

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação

Relatório Final

Segmentação de imagens de gráficos climatológicos

Matheus Sant'ana Lima
Bacharelado em Ciência da Computação

Orientador:
Prof. Dr. Jander Moreira (DC)

Agosto - 2010

Resumo

O trabalho consiste em, mediante a aplicação de técnicas em processamento de imagens, caracterizar a forma do traçado da pena sobre um gráfico climatológico, com a finalidade de obtenção das informações presentes. Os objetos de estudo considerados na pesquisa abrangem anemogramas obtidos através de digitalização de registros impressos na Estação Climatológica atualmente instalada na Universidade Federal de São Carlos, campus São Carlos-SP, e operada pelo Instituto Nacional de Meteorologia (INMET). Dentre as diversas aplicações desta pesquisa, destaca-se o importante auxílio que as informações obtidas destes gráficos podem gerar, como o suporte no planejamento de projetos de engenharia e arquitetura e a gestão de recursos de ações públicas em áreas de risco. A análise dos anemogramas, em particular os gráficos referentes às velocidades de ventos, serviram de base para o projeto. Para a segmentação da imagem, foi utilizada o algoritmo de classificação K-Médias, que classificou cada pixel da imagem em classes que representam o fundo do papel, a grade e o gráfico desenhado, sendo o último o objeto de interesse. Para a remoção da grade que forma a escala para o gráfico foi analisado cada pixel da imagem, e caso a maioria dos vizinhos tivesse sido classificado como fundo, o pixel em questão seria também classificado como fundo, caso contrário, seria atribuída a ele a classe de gráfico. Foi usado o filtro de média, para a remoção de ruídos e reconectar regiões. A técnica conhecida como *chain code* serviu para representar, através de uma sequência de direções, o contorno dos objetos segmentados, nos passos anteriores. O *smoothing* é utilizado para suavizar a sequência de direções, proporcionando uma suavização das bordas. O contorno da traçada da pena é reconstruído a partir do *chain code* modificado. O gráfico em questão possui como característica o movimento rápido da pena, o que forma, sobre o papel, linhas quase que indistintas entre si, dada a proximidade. Sendo inviável detectar e desnecessária cada posição da pena, a interpretação do gráfico é avaliada em termos de seu perfil de valores máximos e mínimos ao longo do tempo, caracterizando os perfis superior (valores máximos) e inferiores (mínimos). A caracterização do traçado da pena é feita com a interpretação das linhas que indicam os perfis superior e inferior, feito pelas concavidades das linhas que delimitam os valores do gráfico. Os resultados foram satisfatórios ao tornar possível analisar os pontos de máximos e mínimos do gráfico de velocidades presentes nos anemogramas, uma vez que a maioria dos pontos de interesse foi detectada. Para a implementação, foi utilizada a linguagem de programação C, juntamente com a biblioteca para visão computacional *Open Source*, OpenCV.

Introdução

O registro de dados climatológicos no Brasil, de forma geral, é bastante escasso e, quando existe, tem seu acesso dificultado por diversas razões. Entre elas, por se tratar de informações que devem ser coletadas por décadas seguidas e por ter condições de conservação e armazenamento muitas vezes precárias. Há, ainda, a conhecida descontinuidade, mudança de prioridade e mesmo descaso para com os registros em diferentes administrações. Outro aspecto, não menos importante, é a grande quantidade de gráficos, o que demanda um esforço humano e de equipamento, além de considerável dispêndio de tempo para trabalhá-los de forma convencional.

Os dados existentes trazem registros de velocidade, direção e rajadas de vento, presentes nos anemogramas, têm seu uso no cálculo de estruturas metálicas, de edifícios altos e nos estudo de conforto térmico de edificações, entre outros.

A vetorização deste gráfico pode ser realizada por meio de softwares comerciais que disponibilizam recursos de vetorização de imagem. Pode-se citar o AUTOCAD, geralmente os sistemas de informações geográficas como o SPRING, o ARCVIEW, o MICROSTATION, CORELDRAW etc. O problema relacionado ao seu uso, quando não o custo de aquisição e licença,

a dificuldade no uso, além de não terem sido projetados para este fim, pois os gráficos são “estruturas” simples, porém com singularidades e particularidades que, tanto quanto possível, devem ser soluções facilitadas, pois seu número é elevado. Tornam-se, assim, inadequados ao trabalho proposto.

Objetivos

Os objetivos deste trabalho incluem a segmentação de imagens digitalizadas de gráficos climatológicos para obtenção da linha que compõe o gráfico, preparando-a para vetorização de dados.

Como objetivos parciais destacam-se:

- Familiarização com os tipos de gráficos e suas características;
- Avaliação e implementação de técnicas de segmentação para obtenção da linha do gráfico;
- Avaliação e implementação de técnicas para tratamento da linha dos gráficos para prepará-las para a segmentação.

Metodologia

Para a realização deste plano de trabalho foram seguidos os passos:

- (a) Revisão bibliográfica sobre segmentação de gráficos e suas aplicações através do estudo de Castleman(1996), Dougherty(1992) e Gonzales(1992);
- (b) Elaboração de rotinas para implementação das técnicas selecionadas para segmentação dos gráficos e obtenção da linha (objeto de interesse) - Para esta etapa foi utilizada a linguagem de programação C, juntamente com a biblioteca *Open Source* para visão computacional, *OpenCV*(2000);
- (c) Avaliação da qualidade da segmentação tendo em vista a futura vetorização dos dados;
- (d) Avaliação das dificuldades da segmentação e elaboração de soluções para contornar os problemas encontrados (levantamento de técnicas e sua implementação);
- (e) Elaboração de artigo para evento científico (CIC ou outros);
- (f) Elaboração do relatório final do trabalho.

Técnicas aplicadas

As técnicas utilizadas nesta pesquisa são descritas na sequência

(a) Segmentação da Imagem da velocidade, dos Anemogramas, utilizando o algoritmo de K-Médias, para a classificação dos objetos componentes do gráfico: Fundo, Gráfico e Grade. Para cada componente ou classe, é aproximado um valor de cor, no padrão RGB, utilizando neste caso o cálculo da distância euclidiana para análise da similaridade, para segmentar cada pixel em uma das K classes. Como dito anteriormente, o algoritmo de K-Médias possui um valor aproximado de cor para cada objeto, desta forma, foram testadas duas abordagens distintas para a classificação destes centroides de cores: O algoritmo de SOM, baseado em redes-neurais, que utiliza valores iniciais de forma randômica e a abordagem utilizada no projeto, que é semi-automática, ao trabalhar com valores iniciais pré-definidos, que são utilizados para o cálculo das aproximações. A escolha dos

valores iniciais mais apropriados ainda é um problema em aberto na literatura pois, tanto a escolha randômica como a manual possuem desvantagens, como por exemplo, uma vez que a escolha é aleatória, uma mesma imagem poderá gerar diferentes segmentações, para diferentes execuções do mesmo código e, caso a segmentação não seja apropriada, a qualidade dos resultados produzidos pelo sistema é comprometido, em contra-partida, a escolha manual requer intervenção humana, o que diminui a eficiência do sistema. Abaixo, é reproduzido o algoritmo utilizado:

Algoritmo de segmentação de imagens por K-Médias

- **passo 1:** entre com a imagem, e inicializar: a quantidade k de grupos (*clusters*), os valores iniciais dos k centróides e definir o critério de parada;
- **passo 2:** calcular a similaridade entre os *pixels* com cada centróide;
- **passo 3:** atribuir cada *pixel* com o grupo do centróide mais similar;
- **passo 4:** calcular a média de cada grupo e atribuir esse valor a cada centróide de acordo com os grupos;
- **passo 5:** verificar o critério de parada, caso não seja satisfeito, voltar ao passo 2.

Figura 3. Algoritmo de K-Médias, segundo Takahashi(2005).

(b) Remoção da grade que interceptava o gráfico realizado pela pena é efetuada, pois, como o gráfico realizado pela pena é nosso objeto de interesse, tanto o fundo, como a grade que o intercepta são objetos indesejáveis, para tanto é analisado, para cada pixel, a maioria dos pixels vizinhos, sendo que, caso a maioria, utilizando janela de 6, tenha sido classificada como fundo, o pixel em questão é atualizado para fundo, caso a maioria tenha sido classificada como gráfico, o pixel é atualizado como gráfico. No caso de empate, é verificado utilizando janela de 8 e, caso isto também falhe, o pixel é atualizado como fundo. A Figura 4 ilustra os possíveis vizinhos para um pixel p , em que, os pixels vizinhos marcados com vermelho indicam região de gráfico e os marcados de verde, como região de fundo.

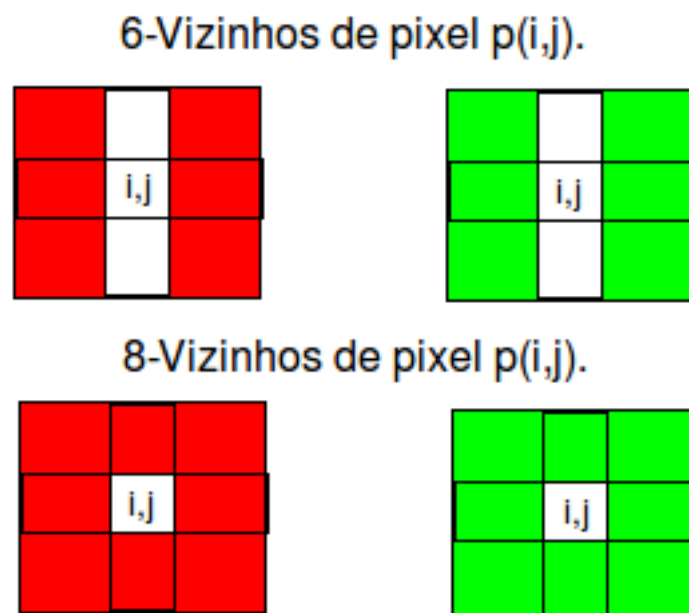


Figura 4. Janela de 6-Vizinhos e 8-Vizinhos, para fundo e gráfico.

(c) O algoritmo para remoção dos vizinhos apesar de eficiente ao remover *pixels* indesejáveis não apresenta resultados satisfatórios para todos os casos, desta forma é utilizado um filtro de *média aritmética*, que irá aproximar os valores para um dado pixel p , utilizando uma janela de 5x5, como demonstra a Figura 9. Com a utilização deste filtro, os contornos dos objetos são suavizados e algumas regiões desconexas são reagrupadas.



Figura 9: Janela 5x5, usada para cálculo da média.

$$\frac{1}{n} \sum_{i=1}^n x_i$$

(d) O algoritmo de *Chain Code* de Freeman(1961) é utilizado para descrever, utilizando uma sequência de caracteres, o comportamento do contorno do objeto de interesse, no caso, o desenho realizado pela pena, já segmentado, no gráfico de velocidade. Este algoritmo utiliza para orientação quatro direções possíveis(4-segmentos): Norte(2), Sul(6), Leste(0) e Oeste(4), ou ainda oito direções: Nordeste(1), Noroeste(3), Sudeste(7) e Sudoeste(5). Na figura 11, é apresentando um padrão para as 4 direções, utilizado na implementação desta técnica.

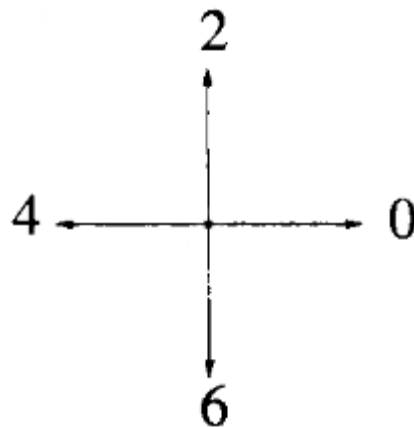


Figura 11: 4-direções para o algoritmo de Chain Code.

O algoritmo busca os objetos de interesse na imagem, percorrendo todo o seu contorno e, no final, retorna ao *pixel* inicial. Para eleger o próximo *pixel* marcado, é analisado o seu vizinho e verificado se ele é um *pixel* de contorno, caso seja, a direção atual é atualizada para a direção do

pixel seguinte eleito, até que seja encontrado o *pixel* inicial.

Na figura 12, é mostrada as direções propostas por Freeman e um exemplo de cadeia de caracteres, como consta em Marchard-Maillet(2005).

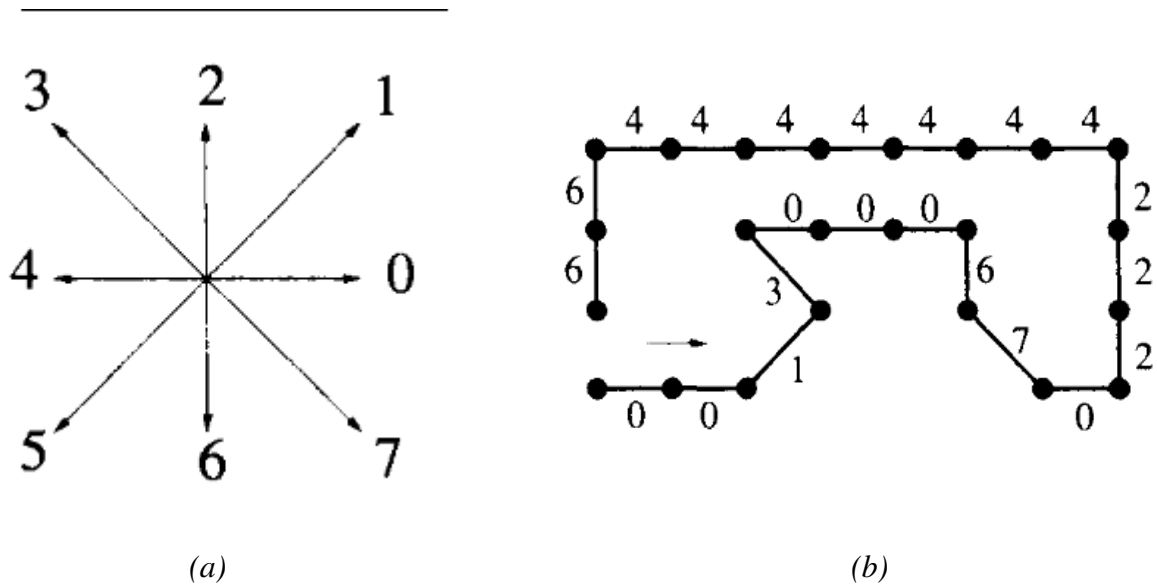


Figura 12: (a) Chain Code 8-direcional. (b) Exemplo de cadeia de caracteres.

(e) A cadeia de caracteres produzida pelo algoritmo de *Chain Code* gera marcações redundantes e desnecessárias para a descrição do contorno, desta forma, é aplicada uma segunda técnica, que suaviza esta sequência, através da análise de cada dois caracteres, substituindo, quando necessário, por um caractere que descreve o mesmo comportamento, utilizando, além das quatro direções citadas no passo (b), as direções: Nordeste(1), Noroeste(3), Sudeste(7) e Sudoeste(5), ilustradas na figura 15, e como resultado é produzida uma sequência menor. Algumas destas situações estão ilustradas na figura 14, em que códigos direcionais desenhados de forma tracejada, são substituídos por outros, desenhados de forma contínua.

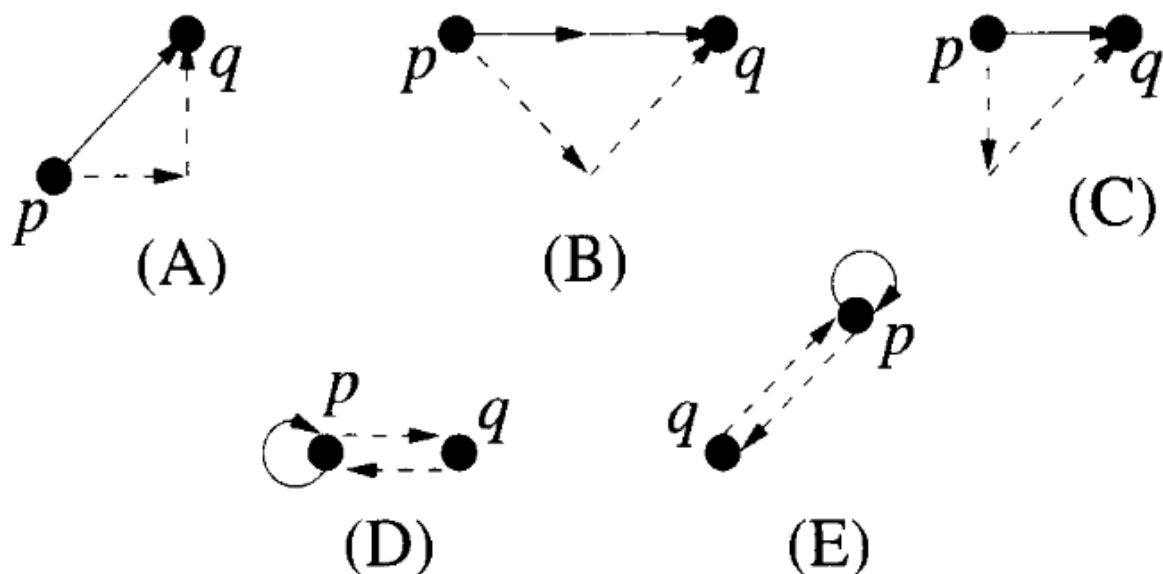


Figura 14: Templates de suavização do Chain Code segundo Marchard-Maillet(2005).

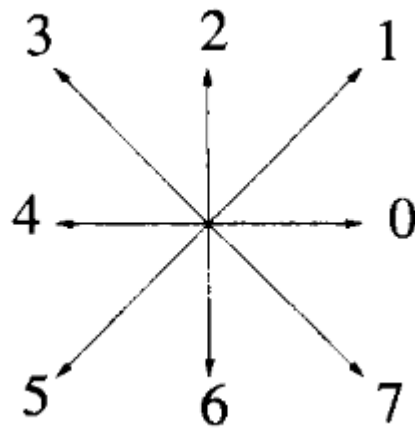


Figura 15: 8-direções para o algoritmo de Chain Code.

É possível observar que na figura 14 (d) e (e), o *pixel* q é removido, uma vez que representa um “pico” no contorno.

(f) A cadeia gerada pelo algoritmo de *Chain Code* representa todo o comportamento do contorno dos objetos, entretanto, é necessário para a representação do comportamento do gráfico, apenas os *pixel* que representam os perfis superior (valores máximos) e inferiores (mínimos), desta forma, para cada coluna da imagem, os *pixels* máximos e mínimos, são organizados em dois vetores distintos, que podem ou não serem suavizados, calculando a média aritmética para cada elemento.

g) Para a detecção dos pontos de máximo e mínimo, foi analisada a concavidade do gráfico, através da verificação dos vetores, para os perfis superiores e inferiores. Nesta abordagem para a detecção dos pontos de máximos e mínimo nas imagens, foi utilizado o conceito de vetor, proveniente do estudo do que é conhecido na matemática como Geometria Analítica. Foram testadas diferentes imagens, inicialmente não utilizamos filtro de média, depois média igual a 5, média igual a 7 e média igual a 9.

A ideia baseia-se no fato de que, dado um plano cartesiano (Dimensão igual a 2, neste caso), um vetor $v1 \rightarrow = AB \rightarrow = B - A$ e $v2 \rightarrow = CD \rightarrow = D - C$, como A, B, C e D são pontos existentes no gráfico presente na imagem e $B = C = X$, temos que, caso ocorra uma transição de $v1$ para $v2$, tal que, $v1$ seja positivo e $v2$ negativo ou $v1$ seja negativo e $v2$ positivo, então o ponto X indica um *pixel* que marca o máximo ou o mínimo de uma função do gráfico representando na imagem. Na figura 22, é apresentada uma representação deste cenário.

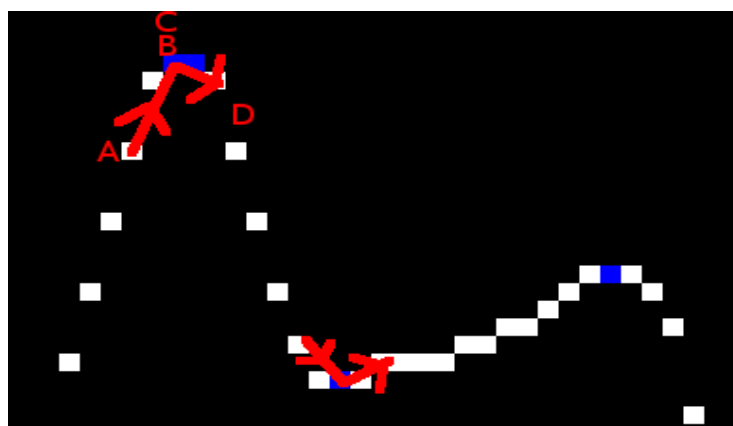


Figura 22: Exemplo de vetores para análise da concavidade.

Resultados e discussão

Os tópicos a seguir apresetam os resultados encontrados na implementação das técnicas abordadas na seção anterior:

(a) K-Médias: Nas figuras 1 e 2, é possível ver uma imagem parcial do gráfico de velocidade, extraído de um anemograma, após a aplicação do algoritmo implementado:

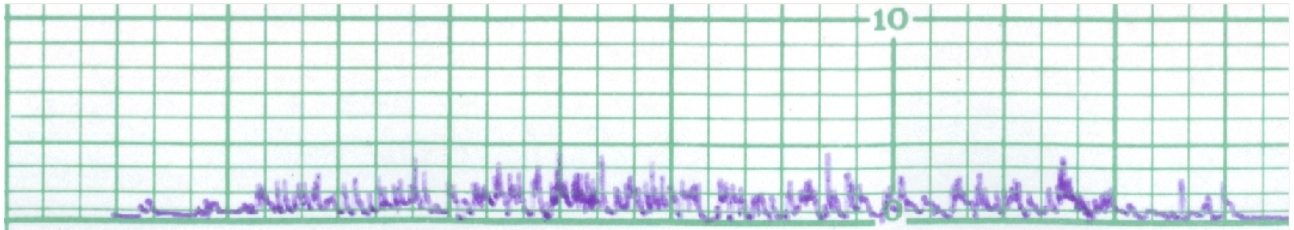


Figura 1: Imagem parcial, referente ao gráfico de velocidade.

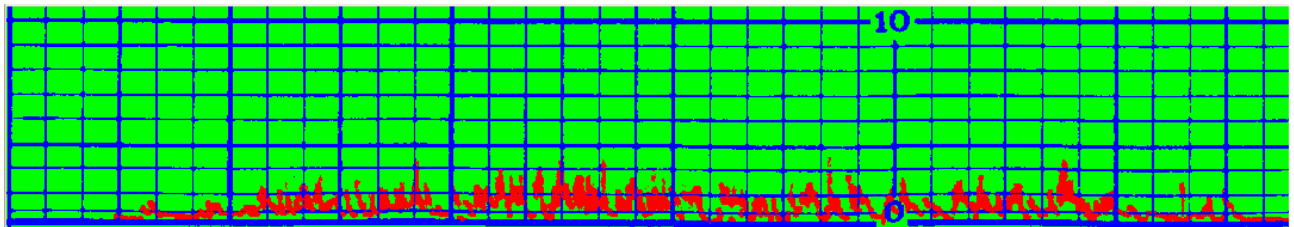


Figura 2. Imagem resultante, após a aplicação do algoritmo de K-Médias, na Figura 1.

É possível observar que, após a execução do algoritmo, a imagem resultante, que anteriormente não possuía agrupamentos, ou seja, não havia classificação para os pixels existentes, foi segmentada em apenas três centros médios(centróides) de cores, no caso, vermelho para o gráfico, verde para o fundo e azul para a grade. Este tipo de segmentação contrasta com a Limiarização Global, que baseia -se em apenas um único limiar T , como consta em Gonzalez(1992), para a classificação dos objetos na imagem, o que, para este tipo de imagem, é insuficiente, dado que é necessário segmentar três objetos diferentes, cada um com seu respectivo limiar.

(b) Remoção de Vizinhos: A execução deste algoritmo é demonstrado nas Figuras 5 e 6. As cores utilizadas, foram dispostas da seguinte forma: Em vermelho, o gráfico segmentado, em azul, a grade que o intercepta e, em verde, o fundo.

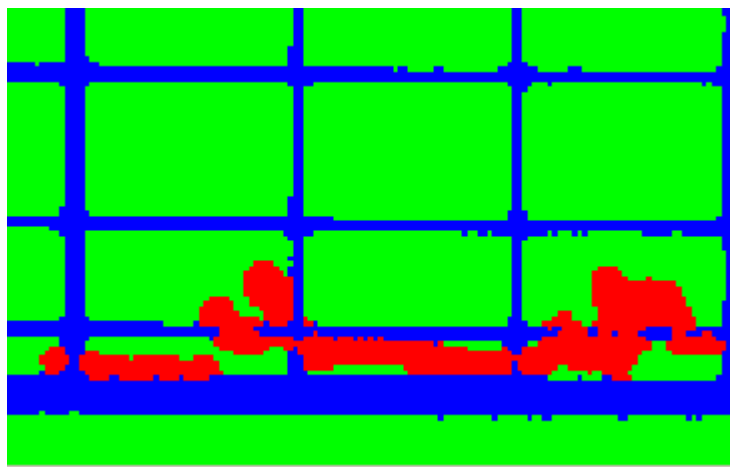


Figura 5: Imagem parcial, resultante do algoritmo de K-Médias.

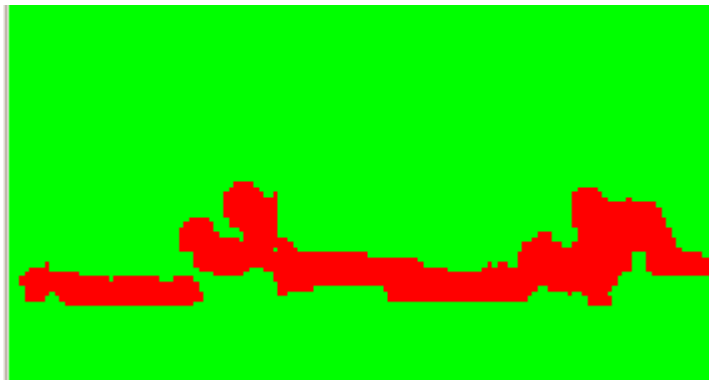


Figura 6. Imagem resultante da execução do algoritmo de análise dos vizinhos, na Figura 5.

Como resultado à execução da análise dos pixels vizinhos, é feita a conexão de algumas regiões que estavam anteriormente separadas na imagem resultante do algoritmo de K-Médias, isto é demonstrado na Figura 6.

(c) Filtro de Média Aritmética: A grade foi removida no passo (b), entretanto, o gráfico é erodido também em regiões que possuem apenas grade como vizinho, isto ocorre em regiões em que o desenho realizado pela pena e a grade se interceptam e a aproximação realizada pelo K-Médias não segmenta de forma apropriada, como demonstra a figura 7. Com a utilização do filtro de Média Aritmética, os contornos dos objetos são suavizados e algumas regiões desconexas são reagrupadas, como demonstra a figura 8.

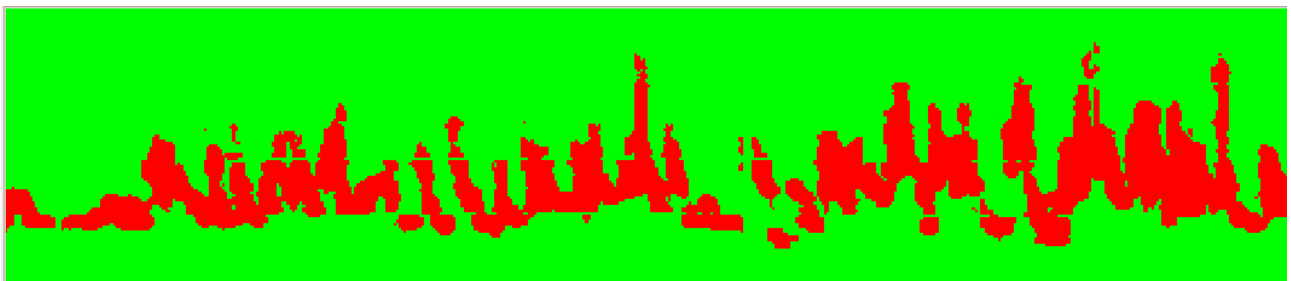


Figura 7: Imagem resultante do algoritmo para remoção de grades.

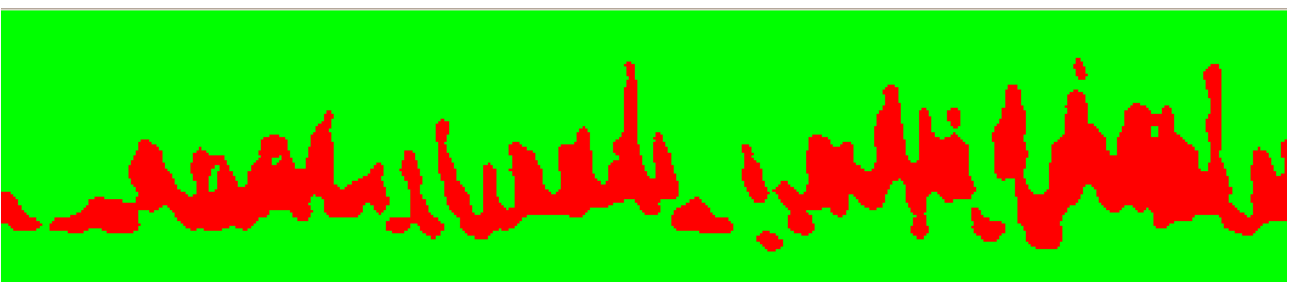


Figura 8: Imagem resultante da suavização pelo filtro de média aritmética, aplicada na Figura 7.

(d) Algoritmo de Chain-Code: Na figura 13, é possível analisar a detecção, pelo algoritmo de *Chain Code*, da borda do gráfico de velocidade, aplicada na imagem suavizada pelo filtro de média. Com isto, é possível a descrição do contorno do objeto de interesse apenas com uma sequência de caracteres, que será utilizada posteriormente para a análise do comportamento do gráfico.



Figura 13: Os pixels em preto foram detectados pelo algoritmo de Chain Code, destacando, desta forma, o contorno do gráfico.

(e) Suavização do Chain-Code: A figura 13 ilustra o contorno detectado pela implementação do algoritmo de Chain Code, sendo possível constatar que a sequência apresenta irregularidades quanto aos pixels marcados, na figura 16 são destacados alguns deles e na figura 17, é apresentando o resultado da figura 16 depois de sucessivas suavizações, o padrão de coloração nesta figura foi: Em vermelho, o gráfico segmentado, em azul, a grade que o intercepta e, em verde, o fundo. Na figura 17, as marcações em branco indicam a borda suavizada.

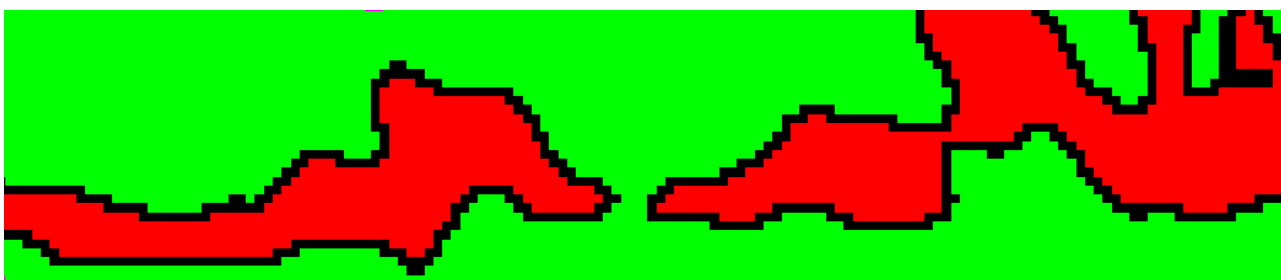


Figura 16: Contorno detectado pelo algoritmo de Chain Code.



Figura 17: Sequencia suavizada, a partir da figura 16.

A suavização do Chain Code reduz as imperfeições e redundâncias do contorno encontrado, o que é importante para a qualidade da análise dos pontos de máximo e mínimo, que devem ser detectados no gráfico, pois diminui a quantidade de pixels marcados de forma incorreta.

(f) Perfis Superior e Inferiores: Na figura 18, o gráfico é representado utilizando média 3; na figura 19, utilizando média 5; na figura 20, utilizando media 7 e na figura 21, utilizando média 9.



Figura 18: Pontos superiores e inferiores, suavizados com média 3.



Figura 19: Pontos superiores e inferiores, suavizados com média 5.



Figura 20: Pontos superiores e inferiores, suavizados com média 7.



Figura 21: Pontos superiores e inferiores, suavizados com média 9.

(g) Pontos de Máximo e Mínimo: Para a realização dos testes, foi utilizada a figura 23, que apresentou os melhores resultados com o filtro de média 3, para suavização do contorno. Os *pixels* marcados em vermelho, no perfil superior, indicam pontos de máximo, os marcados em preto, indicam pontos de mínimo. No perfil inferior, os *pixels* marcados em amarelo, indicam pontos de máximo, os marcados em verde, indicam pontos de mínimo.

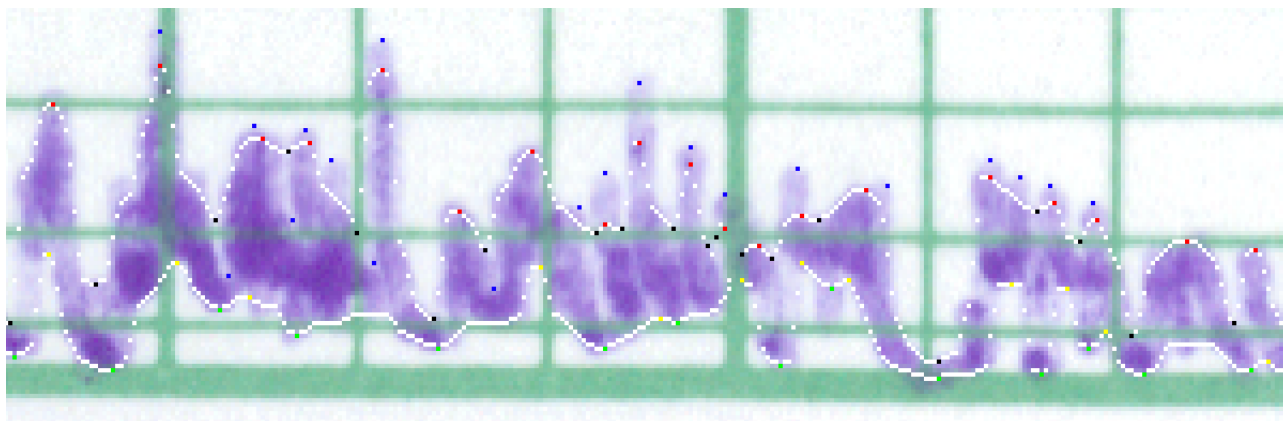


Figura 23: Imagem parcial resultante do gráfico de velocidade, nos anemogramas.

É possível observar que existe uma distancia, em alguns *pixels*, entre os pontos de máximo/mínimo reais, ou seja, desenhados pela pena, e os detectados pelo software, sendo que os *pixels* com a posição correta estão destacados na imagem em azul. A tabela 1, relaciona a quantidade de *pixels* marcados de forma correta e com deslocamento.

Numero de <i>pixels</i> em posição correta	Numero de <i>pixels</i> em posição deslocada
46	20

Tabela 1: Avaliação da qualidade dos resultados da figura 23.

Conclusões

Foi observado, através da análise dos resultados obtidos com a figura 23, que os *pixels* que caracterizam os pontos de máximo e mínimo foram, em sua maioria, detectados com considerável grau de acerto. Entretanto, apesar do resultado satisfatório, alguns destes *pixels* sofrem pequeno desvio, quando comparado ao gráfico real, desenhado pela pena. Este comportamento indesejável, ocorre por diversos fatores, como aproximação imperfeita dos agrupamentos pelo algoritmo de K-Médias; Suavização imperfeita para o filtro de média aritmética; Suavização imperfeita para a sequencia de Chain Code e Perfis de máximo e mínimo sensíveis à suavização pela média aritmética.

Foi concluído, portanto, que a pesquisa cumpriu com seus objetivos pois permitiu a segmentação do desenho do gráfico realizado pela pena e reconheceu os padrões que representam o comportamento deste gráfico, ou seja, seus pontos de máximos e mínimos. Desta forma, as informações pertinentes à vetorização, ou seja, a aproximação dos pontos do gráfico para uma função $f(x)$, ficam a disposição para a submissão para este processo.

Referências Bibliográfica

- Castleman, K. R. *Digital Image Processing*. Upper Saddle River, New Jersey: Prentice Hall, 1996. 667p.
- Dougherty, E. R. *An introduction to morphological image processing*. SPIE Press, 1992.

Freeman, H. On the encoding of arbitrary geometric configurations. IRE Trans. In: Electronic Computer, 1961

Gonzalez, R. C.; Woods, R. E. *Digital image processing*. United States of America: Addison-Wesley Publ. Co. 1992. 716p. (World Student Series).

Intel. *Open Source Computer Vision Library - Reference Manual*. Disponível em: <http://www.lamce.ufrj.br/grva/data/realidade_aumentada/index.php?go=opencv/OpenCVReferenceManual.pdf>. 2000. Acesso em 17 maio 2010 .

Marchard-Maillet, S. Sharaiha, Y.M. *Binary Digital Image Processing A Discrete Aproach*. San Diego, California: Academic Press, 2000. 215p.

A. Takahashi, B.R.C. Bedregal, A. Lyra. *Uma Versão Intervalar do Método de Segmentação de Imagens Utilizando o K-Médias* . In: TEMA Tend. Mat. Apl. Comput. 315-324, 6, No. 2, 2005.