou melhor desempenho, chegando ao resultado esperado, ou seja, o GBlur detectou de forma mais adequada a rodovia que era a feição de interesse no presente trabalho. Desta forma, comprova-se o potencial de uso da MM e do software de domínio público CARTOMORPH, para a aplicação estudada.

Conclusão

O trabalho apresentado alcançou o objetivo esperado no que tange a implementação de detectores de borda convencionais e usando morfologia matemática no *software* CARTOMORPH que está sendo desenvolvido pelo grupo da FCT/UNESP. Tal software, será de domínio público para aplicações específicas na área de Cartografia. No trabalho foram implementados os detectores de bordas convencionais Sobel e Prewitt, utilizando filtros espaciais. Para programar o detector de borda morfológico Gblur foi necessário a implementação dos detectores de Erosão e Dilatação, na comparação visual entre os resultados da aplicação dos três detectores em uma imagem teste, a análise permitiu observar que o detector não linear, Gblur, apresentou melhor resultado em relação aos demais. Tal resultado evidencia o uso de ferramentas morfológicas para a aplicação requerida no presente trabalho.

A implementação dos detectores convencionais foi importante, porque possibilitou a comparação de detectores lineares com o não linear (morfológico), comprovando que o detector que utiliza a morfologia matemática se destacou com uma melhor detecção de borda encontrada no trabalho com o uso da imagem teste selecionada.

Revisão Bibliográfica

GONZALEZ, R. C. (2010); *Processamento Digital de Imagens.*; tradução: YAMAGAMI, C.; PIAMONTE, L. São Paulo: Pearson, 2010.

MATURANA, S. P.; *Algoritmos de Detecção de Bordas Implementados em FPGA*. 2010. Dissertação–Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira.

MELLO, C. A.; *Segmentação-Detecção de Bordas*, nota de aula Aula 5. UFPE.

SDC MORFHOLOGY "TOOLBOX" FOR MATLAB 5, SDC "Information Systems", User's Guide, April 24, 2000.

SOILLE, P. *Morphological image analysis: principles and applications. Springer-Verlag Berlin Heidelberg*, 1999.