Quantização de imagens digitais *

Lenimar N. Andrade

15 de maio de 1999

Resumo

Este texto descreve alguns algoritmos que podem ser utilizados na quantização de imagens digitais. É brevemente descrito como usar o Thumbs para efetuar algumas operações com imagens.

Sumário

1 Definições básicas

Chamamos *imagem digital* a uma função $f : U \longrightarrow \mathbb{R}^3$ onde o suporte U e o espaço de cores $\mathbb{R}^3$ estão discretizados.

Essa imagem é caracterizada pelos seguintes fatores:

- resolução espacial (número de pixels)
- número de cores
- resolução (tons) de cor (n bits $\Rightarrow$ 2^n tons)

O *gamute* de f é o conjunto de cores $f(U)$, que é um conjunto finito. Se $|f(U)| = 2$ então dizemos que a imagem é *binária*.

O *histograma de cor* é um gráfico que associa a cada intensidade de cor C presente na imagem o número de pixels da imagem que utilizam a cor C.

Quantização é uma função sobrejetiva $q : \mathbb{R}^3 \longrightarrow R_k$, onde $R_k = \{p_1, p_2, \cdots, p_k\} \subset \mathbb{R}^3$. Cada p_i é chamado um *nível de quantização*.

O resultado de uma quantização é uma imagem $g : U \longrightarrow R_k$ onde $g = q \circ f$.

*Disponível em ftp://mat.ufpb.br/pub/docs/cursos/quantiz.zip

2 Quantização com limiar constante

A quantização para duas cores (C_0 e C_1, $C_0 < C_1$) com limiar (*threshold*) constante L_0 consiste na substituição de $f(i,j)$ pela cor C_1 se $f(i,j) \geq L_0$ e pela cor C_0 se $f(i,j) < L_0$, $\forall (i,j) \in U$. Para uma imagem com resolução espacial $m \times n$, isto pode ser descrito pelo seguinte fragmento de programa utilizando a linguagem de programação C:

```
for (i = 0; i <= m; i++)
  for (j = 0; j <= n; j++)
    if (f[i][j] >= L0)
      f[i][j] = c1;
    else
      f[i][j] = c0;
```

3 Algoritmo de Floyd-Steinberg (1975)

Quantifica para duas cores C_0 e C_1, calcula o erro na substituição e propaga o erro para os pixels vizinhos. Suponhamos $C_0 < C_1$.

Dada uma constante L_0, para cada $(i,j) \in U$, se $f(i,j) \geq L_0$ então substituímos $f(i,j)$ por C_1 e calculamos o erro $\epsilon = f(i,j) - C_1$. Se $f(i,j) < L_0$ então substituímos $f(i,j)$ por C_0 e calculamos o erro $\epsilon = f(i,j) - C_0$. Em seguida, distribuímos o erro calculado para os pixels vizinhos: $3/8$ do erro para os vizinhos leste e sul e $1/4$ do erro para o vizinho a sudeste, ou seja, somamos $3\epsilon/8$ a $f(i+1,j)$ e a $f(i,j-1)$ e somamos $\epsilon/4$ a $f(i+1,j-1)$.

Uma boa opção para L_0 é a média aritmética das cores C_0 e C_1, isto é, $L_0 = \frac{C_0+C_1}{2}$.

Com resolução espacial $m \times n$, isto pode ser descrito pelo fragmento:

```
L0 = (c0 + c1)/2;
for (i = 0; i <= m; i++)
  for (j = 0; j <= n; j++) {
    if (f[i][j] >= L0) {
      f[i][j] = c1;
      Erro = f[i][j] - c1;
    } else {
      f[i][j] = c0;
      Erro = f[i][j] - c0;
    }
    f[i+1][j] = f[i+1][j] + 3*Erro/8;
    f[i][j-1] = f[i][j-1] + 3*Erro/8;
    f[i+1][j-1] = f[i+1][j-1] + Erro/4;
  }
```

4 *Dithering* ordenado

As operações de quantização interessantes ocorrem quando a função limiar não é constante. Neste caso, essas operações são chamadas de *dithering* (= hesitação).

A operação de quantização conhecida como *dither de Bayer de ordem n* (n potência de 2) consiste em tomar a função limiar variando em uma matriz de ordem $n \times n$.

É comum escolher a matriz D_2 como sendo $D_2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$ e, de um modo geral, se n for uma potência de 2 tal que $n \geq 4$, então $D_n = \begin{bmatrix} 4D_{n/2} & 4D_{n/2} + 2U_{n/2} \\ 4D_{n/2} + 3U_{n/2} & 4D_{n/2} + U_{n/2} \end{bmatrix}_{n \times n}$ onde U_n é uma matriz $n \times n$ em que todos os seus elementos são iguais a 1. Por exemplo,

$$D_4 = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}.$$

Para uma imagem com resolução espacial $r \times s$, temos:

```
for (i = 0; i <= r; i++)
  for (j = 0; j <= s; j++) {
    if (f[i][j] >= D[i % n][j % n])
      f[i][j] = c1;
    else
      f[i][j] = c0;
  }
```

No fragmento anterior, i%m e j%n são os restos das divisões de i e j por m e n, respectivamente.

5 Utilizando o Thumbs para quantificar imagens

Nesta seção descreveremos brevemente como usar o Thumbs (versão 3.20) para efetuar operações de quantização com imagens.

Quando o Thumbs estiver sendo executado, aparece na tela uma lista de imagens disponíveis em disco (veja figura 1). Selecione uma das imagens pressionando duas vezes o *mouse* no *thumbnail* ("'cópia em miniatura") da imagem.

Quando a imagem selecionada estiver na tela em tamanho normal, escolha o item "Image" mostrado na parte superior da tela. Assim, temos acesso a algumas operações básicas tais como: calcular histograma de cor (opção "Histogram"), girar, reduzir ou ampliar a imagem ("Rotate and Resize"), inverter cores ("Invert" ou "Swap Red and Blue"), bem como quantizar a imagem (opção "Depth"). Veja a figura 2.

Na opção "Image/Depth" do Thumbs podemos selecionar o número de cores, o limiar e a técnica a ser utilizada. Veja as figuras 3 e 4.

A figura 5 mostra um resultado obtido com o algoritmo de Floyd-Steinberg e a figura 6 mostra o histograma de cor.

Referências

[1] D. Rogers, *Procedural elements for Computer Graphics*, McGraw Hill, 1985.

[2] J. Gomes & L. Velho, *Computação Gráfica*, vol. 1, SBM, 1998.

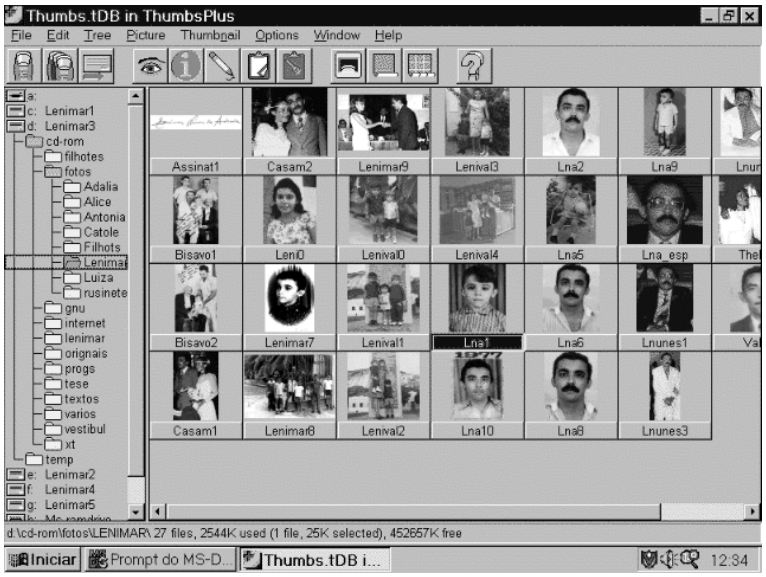

Figura 1: Tela inicial do Thumbs

Figura 2: Menu "Image" do Thumbs selecionado

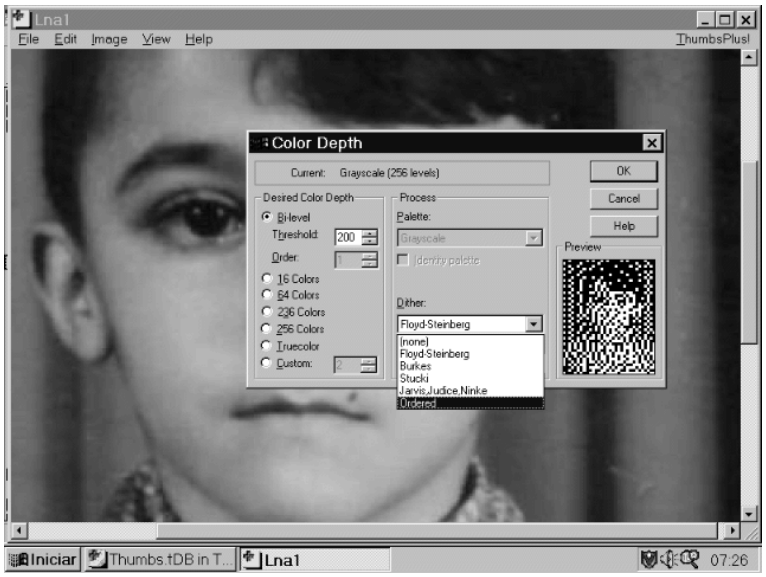

Figura 3: Opção Image/Depth do Thumbs

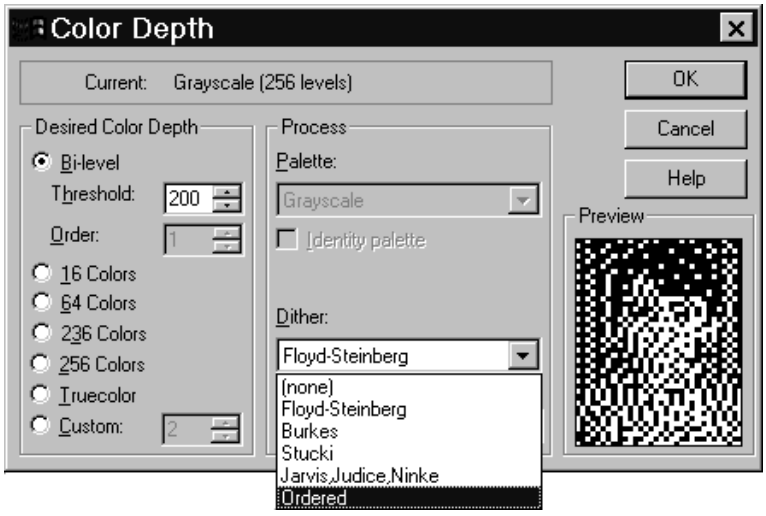

Figura 4: Opção Image/Depth do Thumbs

Figura 5: Algoritmo de Floyd-Steinberg

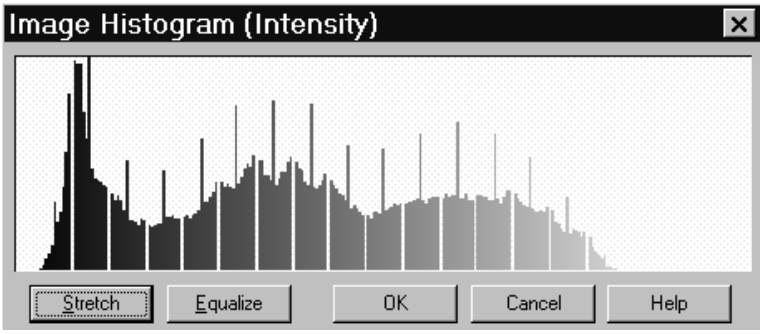

Figura 6: Histograma de cor