

Implementação de um Sistema de Fusão de Imagens no Modelo HSI

Alexandre Scheidt, Gustavo R. Krüger e Yuri R. Guimarães

27 de novembro de 2009

O que será apresentado

- Introdução:
 - Problemática;
 - O Modelo HSI;
 - Proposta;
- Implementação do sistema:
 - Algoritmo de conversão de RGB para HSI;
 - Algoritmo de fusão de imagens HSI;
- Considerações sobre a implementação;
- Resultados.

Problemática

- O problema envolve a combinação de diferentes elementos entre duas imagens idênticas;
- Neste caso específico, o problema combina elementos de cores de uma imagem de qualidade baixa com elementos de intensidade de uma imagem, colorida ou não, de qualidade alta, resultando em uma imagem colorida com boa qualidade;
- Para realizar a segmentação destes elementos em uma imagem, é necessário a utilização de um modelo de representação que a permita;
- O modelo HSI se mostrou mais apropriado pois os elementos de cor são menos relacionados com a intensidade de luz (brilho) da imagem do que em outros modelos de representação, como o RGB.

O Modelo HSI

- Neste modelo, a representação da cor trabalha com os conceitos de tonalidade, saturação e luminância:
 - Tonalidade ou Matiz (H): Do inglês *Hue*, a tonalidade ou matiz representa a cor propriamente dita;
 - Saturação (S): Do inglês *Saturation*, a saturação refere-se à pureza da cor, sendo inversamente proporcional à quantidade de cor branca misturada a uma tonalidade. Isto é, quanto maior a quantidade de cor branca misturada a uma outra cor, menor será a sua saturação (pureza);
 - Luminância ou Intensidade(I): Do inglês *Intensity*. A luminância ou intensidade está relacionada com a quantidade de luz (brilho) da cor;
- As vantagens desse modelo de representação da cor encontra-se, principalmente, na possibilidade de separar a intensidade da informação tonalidade e saturação.

Proposta

- Implementar um sistema que:
 - Captura imagens representadas pelo modelo RGB;
 - Realiza a conversão dos modelos RGB para HSI;
 - Realiza a fusão de uma imagem que conserva as informações HS com outra uma imagem que conserva a informação I, resultando em uma nova imagem HSI.

Implementação do Sistema

- A linguagem utilizada para a implementação foi JAVA;
- Para a resolução do problema alvo, o sistema implementa dois algoritmos:
 - Algoritmo de conversão de RGB para HSI;
 - Algoritmo de fusão de imagens HSI;
- Para melhor compreensão dos algoritmos apresentados a seguir, cada elemento R, G, B, H, S e I foi representado computacionalmente por uma matriz.

Algoritmo de conversão de RGB para HSI

- O algoritmo de conversão de RGB para HSI tomou por base o seguinte modelo matemático:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}.$$

$$h = \cos^{-1} \left\{ \frac{0.5 \cdot [(r-g) + (r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{1/2}} \right\} \quad h \in [0, \pi] \text{ for } b \leq g$$

$$h = 2\pi - \cos^{-1} \left\{ \frac{0.5 \cdot [(r-g) + (r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{1/2}} \right\} \quad h \in [\pi, 2\pi] \text{ for } b > g$$

$$s = 1 - 3 \cdot \min(r, g, b); \quad s \in [0, 1]$$

$$i = (R + G + B) / (3 \cdot 255); \quad i \in [0, 1].$$

Figura: Modelo Matemático da Conversão do Sistema RGB para o Sistema HSI

Algoritmo de conversão de RGB para HSI

- A partir do modelo anterior, o seguinte algoritmo foi implementado:

```
if (r == maior) {  
    if (medio == g) {  
        Base = 0;  
    } else {  
        Base = 5.0 / 6.0;  
        deslocamento = 1/6 - deslocamento;  
    }  
} else {  
    if (g == maior) {  
        if (medio == b) {  
            Base = 2.0 / 6.0;  
        } else {  
            Base = 1.0 / 6.0;  
            deslocamento = 1/6 - deslocamento;  
        }  
    } else {  
        if (medio == r) {  
            Base = 4.0 / 6.0;  
        } else {  
            Base = 3.0 / 6.0;  
            deslocamento = 1/6 - deslocamento;  
        }  
    }  
}  
h = Base + deslocamento;
```

Algoritmo de conversão de RGB para HSI

```
l = maior / 255;  
s = 0;  
h = 0;  
  
if (l == 0 || maior == menor) {   Imagem em escala de cinza  
    s = 0;  
    h = 0;  
} else {  
    s = (l - menor / 255) / l;     pela formula  
}
```

Figura: Algoritmo que implementa a conversão das componentes I e S

Algoritmo de conversão de HSI para RGB

- Para realizar a persistência da imagem em disco, realizou-se o processo inverso com base no seguinte modelo matemático:

$$B = I(1 - S)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R + B)$$

Para $0 \leq H < 120$

$$R = I(1 - S)$$

$$G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 3I - (R + B)$$

Para $120 \leq H < 240$

$$G = I(1 - S)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$R = 3I - (R + B)$$

Para $240 \leq H < 360$

Figura: Conversão do Modelo HSI para o Modelo RGB

Algoritmo de fusão de imagens HSI

- O algoritmo possui 3 entradas:
 - Informações H e S da primeira imagem;
 - Informação I da segunda imagem;
 - Escalar que representa o tamanho da imagem resultante;
- A partir deste conjunto de entrada, o algoritmo funciona, de modo geral, da seguinte forma:
 - Para cada posição (píxel) das matrizes H, S e I da imagem resultante, busca-se as posições correspondentes nas matrizes H, S e I de entrada e atribui-se o valor encontrado.

Mapeando píxels em imagens de diferente resolução

- Normalmente, o problema envolve duas imagens de entrada de tamanho diferente. Desta forma, se o tamanho de uma das imagens de entrada for diferente do tamanho da imagem resultante, não existiria a correspondência de 1 píxel da imagem resultante para cada píxel da imagem de entrada;
- Nestes casos, optou-se por interpolar valores para píxels da imagem resultante que não possuam um píxel correspondente nas imagens de entrada;

Mapeando píxels em imagens de diferente resolução

- Como exemplo:
 - Considere que uma das imagens de entrada tenha 200 píxels de largura e 100 de altura; e a imagem resultante tenha 400 píxels de largura e 300 de altura;
 - Considere agora a posição (30, 20) da matriz I da imagem de entrada;
 - Dividindo a coluna e a linha pela largura e altura, respectivamente, tem-se os valores: (0.075, 0.066);
 - Estes valores são multiplicados pela respectivas largura e altura da imagem de entrada: $(0.075 * 200, 0.066 * 100) = (15, 6.6)$;
 - Este resultado representa a posição da imagem de entrada cujo valor deveria ser atribuído à posição (30, 20) da imagem resultante, no entanto, o índice 6.6 não existe. Portanto, o valor desta posição deve ser interpolado com base nas quatro posições vizinhas.

Mapeando píxels em imagens de diferente resolução

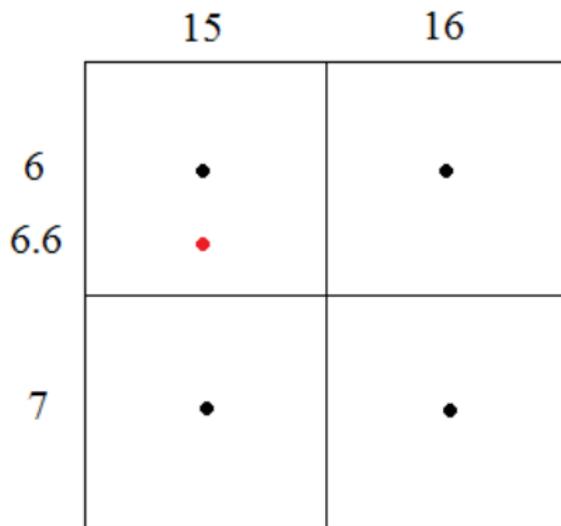


Figura: Representação do exemplo anterior

Mapeando píxels em imagens de diferente resolução

- De forma geral, a equação que calcula o valor de uma determinada posição da imagem resultante é:

$$P_r = \frac{\frac{P_1}{\text{dist}(P_x, P_1)} + \frac{P_2}{\text{dist}(P_x, P_2)} + \frac{P_3}{\text{dist}(P_x, P_3)} + \frac{P_4}{\text{dist}(P_x, P_4)}}{\frac{1}{\text{dist}(P_x, P_1)} + \frac{1}{\text{dist}(P_x, P_2)} + \frac{1}{\text{dist}(P_x, P_3)} + \frac{1}{\text{dist}(P_x, P_4)}}$$

- O resultado desta equação representa uma média ponderada dos valores dos quatro vizinhos.

Considerações sobre a implementação

- Foram implementados dois métodos para a Fusão de imagens:
 - Método por replicação de píxel;
 - Método por interpolação;
- Considerou-se o método por interpolação por obter melhores resultados;
- A equipe encontrou dificuldades ao trabalhar com imagens com grande resolução, fato que foi parcialmente solucionado.

Resultados

- Os resultados podem ser melhor visualizados com a execução do sistema.

Bibliografia

Gonzalez, R. C.; Woods, R. E. Processamento de imagens digitais. São Paulo: Ed. Edgard Blücher, 2000. 509p.

Aboud, S. R. Luciano Vieira Dutra, L. V. Prado, B. R. Segmentação de imagens multiespectrais utilizando-se o atributo matiz. XIV Simpósio Brasileiro de Sensoriamento Remoto, Natal, Brasil, 25-30 abril 2009, INPE, p. 6719-6726.

FIM