Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso Superior de Tecnologia em Sistemas de Computação
Disciplina: Computação Gráfica
AD1 1° semestre de 2017.

1) Descreva um problema sócio-econômico que pode usufruir de métodos estudados na Computação Gráfica (1.0 ponto).

R: O desaparecimento de crianças e adolescentes é um problema que aflige a sociedade diariamente [1]. A Computação Gráfica tem auxiliado os órgãos públicos na solução deste problema através de ferramentas para envelhecimento digital de fotos, que permite estimar como uma criança ou adolescente aparentará após alguns anos, tornando possível encontrar seu paradeiro.

Um exemplo de método que deu origem a um sistema de envelhecimento é o Illumination-Aware Age Progression (Progressão de Idade Sensível a Iluminação), proposto recentemente por Ira Kemelmacher-Shlizerman. O método utiliza inúmeras imagens disponíveis na internet para construir um subespaço de imagens médias. Cada imagem média descreve um protótipo de face de homem ou mulher de 1 a 80 anos de idade. De modo informal, pode-se dizer que o método aprende características que determinam o que seria um homem ou mulher de uma certa idade sob diferentes valores de iluminação, determinado as diferenças entre uma idade e outra. Assim, dada uma nova foto, o sistema acrescenta as diferenças, deformando a imagem para produzir sua versão em uma determinada idade [2].

Referências:

[1] http://www.ofluminense.com.br/pt-br/content/programa-simula-envelhecimento-de-crian%C3%A7as-desaparecidas (acessado em 22/02/2017 - 10:33 AM)

[2] Ira Kemelmacher-Shlizerman, Supasorn Suwajanakorn, and Steven M. Seitz. 2014. Illumination-Aware Age Progression. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR '14). IEEE Computer Society, Washington, DC, USA, 3334-3341. DOI: http://dx.doi.org/10.1109/CVPR.2014.426

2) Descreva a diferença entre um objeto 2D vetorial e um objeto 2D *raster*. Descreva duas aplicações (exemplos de software) que trabalham com cada um dos dois tipos de objetos (1.0 ponto).

R: Um objeto 2D *raster* é aquele definido através de uma decomposição do espaço, no qual o espaço ambiente é subdividido em células. As células ocupadas pelo objeto são identificadas através de uma função característica (função que retorna 1 caso ela seja ocupada e zero caso contrário). Já um objeto vetorial é especificado por vetores que determinam posições de pontos no espaço, que podem definir vértices de uma linha poligonal aberta (ou fechada) ou pontos de controle de curvas iterativas.

Um exemplo de software que usa objetos *raster* é GIMP [3]; já o um exemplo de software 2D vetorial é o Inkspace [4].

Enlaces:

[3] https://www.gimp.org/. Acessado em 22/02/2017 - 10:52 AM
[4] https://inkscape.org/en/. Acessado em 22/02/2017 - 10:52 AM

3) Suponha a existência de uma função *amostrar*(*t*), implementada em uma linguagem de programação tal que, para um dado parâmetro real t, $0<t\leq t_{max}$, retorne as coordenadas x e y de um ponto p(t), sobre uma curva paramétrica simples fechada 2D. Descreva um algoritmo que, usando *amostrar*(*t*), seja capaz de computar o comprimento da curva de modo aproximado (1.0 ponto). (Anulada)

Algoritmo- PlotarCurva(amostrar(t),delta,t_{max})

Entrada: amostrar(t) - função que amostra a função paramétrica
delta - intervalo de discretização do parâmetro t
t_{max} - limite superior do intervalo ($0<t\leq t_{max}$)
Saída: comprimento

comprimento ← 0
x_0,y_0 ← amostrar(0)
t ← delta
Enquanto $t<t_{max}$ faça
x_1,y_1← amostrar(t)
compAresta ← $\sqrt{(x1-x0)^2+(y1-y0)^2}$
comprimento ← comprimento + compAresta
t ← t + delta
x_0,y_0 ← x_1,y_1
Fim_Enquanto
Retornar comprimento

Anulada por especificação incompleta no enunciado

4) Explique a diferença entre uma curva planar e uma curva espacial. Descreva uma aplicação para cada tipo de curva (1.0 ponto).

R: Uma curva planar é um objeto gráfico unidimensional imerso em um ambiente bidimensional. Isto significa que cada um de seus pontos pode ser descrito por duas coordenadas x e y. Uma curva espacial é aquela que mora em um ambiente tridimensional tendo 3 coordenadas x,y e z.

5) Explique como uma superfície contínua pode ser representada de forma discreta (1.0 ponto).

R: Um modo de se representar uma superfície contínua é tomar um conjunto de amostras e estruturá-las de modo que ela possa ser reconstruída com um número finito de elementos. No caso de superfícies paramétricas, pode-se tomar um conjunto regular de amostras no domínio de parâmetros, aplicar a função que descreve a superfície sobre tais amostras e finalmente triangular o conjunto de pontos resultante. No caso de superfícies implícitas podemos amostrar o espaço envolvente usando decomposição espacial, obtendo assim um modelo voxelizado, sobre o qual, posteriormente, aplica-se um método de poligonização como, por exemplo, o Marching Cubes (ver questão 9).

6) Defina como calcular o plano tangente para uma superfície paramétrica e para uma superfície implícita (1.0 ponto).

No caso de uma superfície paramétrica $f:(u,v) \to (x(u,v), y(u,v), z(u,v))$, o vetor normal é dado pelo vetor unitário $\vec{n} = \dfrac{\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}}{\left|\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}\right|}$, onde $\frac{\partial f}{\partial u} = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right)$ e $\frac{\partial f}{\partial v} = \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}\right)$. A normal somente estará definida se $\frac{\partial f}{\partial u}$ e $\frac{\partial f}{\partial u}$ não forem colineares.

No caso de uma superfície implícita $f:(x,y,z) \to \Re$ a normal é dada por $\vec{n} = \dfrac{\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)}{\left|\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)\right|}$ desde que $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) \neq 0$.

Com base na normal $\vec{n}$, podemos definir o plano tangente em um ponto p através da equação $\vec{n}(x-p) = 0$, onde $p \in S$ e $x \in R^3$, onde S é a superfície em questão.

7) Explique porque na Computação Gráfica utiliza-se superfícies para as quais seja possível definir um plano tangente (1.0 ponto).

R: Em comutação gráfica a existência do plano tangente indica que localmente ela se comporta como um plano (espaço Euclidiano bidimensional). Além disso, neste caso é possível definir um vetor normal que determina a orientação local da superfície. Com efeito, como será visto mais tarde no curso, podemos calcular a intensidade de luz refletida em um ponto da superfície para o qual seja conhecido o vetor normal.

8) Que tipo de descrição geométrica é facilmente utilizável para descrever objetos que podem ser fabricados em impressoras 3D. Justifique sua afirmação. (1.0 ponto)

R: Objetos fabricáveis são sempre sólidos logo, modelos que descrevem sólidos, como os baseados em voxels e os baseados em representação por fronteira (B-rep), são os mais adequados. É fisicamente impossível construir objetos que não tenham volume, logo não é possível utilizar descrições baseadas em curvas ou superfícies não compactas (fechadas e limitadas).

9) Considere um objeto gráfico 3D representado por um conjunto de *voxels*. Descreva um método para obter uma representação da superfície que delimita fronteira do sólido correspondente (1.0 ponto).

R: Uma solução aproximada pode ser obtida considerando que o conjunto de voxels é uma discretização de uma função implícita $f:(x,y,z)\rightarrow\Re$. Deste modo, a cada voxel v_{ijk} associamos um valor negativo, por exemplo, igual a -1 se o voxel é ocupado pelo sólido e 1 caso contrário. Deste modo, o problema se resume a solucionar o problema de poligonização de superfícies implícitas, que pode ser resolvido, por exemplo, através do algoritmo Marching Cubes [5].

O Marching Cubes é um algoritmo porposto por Lorensen e Cline, que procura extrair uma superfície poligonal a partir modelos volumétricos. Uma superfície é construída ajustando-se um ou mais polígonos em cada voxel do modelo volumétrico que contenha parte da superfície. O algoritmo associa a cada voxel um número de 8 bits, onde cada bit associado a um único vértice, tem valor igual a 1, quando o vértice é interior a superfície, e zero quando ele é exterior. A configuração de polígonos no interior do voxel representando a superfície depende da configuração de valores de bits associada ao voxel. O algoritmo utiliza esse número para consultar uma tabela que mostrará a disposição dos vértices e faces geradas para cada configuração.

Para saber se uma superfície passa por uma aresta, basta que exista um vértice da aresta que esteja dentro do volume e outro que esteja fora. As 256 possíveis combinações geradas pelos vértices geram uma tabela de configurações de faces. Estas se resumem a 15 variações mostradas na figura abaixo, porém com rotações diferentes:

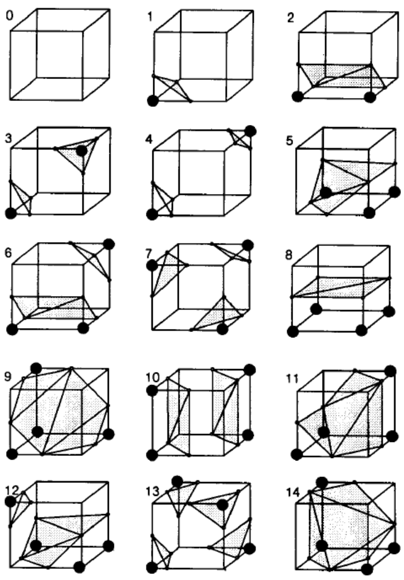


Figure 3. Triangulated Cubes.

Figura obtida de [5]

[5] William E. Lorensen, Harvey E. Cline: Marching Cubes: A high resolution 3D surface construction algorithm. In: Computer Graphics, Vol. 21, Nr. 4, July 1987

10) Faça uma pesquisa sobre L-Systems e como eles podem ser usados para definir padrões encontrados na natureza(1.0 ponto).

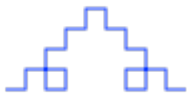
R: L-Systems ou sistemas L são gramáticas formais que consistem de um *alfabeto*, um conjunto de *regras de produção*, que permite formar cadeias válidas dentro da linguagem descrita pela gramática, e um *axioma* ou cadeia inicial. Além disso é preciso definir um mecanismo que converta as cadeias geradas em estruturas geométricas. Sistemas L estão muito utilizados para descrever estruturas fractais [6].

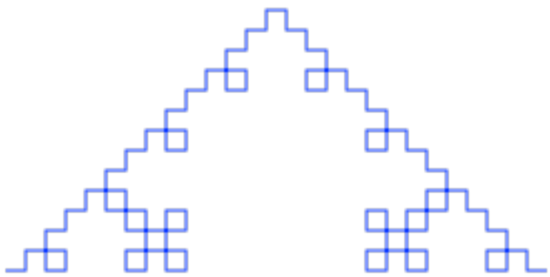
Um exemplo bem simples e de fácil compreensão, publicado em [6], que aqui reproduzimos para fins didáticos, é a curva de Koch. Observe que o *alfabeto* pode ser composto de *variáveis*, isto é símbolos que podem ser substituídos e *constantes*. No exemplo abaixo, a *variável* é F e as *constantes* descrevem operações de girar à esquerda e à direita de 90 graus.

variáveis : F
constantes : + −
axioma : F
regra de produção: (F → F+F−F−F+F)

$n = 0$:
F

$n = 1$:
F+F−F−F+F


$n = 2$:
F+F−F−F+F + F+F−F−F+F − F+F−F−F+F − F+F−F−F+F + F+F−F−F+F


$n = 3$:
F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F +
F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F −
F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F −
F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F +
F+F−F−F+F+F+F−F−F+F−F+F−F−F+F−F+F−F−F+F+F+F−F−F+F


Referência [6]: https://en.wikipedia.org/wiki/L-system. Acessado em 22-02-2017 às 12:04h.