Tô Màu

Tài liệu hướng dẫn sinh viên xây dựng chương trình tô màu hình học cơ bản

Bộ môn **[Thị giác máy tính & Robot]**
Khoa Công nghệ thông tin
ĐH Khoa học tự nhiên TP HCM

MỤC LỤC

1. Tóm tắt lí thuyết 1
2. Case Study 1
3. Hướng dẫn cụ thể 1
4. Bài tập 2
5. Bài tập về nhà 3
6. Tài liệu tham khảo 3

TÔ MÀU

1. Tóm tắt lí thuyết

Cài đặt các thuật toán tô màu hình học theo đường biên và loang lân cận

2. Case Study

Thuật toán tô màu dựa theo đường biên

3. Hướng dẫn cụ thể

Thuật toán tô màu dựa trên đường biên sử dụng phương pháp tô 4 điểm lân cận

```
void BoundaryFill (int x, int y, RGBColor F_Color, RGBColor B_Color)
{
    RGBColor currentColor;

    currentColor = GetPixel(x, y);

    if(!IsSameColor(currentColor, B_Color) && !IsSameColor(currentColor, F_Color))
    {
        PutPixel(x, y, F_Color);
        FillLeft(x - 1, y, F_Color, B_Color);
        FillTop(x, y + 1, F_Color, B_Color);
        FillRight(x + 1, y, F_Color, B_Color);
        FillBottom(x, y - 1, F_Color, B_Color);
    }
}
```

Trong đó RGBColor là một struct tự định nghĩa cho ba thành phần màu (R,G,B) như sau:

```
typedef struct _RGBColor
{
    unsigned char r;
    unsigned char g;
    unsigned char b;
}RGBColor;
```

Hàm IsSameColor tự cài đặt, thực hiện việc so sánh xem hai màu có giống nhau (cả ba thành phần màu) hay không.

Các hàm FillLeft, FillRight, FillTop, FillBottom cài đặt tương tự như hàm BoundaryFill.

Lưu ý: Do hàm glReadPixels sử dụng hệ tọa độ có chiều dương của trục y hướng lên nên ta cần phải truyền vào h – y (tọa độ x, y truyền vào nằm trong hệ tọa độ màn hình với chiều dương trục y hướng xuống), h là chiều cao của cửa sổ chương trình.

Lấy và gán giá trị màu của điểm ảnh

```
RGBColor GetPixel(int x, int y)
{
    unsigned char * ptr = new unsigned char [3];

    glReadPixels(x, h -  y, 1, 1, GL_RGB, GL_UNSIGNED_BYTE, ptr);

    RGBColor color;
    color.r = ptr[0];
    color.g = ptr[1];
    color.b = ptr[2];

    return color;
}
```

Ghi chú: Đối với SharpGL

```
gl.ReadPixels(x, y, width, height, format, type, pixelData); // Format =
OpenGL.GL_RGB, type = GL_BYTE, pixelData = new byte[3]
```

Thư viện OpenGL có hỗ trợ hàm gán giá trị màu của pixel tại vị trí (x, y). Trong các bài hướng dẫn tuần trước, để vẽ một pixel ta sử dung hàm glVertex đặt giữa glBegin(GL_POINTS) và glEnd. Ngoài cách làm này, ta có thể cài đặt một hàm PutPixel sử dụng hàm glDrawPixels của thư viện OpenGL như sau

```
void PutPixel(int x, int y, RGBColor color)
{
    unsigned char * ptr = new unsigned char [3];
    ptr[0] = color.r;
    ptr[1] = color.g;
    ptr[2] = color.b;

    glRasterPos2i(x, y);
    glDrawPixels(1, 1, GL_RGB, GL_UNSIGNED_BYTE, ptr);

    glFlush();
}
```

Ghi chú: Đối với SharpGL

```
gl.DrawPixels(int width, int height, uint format, byte[] pixels);
```

4. Bài tập

Hoàn thiện thuật toán tô màu đã trình bày trên

5. Bài tập về nhà

Phần tự nghiên cứu cài đặt:

- Thuật toán tô màu Loang: Flood Fill.
- Thuật toán tô màu theo hàng: ScanLine

6. Tài liệu tham khảo

- gear.kku.ac.th/~rujchai/files/CG471-07.ppt
- Mở rộng: http://www.codeproject.com/KB/GDI/QuickFill.aspx