Algoritmos e Estruturas de Dados

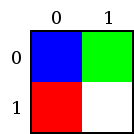
Trabalho Opcional

Prof. Alexandre Zaghetto  
[zaghetto@unb.br](mailto:zaghetto@gmail.com)

Escreva o programa **em linguagem de programação C.**

No presente trabalho vamos explorar alguns princípios da filtragem de imagens no domínio espacial em conjunto com os conceitos de vetor, matriz, *struct*, ponteiro, leitura e escrita em arquivos binários. Para isso:

1. Estude o formato de armazenamento de arquivos de imagem *bmp* (bitmap), exemplificado a seguir:

[](http://en.wikipedia.org/wiki/File:Bmp_format.svg)

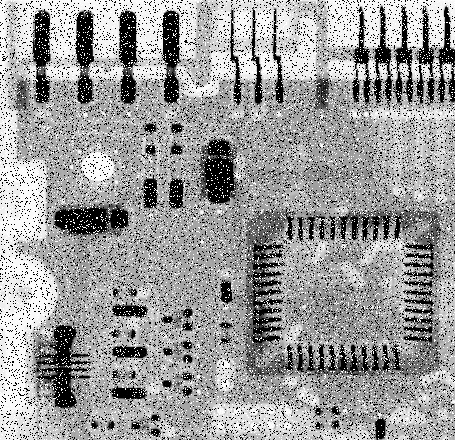
Example of a 2x2 Pixel Bitmap, with 24 bits/pixel encoding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Offset** | **Size (bytes)** | **Hex Value** | **Value** | **Description** |
| 0h | 2 | 42 4D | "BM" | Magic Number (unsigned integer 66, 77) |
| 2h | 4 | 46 00 00 00 | 70 Bytes | Size of the BMP file |
| 6h | 2 | 00 00 | Unused | Application Specific |
| 8h | 2 | 00 00 | Unused | Application Specific |
| Ah | 4 | 36 00 00 00 | 54 bytes | The offset where the bitmap data (pixels) can be found. |
| Eh | 4 | 28 00 00 00 | 40 bytes | The number of bytes in the header (from this point). |
| 12h | 4 | 02 00 00 00 | 2 pixels | The width of the bitmap in pixels |
| 16h | 4 | 02 00 00 00 | 2 pixels | The height of the bitmap in pixels |
| 1Ah | 2 | 01 00 | 1 plane | Number of color planes being used. |
| 1Ch | 2 | 18 00 | 24 bits | The number of bits/pixel. |
| 1Eh | 4 | 00 00 00 00 | 0 | BI\_RGB, No compression used |
| 22h | 4 | 10 00 00 00 | 16 bytes | The size of the raw BMP data (after this header) |
| 26h | 4 | 13 0B 00 00 | 2,835 pixels/meter | The horizontal resolution of the image |
| 2Ah | 4 | 13 0B 00 00 | 2,835 pixels/meter | The vertical resolution of the image |
| 2Eh | 4 | 00 00 00 00 | 0 colors | Number of colors in the palette |
| 32h | 4 | 00 00 00 00 | 0 important colors | Means all colors are important |
| Start of Bitmap Data | | | | |
| 36h | 3 | 00 00 FF | 0 0 255 | Red, Pixel (1,0) |
| 39h | 3 | FF FF FF | 255 255 255 | White, Pixel (1,1) |
| 3Ch | 2 | 00 00 | 0 0 | Padding for 4 byte alignment (Could be a value other than zero) |
| 3Eh | 3 | FF 00 00 | 255 0 0 | Blue, Pixel (0,0) |
| 41h | 3 | 00 FF 00 | 0 255 0 | Green, Pixel (0,1) |
| 44h | 2 | 00 00 | 0 0 | Padding for 4 byte alignment (Could be a value other than zero) |

b) Escreva um programa para ler as informações do cabeçalho do arquivo (desde “Magic Number” até “Means all colors are important”) para uma *struct*, mostrando em seguida os dados na tela do computador.

c) Leia os dados da imagem (a partir de Start of Bitmap Data) para três matrizes matR, matG e matB alocadas dinamicamente a partir dos valores de altura e largura da imagem lidos no cabeçalho.

d) Aplique o filtro de média ou de mediana aos três planos, de acordo com a chamada feita pelo usuário, gerando planos filtrados matRfilt, matGfilt e matBfilt. O tamanho do filtro (3x3, 5x5, etc.) deve ser fornecido ao programa via linha de comando, juntamente com o nome da imagem de entrada e da imagem de saída. Verifique o efeito de ambas as filtragens na imagem de teste boardsaltandpepper.bmp e lena\_noisy.bmp de fornecidas.

e) Escreva apropriadamente o cabeçalho e os novos planos matRfilt, matGfilt e matBfilt em um arquivo binário, com a extensão *bmp*, e observe o efeito da filtragem em um visualizador de imagens qualquer.