

## EA978 – Lista 7 – Rasterização

1. Seja uma área de desenho de  $64 \times 64$  *pixels*, preenchida com a cor preta, e um segmento  $P_0P_1$ , onde  $P_0 = (10, 50)$  e  $P_1 = (45, 4)$ . Considere ainda que a cor atribuída a  $P_0$  seja  $RGB_0 = (1, 0, 1)$  e a cor a  $P_1$ ,  $RGB_1 = (1, 1, 0)$ . Rasterize o segmento com as cores interpoladas linearmente, com uso do

- (a) algoritmo DDA
- (b) algoritmo de ponto médio

Compare os dois algoritmos, quanto aos resultados, tipo de operações e quantidade de operações envolvidas,

2. Dados 2 triângulos:  $((1, 50, 10, 1), (12, 1, 4, 1), (57, 49, 10, 1))$  e  $((15, 4, 5, 1), (63, 2, 15, 1), (11, 58, 12, 1))$ . O primeiro é azul e o segundo é vermelho. Considere ainda que o observador esteja olhando na direção do semi-eixo  $z$  positivo.

- (a) Rasterize os 2 triângulos, com uso do algoritmo de *scan-line* com *z-buffer*. Mostre explicitamente os passos do procedimento.
- (b) O algoritmo de *scan-line* é eficiente porque explora alguns tipos de coerência para reduzir o número de operações. Indique dois tipos de coerência explorados no seu procedimento.
- (c) Para tirar melhor proveito das coerências, é necessário pré-processar os dados, estruturando-os de forma mais apropriada. Como é esta estruturação no algoritmo de *scan-line*?

3. Como se representa uma imagem no domínio espectral a partir da sua contrapartida no domínio espacial? Visualmente qual tipo de análise podemos fazer com uso de imagens representadas no domínio espectral? Qual tipo de melhoramento podemos obter trabalhando com as imagens no domínio espectral?

4. Dados os valores, entre 0 e 255, das amostras de uma imagem em tons de cinza gerada pelo procedimento

```
int i, j, c;

for (i = 0; i < 64; i++) {
    for (j = 0; j < 64; j++) {
        c = 155 * sin((i*M_PI)/10) * cos ((j*M_PI)/5) + 150;
        if (c < 0) c = 0;
        else if (c > 255) c = 255;
        imagem[i][j] = (GLubyte) c;
    }
}
```

- Represente o histograma desta imagem. É uma imagem clara ou escura? Justifique.
- Determine os 16 níveis de quantização, utilizando
  - quantização uniforme

- algoritmo de populosidade
  - algoritmo de corte mediano
  - Particione os valores de tons de cinza em 16 células de quantização, com base nos 16 níveis de quantização obtidos com o algoritmo de populosidade.
  - Com somente dois valores, 0 e 255, quantize esta imagem com aparência de 16 níveis de cinza com uso da
    - técnica de *dither* de Bayer
    - técnica de difusão de erro Floyd-Steinberg
5. OpenGL inclui na sua arquitetura o fluxo de processamento de pixels. Através de `glReadPixels` você pode ler os valores de cada pixel nos *buffers*. Há dois comandos `glHistogram` e `glGetHistogram` que permitem você obter os valores dos *pixels* de uma imagem, quando devidamente habilitadas. Os comandos `glMinmax` e `glGetMinmax` fornecem os valores máximo e mínimo atribuídos aos *pixels* de uma imagem. Utilize estes comandos para “fazer uma estatística” de uma das imagens no último item da lista 5.
6. Qual é a técnica básica de imageamento no Povray (<http://www.povray.org>)?