

Universidade Federal de Santa Maria
Departamento de Eletrônica e Computação
Prof. Cesar Tadeu Pozzer
Disciplina: Computação Gráfica
pozzer@inf.ufsm.br
01/08/2008

Exercícios

Computação Gráfica

Assuntos:

1. Fundamentos
2. Cor
3. Imagem
4. Fundamentos Matemáticos
5. CG 2D
6. Curvas 2D
7. CG 3D
8. Síntese de Imagens
9. OpenGL Básico

1 Fundamentos

1. Defina os 4 universos para o seguinte problema: criar um sistema para projetar cortes de chapas metálicas (em forma poligonal) no computador.
2. Compare o universo matemático e de representação entre um terreno e uma imagem. Descreva como pode ser o universo de implementação para o exemplo do terreno.
3. Compare as tecnologias utilizadas na construção de um monitor CRT com um monitor LCD. Comente vantagens e desvantagens dos dois tipos monitores.
4. Pesquise sobre o funcionamento das impressoras matricial, laser e jato de tinta.
5. Faça um histórico da evolução das placas de vídeo em relação à memória, recursos e desempenho. Faça um quadro comparativo das placas vendidas atualmente.

2 Cor

- 1) Comente sobre os tipos de cores primárias e aplicações práticas para cada conjunto, justificando o uso.
Qual o cálculo de conversão entre elas?
- 2) O que é uma cor monocromática?
- 3) O que é o balanço de branco (*White ballance*) de máquinas fotográficas digitais? Porque é necessário?
- 4) Qual a diferença entre cores primárias aditivas e subtrativas?
- 5) O que é o *gamut* de cores de um sistema?
- 6) É possível representar computacionalmente qualquer cor presente no mundo real com precisão absoluta?
Comente como isso pode ser feito caso possível ou em que fases da conversão que podem ser introduzidos erros.
- 7) Como visto em aula, a luz ao atravessar um prisma (ou gota de água) sobre refração e por isso há uma separação das cores espectrais (pelo comprimento de onda), formando o arco íris. Em uma máquina fotográfica isso também ocorre quando a luz passa pelas lentes?
- 8) Implemente uma interface gráfica interativa para escolha de cores do modelo HSL (semelhante a vista em aula).

- 9) Como você faria para codificar cores no formato RGB em 8 bits por pixel? Justifique.

3 Imagem

- 1) O que é resolução de uma imagem? Quais parâmetros definem a resolução?
- 2) Em qual resolução em pixels (vertical e horizontal) pode ser impressa uma imagem digital com 4 MPixel em proporção 10x15 ?
- 3) Em qual resolução em DPI pode ser impressa uma imagem digital com 10 MPixel em proporção 10x15 com tamanho 20x30cm?
- 4) Uma fotografia digital tirada em 6 MPixel tem o dobro de resolução que uma foto em 3MPixel. Isso é verdadeiro? Justifique.
- 5) Implemente uma função para carregar uma imagem no formato BMP. Utilize a Canvas2D para visualizar a imagem. Faça o mesmo para os formatos PCX, GIF, TIFF e JPEG.
- 6) Explique o que é amostragem de um sinal. Porque uma imagem digital deve ser amostrada? Justifique. Compare uma imagem digital com um mapa de alturas (*height field*).
- 7) O que significa quantizar um sinal. Para que serve a quantização na compressão do formato JPEG? Que efeito ela tem na imagem gerada? Comente em detalhes.
- 8) Qual a diferença entre compactação e compressão de imagens. Comente em detalhes 3 formas de compressão de imagem e compare os resultados das técnicas.
- 9) Explique o funcionamento da transformada cosseno.
- 10) O que são funções ortogonais? Encontre uma forma de mostrar que as funções da DCT são ortogonais e que aquela fórmula é genérica para qualquer número de funções ortogonais.
- 11) O que significa dizer que uma imagem tem altas freqüências? Como isso é caracterizado na imagem? O que acontece se as freqüências mais altas forem removidas da imagem? E o que acontece se forem removidas as baixas freqüências?
- 12) Compare a diferença de freqüência em um sinal de áudio com um sinal de vídeo (imagem).
- 13) Para processar a DCT bidimensional utiliza-se 64 funções base. Pode-se dizer que estas funções são LI? Qual a relação com uma base do espaço euclidiano R^3 ? Comente em detalhes.
- 14) Explique o que é a codificação de entropia no formato JPEG? Porque é processado em zig-zag?
- 15) Avalie por que a compressão JPEG utiliza uma matriz de 8x8 coeficientes. Que diferença teria se a matriz fosse 4x4, ou 2x2 ou 1x1 ou 16x16, por exemplo?
- 16) Todas máquinas fotográficas digitais apresentam zoom digital. O que se pode dizer em relação à freqüência de uma imagem que passou por este processo? E em relação à resolução?
- 17) Aplicando-se a DCT sobre uma imagem, retiram-se (zera-se) todas as baixas freqüências (coeficientes). Na seqüência aplica-se a IDCT. Aplicando-se novamente a DCT sobre a imagem resultante, existirão coeficientes não nulos em baixas freqüências? Comente.
- 18) O que acontece com uma imagem JPG caso for aberta e salva várias vezes? E se fosse uma imagem BMP?
- 19) Implemente a transformada cosseno unidimensional para um vetor com um número variado de amostras múltiplas de dois.

4 Fundamentos Matemáticos

- 1) Verifique se a base $e_1 = (1,2,1)$, $e_2 = (2,9,0)$ e $e_3 = (3,3,4)$ é uma base do R^3 .
- 2) Seja $S = \{e_1, e_2, e_3\}$ a base do exercício anterior. a) Encontre o vetor de coordenadas $v = (5, -1, 9)$ em relação à S .
- 3) Em que ordem deve-se multiplicar um ponto no espaço R^n por uma matriz de transformação de dimensão n ? (o mesmo vale para a concatenação de matrizes de transformação) Comente sobre a teoria envolvida em relação à multiplicação de matrizes.
- 4) O que é e para que é usado em computação gráfica coordenadas homogêneas? Dê um exemplo.
- 5) Dado um vetor no R^2 qualquer, determine dois vetores perpendiculares a este vetor.
- 6) Dado um vetor no R^3 , quantos vetores perpendiculares a ele existem?
- 7) Comente vantagens da utilização de vetores unitários

- 8) Considerando-se que um carro de corrida está na posição P(10,-40), indo para a posição P(-70, 0). Escreva um programa em C++ para definir os vetores posição e direção de deslocamento do veículo e um valor escalar que representa a velocidade do veículo. Definir e inicializar a classe carro com os parâmetros. O veículo deve andar duas unidades a cada passo até chegar ao destino. Quanto chegar ao destino, ele deve permanecer parado. Sempre que possível, as operações devem ser sobre vetores.
- 9) Considerando-se que um carro de corrida está na posição P(-10,-20), indo para a posição P(30, 10). Escreva um programa em C++ para fazer o veículo chegar até a posição destino após N loops de renderização. O veículo não tem aceleração. Defina uma classe e métodos para a entidade carro. Defina os vetores posição e direção, e como escalar a velocidade deste veículo. Use conceitos matemáticos de vetores para representar as grandezas.
- 10) Um tanque de guerra esta na posição (x_1, y_1) e mirando na posição (x_2, y_2) . Determine quantos graus e em que direção o tanque deve girar para mirar no alvo na posição (x_3, y_3) . Após escreva um algoritmo para mover o tanque, em n incrementos de posição, até a posição intermediária entre (x_1, y_1) e (x_3, y_3) .

5 Computação Gráfica 2D

- 1) Implemente um programa gráfico que faz uso das primitivas: ponto, linha e retângulo. Use laços de repetição para gerar figuras mais complexas, com preenchimento, fazendo uso de cores diversas. Use funções de *delay* para simular pequenas animações de linhas ou qualquer outro elemento.
- 2) Implemente função paramétrica para desenho de círculos. A função deve receber o raio e a posição onde desenhar o círculo. Defina uma equação para determinar qual deve ser o incremento angular para que o círculo gerado não fique com falhas.
- 3) Implemente uma função que gera poliedros de n lados, circunscritos a um círculo de raio r , onde n e r são fornecidos pelo usuário.
- 4) Implemente uma função para gerar discos (círculos preenchidos) usando somente a primitiva círculo. A função deve receber o raio do disco a ser gerado. Observe para não deixar pixels sem preenchimento no interior do disco.
- 5) Proponha um algoritmo para encontrar o fecho convexo de um grupo de pontos gerados aleatoriamente.
- 6) Implemente um programa para fazer a visualização gráfica de árvore binária de busca.
- 7) Implemente uma rotina que lê um arquivo texto, composto de duas colunas: x e y , de valores inteiros. Cada linha da coluna representa um ponto P . Plote um gráfico na tela que faz a união de cada ponto P consecutivo do arquivo, gerando um gráfico de linha. Aplique escalas nos dados para que o gráfico caiba na tela.
- 8) Implemente função para fazer a geração de gráficos em barra, linha e de pizza (semelhante ao Excel) a partir de dados lidos de um arquivo.
- 9) Usando a primitiva linha, faça uma função para simular um tecido trama.
- 10) Escreva um algoritmo para desenhar uma espiral fazendo uso de coordenadas polares e por função implícita.
- 11) O que é uma transformação afim. De exemplos e justifique. De um contra-exemplo.
- 12) Implemente o algoritmo de *flood-fill* para preenchimento de figuras côncavas e/ou convexas geradas aleatoriamente.
- 13) Implemente um algoritmo para desenho de retas utilizando somente a primitiva pixel.
- 14) Implemente funções para plotar funções trigonométricas sin, cos e tangente.
- 15) Faça um programa que mostra um relógio digital e um analógico a partir da hora do sistema. Deverá mostrar hora:mim:seg. A escolha entre analógico e digital deve ser feita pelo usuário. As horas no relógio digital devem ser desenhadas usando linhas.
- 16) Implemente um programa que simula uma roda gigante em movimento. Deve-se estipular o número de assentos que ela vai ter e a velocidade de rotação.
- 17) Implemente um programa que simula uma bicicleta em movimento. Deve estar animados as rodas, o pedal e as pernas do indivíduo.
- 18) Faça um função para geração de engrenagens. Deve ter como parâmetro o número de dentes, raio e comprimento dos dentes. A partir desta função faça um programa que simula a rotação de diversas engrenagens conectadas com tamanhos variados. Deve-se controlar o sincronismo de rotação entre as engrenagens em função do número de dentes ou raio.

- 19) Função para desenhar uma linha com setas nas extremidades. A linha deve ser definida por dois pontos e a função deve ter parâmetros também para especificar a largura e comprimento da seta.
- 20) Plote na tela as seguintes séries de Fourier. Faça uma implementação que permita estipular a quantidade de termos da série a serem somados, e verifique graficamente como o número de termos influencia na qualidade do gráfico gerado. Faça também ajustes nas escalas do gráfico.

$$y = \frac{4}{\pi} \left(\frac{\sin x}{1} + \frac{\sin 3x}{3} + \frac{\sin 5x}{5} + \dots \right)$$

$$y = \frac{2}{\pi} - \frac{4}{\pi} \left(\frac{\cos 2x}{1*3} + \frac{\cos 4x}{3*5} + \frac{\cos 6x}{5*7} + \dots \right)$$

$$y = \frac{\pi^2}{3} - 4 \left(\frac{\cos x}{1^2} - \frac{\cos 2x}{2^2} + \frac{\cos 3x}{3^2} - \dots \right)$$

- 21) Implemente um editor de figuras vetoriais semelhante ao Word, onde pode-se fazer a inserção e redimensionamento de figuras vetoriais.
- 22) Simule o deslocamento e movimento de pistão hidráulico conectado a um braço mecânico (como ocorre em escavadeiras).

6 Curvas 2D

- 1) Implemente função para plotar as *blending functions* das curvas de Bézier, Hermite e B-Spline. Para Bézier, utilize polinômios de grau 2,...,N.
- 2) Qual a limitação do uso de polinômios de grau 2 na representação de curvas em um espaço 3D?
- 3) Descreva matematicamente a relação para assegurar continuidade C^2 na junção de dois segmentos de curvas de Bézier.
- 4) Tanto em Bézier como Hermite a curva passa pelo primeiro e último ponto de controle. Comente a diferença que os pontos intermediários têm na geração da curva.
- 5) Faça uma função que recebe um conjunto de pontos de controle e retorne um conjunto de segmentos de reta que aproximem a curva B-Spline que passa pelos pontos de controle.
- 6) Função para desenhar uma espiral utilizando curvas B-Spline. Compare o desempenho desta função em relação a uma implementação de por meio de funções paramétricas.
- 7) Descubra qual o grau máximo que podem ser utilizados os polinômios de Bézier antes de gerar instabilidades numéricas ou falta de precisão de ponto flutuante. Teste para variáveis do tipo float e double.

7 Computação Gráfica 3D

1. Implemente uma câmera sintética com 6 graus de liberdade. A câmera deve oferecer opções de realizar projeção em perspectiva com 1, 2 ou 3 pontos de fuga, bem como projeção ortográfica. Esta câmera deve ser utilizada juntamente com os demais exercícios desta seção.
2. Implemente uma função para gerar um tetraedro a partir de um triângulo equilátero. Gere uma animação de transição. Faça o mesmo para um cuboctaedro. Para cada face gerada, gere um vetor normal que também deve ser visualizado. Veja outros poliedros mais complexos em:
 - a. http://pt.wikipedia.org/wiki/S%C3%B3lidos_de_Arquimedes,
 - b. http://pt.wikipedia.org/wiki/S%C3%B3lidos_de_Catalan
 - c. http://pt.wikipedia.org/wiki/Poliedros_de_Kepler-Poinsot.
3. Implemente uma função para gerar um icosaedro (objeto com 20 triângulos equiláteros) visualizado em estrutura de *wireframe*. Expanda este algoritmo para subdivisão sucessiva de cada lado por outros 4 triângulos. Aplique este processo de subdivisão também sobre o tetraedro.
4. Implemente um algoritmo para fazer a remoção de faces ocultas de um poliedro qualquer. Visualize este objeto em *wireframe* para testar a eficácia do algoritmo.
5. Visualize um braço robótico 3D, com articulações nas juntas. Deve permitir rotação no eixo y apenas no braço conectado à base.

6. Fazendo uso da técnica de *sweep*, visualize com a técnica de wireframe:
 - a. Um fio de luz transado com fio no centro
 - b. Uma escadaria em formato de caracol
 - c. Uma mola em formatos espiral
 - d. Uma taça de vinho
 - e. Uma argola
7. Crie uma malha poligonal (de triângulos) a partir de um algoritmo de geração de esfera.
8. Crie um terreno a partir de uma imagem. A cor de cada pixel (ou luminância) deve determinar a altura (coordenada y) de cada ponto do terreno.

8 Síntese de Imagens

- 1) Implementar a técnica de *bump mapping* para controle da reflexão difusa sobre uma superfície plana. Utilize funções trigonométricas como *bump function*.
- 2) Aplique a iluminação difusa e especular (*Phong model*) sobre uma superfície plana. A posição da fonte luminosa e coeficientes de reflexão difusa e especular podem ser modificadas durante a visualização.
- 3) Implemente um algoritmo de Ray-tracing para fazer a visualização de uma esfera.
- 4) Pinte as curvas de decaimento da reflexão especular do modelo de reflexão Phong para diferentes valores de n ($\cos^n \phi$).

9 OpenGL

- 1) Modele em OpenGL um piano de cauda, modelado por meio de curvas de Bézier. Utilize uma malha bem refinada para melhorar o processo de iluminação. Deve-se poder simular o movimento de teclas. Utilize a pilha de transformação para fazer a disposição das partes do piano.
- 2)