

UM SISTEMA DE PROCESSAMENTO DE IMAGENS DIRIGIDO A MICROCOMPUTADOR COMPATÍVEL COM PC.

ADILSON GONZAGA¹
CELSON APARECIDO DE FRANÇA²

¹EESC-USP - Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica
Av. Dr. Carlos Botelho, 1465
São Carlos, SP, Brasil

²EESC-USP - Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica
Av. Dr. Carlos Botelho, 1465
São Carlos, SP, Brasil

ABSTRACT. This paper describes the development of an image processing system based on a PC microcomputer. The digitized images are obtained by a low cost handy scanner. The program generates the gray scale image using dithering images. The processing techniques consist of counting sixteen pixels of ones or zeros in a 4 x 4 matrix, moved over the image and generating the equivalent gray level. Some algorithms of edge detectors have been implemented and some results are appreciated.

1. Introdução.

Um problema que aparece aos pesquisadores em Visão Computacional e Processamento de Imagens, com disponibilidade de uso de máquinas de baixo custo e relativo desempenho gráfico (386, 486, etc...), é a inexistência de pacotes de processamento de imagens com código fonte aberto para permitir a inclusão de rotinas do usuário. As imagens de entrada, obtidas por exemplo através de um "Scanner" de mão, ou geradas por algum outro processo, são armazenadas compactadas em formatos específicos (.IMG, .PCX, .TIF, .GIF, etc...). Estas imagens podem também estar codificadas em escala de cinza, em cores, ou em dois níveis através de "dithering", além disso, podem também estar armazenadas de forma vetorial ou mapeada por bit ("Bit-mapped").

A proposta deste trabalho, é a de desenvolver um sistema de baixo custo para o processamento de imagens e Visão Computacional, a partir de imagens geradas por um "Handy Scanner" de dois níveis através de "dithering" e no formato "Bit-mapped". Estas imagens são então descompactadas e transformadas em escala de cinza, permitindo assim, a implantação de algoritmos de processamento de imagens em um microcomputador compatível com PC.

2. Descompactação de Imagens.

Uma grande quantidade de formatos de arquivos de imagem existem, praticamente um para cada aplicação particular. Alguns são mais populares como o .IMG desenvolvido pela Digital Research para o GEM, o .PCX para o PC Paintbrush da ZSoft, o .GIF da CompuServe e o .TIF ou "Tagged Image File Format" da Aldus e Microsoft.

Os dados em um arquivo de imagem codificado são em geral de três tipos [Rimmer (1990)]:

- 1) Bytes de chave: Dizem ao decodificador o que fazer com o próximo pacote de dados.
- 2) Bytes de índice: Dizem ao decodificador quanto do resto do arquivo deverá ser incluído no pacote corrente.
- 3) Bytes de dados: Bytes que serão decodificados para recuperar a imagem.

O formato .IMG permite imagens de diversos tamanhos e possui um cabeçalho 16 ou 18 bytes, um dos menores entre os formatos existentes. Cada palavra é armazenada na forma direta com o byte mais significativo em primeiro lugar.

Após o cabeçalho, cada plano da imagem é armazenado como uma seqüência de linhas de varredura. Cada linha tem um contador de repetição opcional seguido de um ou mais trechos de dados de imagem.

O programa desenvolvido promove a descompactação de uma imagem no formato .IMG, transformando-a em uma matriz numérica correspondente à imagem.

Cada bit de cada byte desta matriz representa um pixel da imagem dentro da linha de varredura do display, da esquerda para a direita e de cima para baixo. O pixel é considerado aceso, ou branco, quando igual a "zero" ou apagado (preto) quando igual a "um".

Esta imagem obtida através do "Scanner" manual pode ser simplesmente binária ou representar tons de cinza através da técnica de semi-tons ou "dithering".

3. Imagens em semi-tons.

Como grande parte dos equipamentos de reprodução de imagens em papel, opera em dois níveis, como por exemplo as impressoras matriciais e a Laser, a imagem em dois níveis é amplamente utilizada. Para que a aparência de escala de cinza seja obtida, os valores das intensidades são distribuídos espacialmente de acordo com determinados algoritmos, causando a impressão de escala de cinza. Este fenômeno é explorado em impressões de revistas, jornais e livros em uma técnica conhecida como "half-toning" ou "dithering" em computação gráfica. Várias técnicas existem [Jarvis et al. (1976)] [Foley et al. (1990)] para a distribuição dos pontos pretos e brancos em uma área, correspondente às intensidades da imagem em cada ponto.

Qualquer das técnicas existentes para distribuir o efeito de cinza sobre um arranjo de dois níveis mantém, no entanto, a proporcionalidade entre números de pontos pretos e nível de cinza equivalente. A imagem dita assim "ditherizada" apresenta o aspecto da Fig.1. Isto, a nível de visualização impressa, é suficiente, mas não é viável o seu processamento, pois seus tons de cinza não compõe uma matriz de pontos da imagem.

Os "Scanners" de mão de mais baixo custo geram imagens "ditherizadas" em diversos formatos (.IMG, .PCX, .TIF, etc...). Utilizando um destes digitalizadores podemos obter imagens de fotografias, tanto coloridas como em preto e branco, "ditherizadas"



Fig.1 - Imagem "ditherizada".

a nível de cinza equivalente.

Independente do algoritmo empregado para gerar este efeito (Bayer, Floyd-Steinberg, Holladay [Foley...]), o processo de geração da matriz de imagem em nível de cinza, pode ser único. Se deslocarmos uma

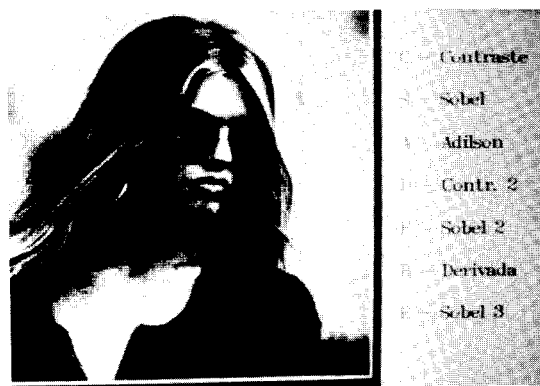


Fig. 2 - Imagem em 16 níveis de cinza.

matriz geradora de pixel em nível de cinza ($p \times p$), sobre uma imagem "ditherizada" e contarmos o número de pontos pretos, ou seja, bits em nível lógico um, podemos ter a representação em p^2 níveis de cinza.

Dessa forma, obtem-se da imagem da Fig.1 sua matriz e o resultado mostrado no display do microcomputador com placa VGA (640 x 480 x 16 cores) é apresentado na Fig.2. Esta imagem, agora em escala de 16 tons de cinza pode ser processada, com resultados mostrados no mesmo display.

4. Processamento da Imagem.

A Visão Computacional lida com a derivação de descrições do mundo através das imagens deste mundo. Nestas imagens qualquer tipo de estrutura significativa aparece como descontinuidade na intensidade luminosa da cena. Um esforço razoável tem sido realizado no sentido de se determinar técnicas eficientes para a detecção destas descontinuidades. Uma imagem obtida do mundo real apresenta uma grande quantidade de informação entre dados relevantes e ruídos, que devem ser reduzidos para uma maior eficiência dos algoritmos de reconhecimento e classificação.

As técnicas de Visão Computacional a Nível Básico operam sobre imagens reais binárias, em escala de cinza ou em cores, reduzindo a quantidade de informação presente na cena. Nesta etapa, procuram detectar descontinuidades significativas da cena representadas através das bordas dos objetos e de suas fronteiras. É necessário também, que o ruído seja minimizado e que apenas pontos representativos da estrutura da cena possam ser gerados.

5. Detecção de Bordas.

As bordas de uma imagem são o resultado de mudanças em alguma propriedade física ou espacial de superfícies iluminadas.

A maioria das técnicas de detecção de bordas empregam operadores diferenciais [Pratt (1978)] de primeira ou de segunda ordem. Os operadores diferenciais ressaltam os contornos das bordas mas também amplificam o ruído da cena. Alguns trabalhos discutem comparativamente alguns destes operadores mais conhecidos [Fleck (1992)] [Peli/Malah (1982)]. Grande parte dos operadores de borda utilizam algum tipo de suavização da imagem antes da operação diferencial. O operador de Marr utiliza uma máscara Gaussiana convoluída com a imagem; o operador de Canny também suaviza a imagem. Isto pode atenuar as bordas fracas, onde o contraste é pequeno.

O nosso sistema permite a implantação de qualquer algoritmo de detecção de bordas, pois a imagem obtida através de "Scanner" e descompactada em 16 níveis de cinza pode ser processada através de linguagem C e mostrada em um terminal de vídeo VGA.

A primeira técnica aplicada leva em consideração apenas imagens de alto contraste cujo ruído fique abaixo de um limiar. Compreende duas etapas:

1) Binarização da imagem através da aplicação de um nível de "threshold" que pode por exemplo ser calculado através do vale do histograma.

2) Aplicação de OU-Exclusivo nas linhas e colunas, bit a bit, da imagem binária.

A operação de OU-exclusivo entre os elementos binários da imagem é o meio mais rápido de detecção de bordas que envolve menor custo computacional. O procedimento pode ser realizado em Assembler ou em

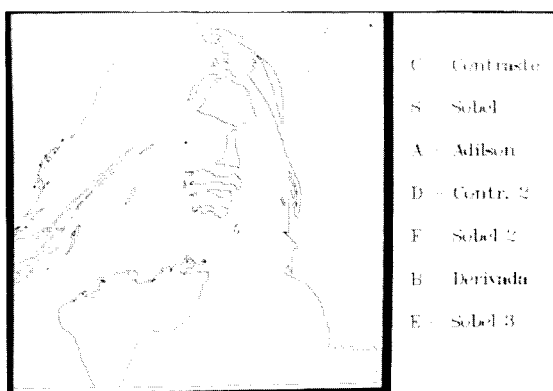


Fig.3 - Imagem de bordas. Método OU-exclusivo.

linguagem C. A borda é detectada onde existe variação de bit "zero" para bit "um" ou vice-versa, permanecendo constante e igual a "zero" em partes toda

branca ou toda preta da imagem. A Fig.3 mostra uma imagem de bordas, gerada a partir da imagem da Fig.2 através de:

$$b_{(i,j)} = (a_{(i,j)} \oplus a_{(i,j+1)}) \cdot (a_{(i,j)} \oplus a_{(i+1,j)})$$

Evidentemente é um detector extremamente simples mas que tem suas aplicações em imagens industriais, algumas imagens médicas e OCR.

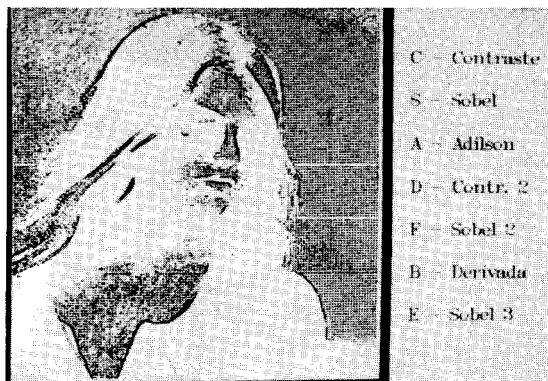


Fig.4 - Detecção de bordas por Sobel.

Um dos detectores mais populares é o detector de borda de Sobel. A imagem vista na Fig.4 é uma imagem de bordas detectadas por Sobel, aplicando-se um nível de "threshold". Pode-se observar que o ruído proveniente de variações no nível de cinza do fundo é ressaltado e as regiões de baixa iluminação, com bordas fracas, não é detectada. Shu sugere [Shu (1989)] após a aplicação de Sobel um "threshold" diferenciado e uma busca heurística para detecção de bordas de um pixel. A

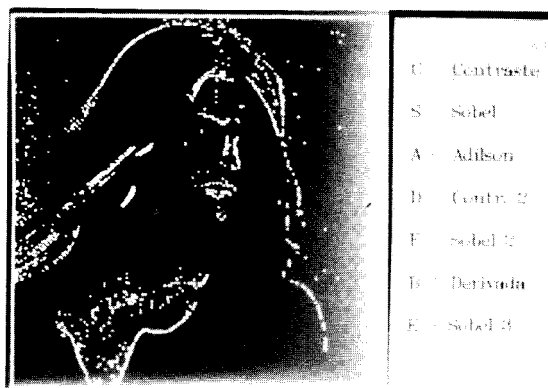


Fig.5 - Detecção de bordas usando "threshold" diferenciado no detector de Sobel.

Fig.5 apresenta a imagem processada de acordo com a sugestão de Shu, com um "threshold" diferenciado.No

programa desenvolvido, não foi implementada a busca heurística proposta devido ao custo computacional envolvido.

O algoritmo de Johnson aplicado sobre Sobel [Johnson (1990)] divide o valor de $B(i,j)$ resultante da convolução de Sobel, pela média da vizinhança em uma máscara de 3×3 , somada a uma constante "d", ou seja:

$$S(i, j) = \frac{B(i, j)}{(1/9A(i, j) \otimes C) + d}$$

onde "d" é escolhido de tal forma a providenciar uma melhor relação entre a sensibilidade de ruído e o

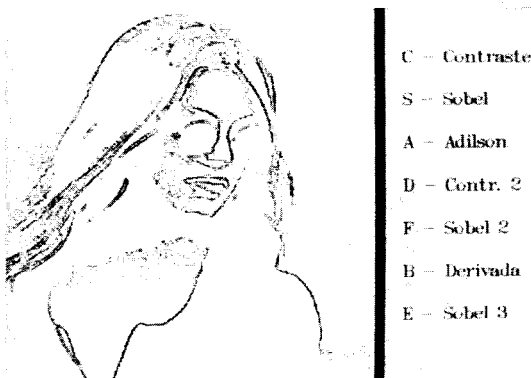


Fig.6 - Aplicação do algoritmo de Johnson em Sobel.

"range" dinâmico para a combinação do sensor e da cena. A máscara C é um "template" de 3×3 , com todos seus valores iguais a "um".

O resultado deste equacionamento é visto na Fig.6 não tendo sido obtido um resultado tão bom como o mostrado em seu artigo. Acontece que a escolha do parâmetro "d" é difícil para nosso sistema e nenhuma informação adicional é fornecida no referido trabalho.

6. Conclusões.

O sistema apresentado neste trabalho é a primeira fase do desenvolvimento mais completo de um Sistema de Visão Computacional. O código fonte gerado em linguagem C, deverá ser ampliado para permitir testes comparativos de detectores de bordas mais elaborados, algoritmos de afinamento, técnicas de segmentação, rotulagem, descrição e reconhecimento de objetos em cenas complexas obtidas de imagens em nível de cinza.

Os resultados até agora obtidos mostram-nos um sistema de baixo custo, instalável em qualquer microcomputador do tipo PC com VGA e um "scanner" manual de dois níveis. Caso a imagem já esteja em

escala de cinza, o programa realiza apenas a descompactação, gerando a matriz de imagem.

Outros formatos, além do .IMG, estão sendo estudados e novos módulos serão acrescentados ao programa fonte, permitindo trabalhar com formatos .PCX, .TIF e .GIF.

Os detectores de bordas avaliados neste sistema nos permitem concluir por uma melhor técnica a ser aplicada em cada caso em particular.

Apesar da limitação em 16 níveis de cinza, o sistema apresenta um resultado bastante satisfatório no que se refere ao estudo comparativo de diferentes algoritmos.

Finalmente, o sistema deverá ser continuado, ampliando seu alcance e o código fonte em linguagem C, está disponível a quem necessitar processar ou desenvolver novos algoritmos em microcomputadores do tipo PC.

Referências Bibliográficas.

[Foley et al. (1990)]

Foley, J.D., Van Dam, A., Feiner, S.K. and Huges, J.F. - *Computer Graphics, principles and practice*, Addison-Wesley Pub. Co., (1990).

[Jarvis et al. (1976)]

Jarvis, J.F., Judice, C.N. and Ninke, W.H. - A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays, *Computer Graphics and Image Processing* 5 (1976) 13-40.

[Johnson (1990)]

Johnson, R.P. - Contrast Based Edge Detection, *Pattern Recognition* 23 (3/4), 311-318 (1990).

[Peli/Malah (1982)]

Peli, T. and Malah, D. - A study of edge detection algorithms, *Computer Graphics and Image Processing* 20, 1-21 (1982).

[Pratt (1978)]

Pratt, W.K. - *Digital Image Processing*, John Wiley and Sons, New York (1978).

[Rimmer 1990]

Rimmer, S. - *Bit-Mapped Graphics*, Windcrest Books, McGraw-Hill Inc., (1990).

[Shu (1989)]

Shu, J.S.P. - One-Pixel-Wide Edge Detection, *Pattern Recognition* 22 (6), 665-673 (1989).