

- 1) Descreva um problema do mundo real que pode ser resolvido através da Computação Gráfica (1.25 ponto).

A Arquitetura é uma das áreas que mais usufrui das técnicas de Computação Gráfica. Atualmente, segundo Lage et al., existe uma demanda para solucionar problemas relação a criação de projetos que valorizem um imóvel a ser construído, levando em conta as restrições físicas do processo de construção e também da qualidade da visibilidade do empreendimento. Este último aspecto tem impacto considerável na valorização do imóvel. Em um trabalho recente, Lage et al. [1] abordam tal problema através da criação de um arcabouço baseado em um catálogo, que permite a exploração iterativa de vários conceitos de design, com capacidade de ajuste de preferência de visibilidade. O método se baseia em medidas de visibilidade combinados com esquemas de indexação, para computação eficiente da visibilidade e navegação entre as possibilidades de design. Uma das grandes inovações é o entendimento do conjunto de modelos candidatos como um espaço multidimensional, onde cada ponto do espaço é uma possível solução de design associado a uma métrica de qualidade. A metodologia permite que tal espaço seja eficientemente explorado de forma interativa, com base no armazenamento das relações topológicas (esquema de vizinhanças) dos pontos do espaço.

Na figura abaixo pode-se ver diferentes opções de design em um catálogo, sua seleção e posterior inserção em um modelo 3D de uma cidade após seu refinamento. A figura mais a direita ilustra a renderização e combinação do modelo 3D com o cenário real.

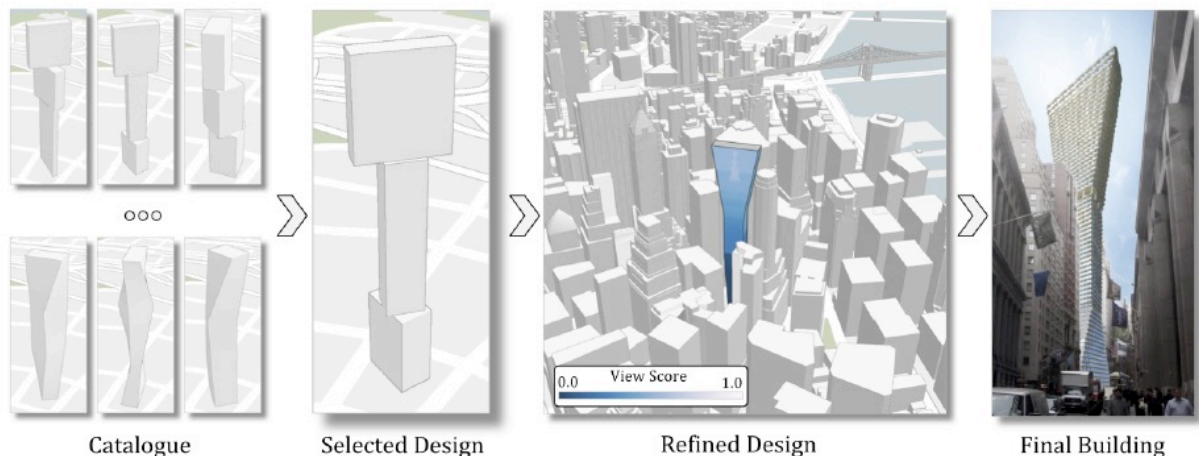


Figura obtida de [1] utilizada com permissão de Marcos Lage.

[1] Harish Doraiswamy, Nivan Ferreira, Marcos Lage, Huy Vo, Luc Wilson, Heidi Werner, Muchan Park, and Cláudio Silva. 2015. Topology-based catalogue exploration framework for identifying view-enhanced tower designs. *ACM Trans. Graph.* 34, 6, Article 230 (October 2015), 13 pages. DOI: <https://doi.org/10.1145/2816795.2818134>

- 2) Faça uma pesquisa sobre impressoras 3D e cite um problema real que pode usufruir de sua utilização (1.25 ponto).

Dispositivos de impressão 3D são dispositivos capazes de manufaturar objetos reais a partir de modelos virtuais, através de um processo de adição de material. O objeto a ser manufaturado é subdividido em camadas finas, que são impressas e superpostas umas sobre as outras, produzindo o volume do artefato final. Diferentemente de processos de manufatura por subtração, pouco material é desperdiçado, já que somente o material necessário é usado na construção do artefato, excetuando-se apenas a parte para suporte da estrutura.

Os materiais usados no processo de impressão 3D podem ser termopolímeros ou metais, incluindo alumínio ou aço. O material é aquecido em uma câmara de impressão até uma temperatura ligeiramente mais alta que seu ponto de fusão. A espessura das camadas por sua vez dependem do tipo de material, variando de 0.1 mm para os polímeros e 20 micrones para metais.

Suas aplicações são inúmeras, incluindo as de uso doméstico, na indústria automobilística, construção de armamentos e particularmente em medicina.

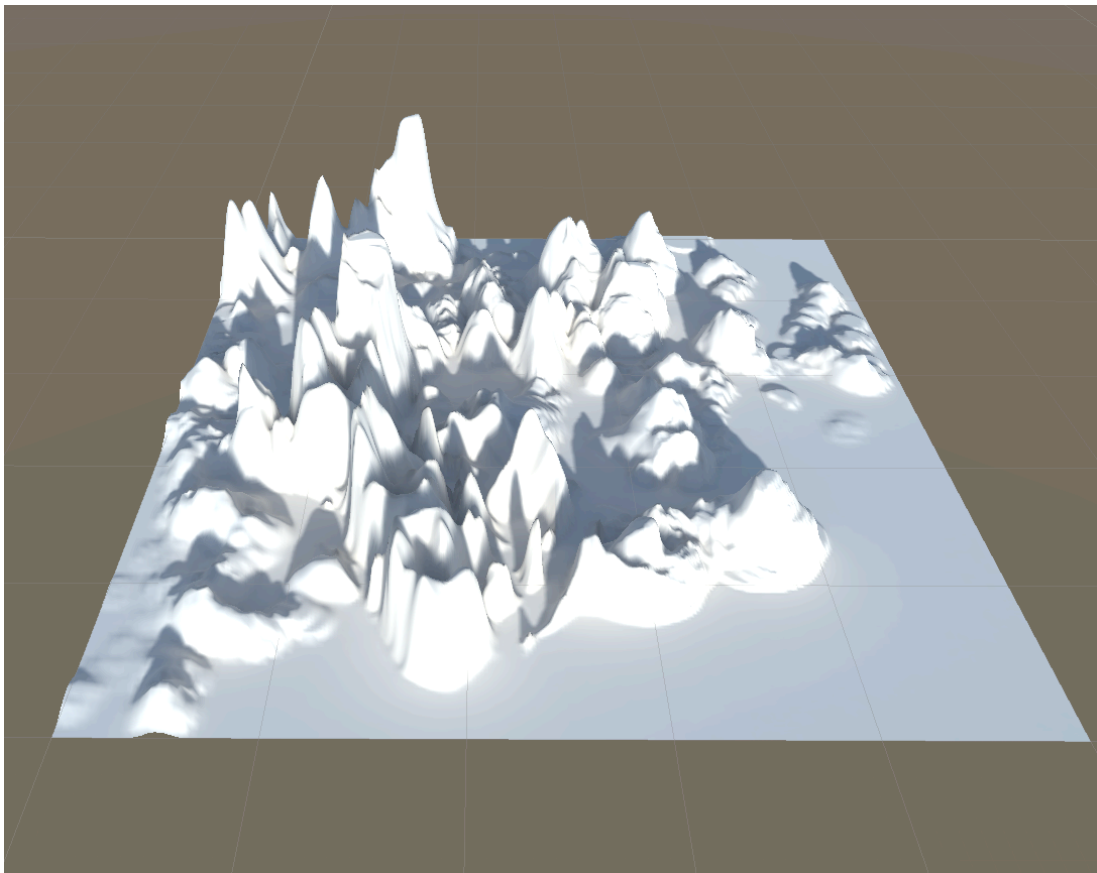
Uma das aplicações mais importantes é a criação de próteses customizadas para pessoas que tenham perdido parte de um membro. O processo envolve o escaneamento do membro, para que um modelo adaptado seja produzido em um modelador 3D e finalmente impresso na impressora 3D. Já existem protótipos bastantes promissores do uso das impressora para produção de próteses de custo menor do que as produzidas em escala industrial (ver a referência [2]).

[2] <http://www.uff.br/?q=noticias/03-07-2018/alunos-da-uff-desenvolvem-proteses-roboticas-de-baixo-custo>. Acessada em 27/08/18 às 17:18h)

- 3) Cite um exemplo de objeto gráfico que pode ser usado em um motor de jogos (Game Engine) de sua escolha e classifique tal objeto (1.25 ponto).

Um objeto gráfico muito comum em jogos 3D são os modelos de terreno. Modelos de terreno são objetos gráficos espaciais bidimensionais. São tipicamente descritos por superfícies com bordo, já que em geral terrenos não são representados como sólidos. Uma exceção são os jogos tipo Minecraft onde terrenos são de fato descritos por sólidos voxelizados. Modelos de terreno volumétricos são utilizados em casos onde eles precisam ser parcialmente destruídos, necessitando assim revelar parte de seu interior.

Segue abaixo um modelo de terreno gerado no motor Unity3D.



- 4) Considere uma aplicação de realidade virtual ou um vídeo jogo. Explique onde seria apropriado usar objetos gráficos paramétricos (curvas ou superfícies) ou objetos implícitos. Cite exemplos de formas presentes na cena que seriam mais facilmente representadas por um ou outro tipo de representação (1.25 ponto).

Uma aplicação de realidade virtual envolve muitos componentes como: um motor de realidade virtual com todos os componentes de software para visualização, *tracking* da cabeça e membros do usuário e tratamento das entradas dos dispositivos de entrada; dispositivos de interação como *joysticks*, *trackballs* e *headmounted displays*; base de dados e outros.

Com relação ao aspecto de visualização em um ambiente virtual, pode-se afirmar que as formas que compõem o cenário podem ser de diferentes tipos. Para a representação, por exemplo, de artefatos em uma plataforma de exploração de petróleo, é possível utilizar superfícies paramétricas geradas por Nurbs na representação de tubulações e peças. Já um reservatório poderia ser descrito de forma implícita, já que é necessário inspecionar ou interagir com as partes internas de tais objetos.

- 5) Faça uma pesquisa sobre as Splines Naturais Cúbicas (*Natural Cubic Splines*). Qual a sua diferença em relação às *B-Splines*? (1.25 ponto).

As Splines Naturais Cúbicas são um caso particular das Splines Interpolantes Cúbicas, isto é um conjunto de segmentos de curvas cúbicas que interpola um conjunto de pontos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

Existem inúmeras formas de se interpolar um conjunto de pontos. As Splines Interpolantes se inspiram na teoria linear das vigas, a qual afirma que para pequenos deslocamentos, a quarta derivada do deslocamento da viga é nula. Se consideramos que a curva interpolante é dada por $y=S(x)$, com $x_1 \leq x \leq x_n$ e $x_2-x_1=x_3-x_2=\dots=x_n-x_{n-1}=h$, isto é os pontos são igualmente espaçados em x , então $S(x)^{iv}=0$ [3].

Integrando esta restrição (de que a quarta derivada é nula) chegamos a conclusão de que os segmentos interpolantes são polinômios de grau 3 (um polinômio de grau 3 derivado quatro vezes se anula).

Logo, $S(x)$ é da forma:

$$S(x) = \begin{cases} a_1(x-x_1)^3 + b_1(x-x_1)^2 + c_1(x-x_1) + d_1, x_1 \leq x \leq x_2 \\ a_2(x-x_2)^3 + b_2(x-x_2)^2 + c_2(x-x_2) + d_2, x_2 \leq x \leq x_3 \\ \vdots \\ a_{n-1}(x-x_{n-1})^3 + b_{n-1}(x-x_{n-1})^2 + c_{n-1}(x-x_{n-1}) + d_{n-1}, x_{n-1} \leq x \leq x_n \end{cases}$$

Para determinar os coeficientes (a_i , b_i , c_i e d_i) dos polinômios de grau 3, que interpolam os pontos dados, é necessário impor algumas restrições.

Primeiramente impomos a restrição de que $S(x_i) = y_i$, isto é, a curva interpola os pontos dados (x_i, y_i) . Além disso, adicionamos mais três restrições determinadas pela teoria das vigas: para uma viga que sofre apenas forças externas, a derivada dos deslocamentos é contínua, logo $S(x)''$ deve ser contínua. Conseqüentemente $S(x)$, $S(x)'$ e $S(x)''$ devem ser contínuas. A existência de segunda derivada contínua tem a ver com a continuidade da curvatura e o aspecto suave e agradável das curvas geradas pela Spline Cúbica.

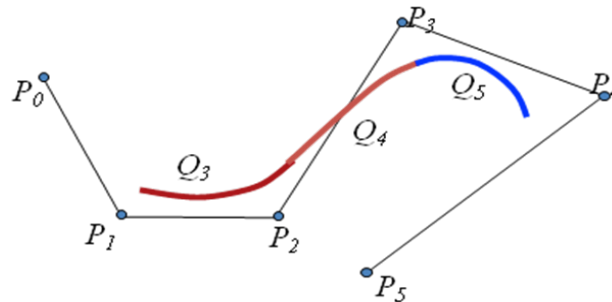
Com estas quatro restrições é possível determinar, após algumas manipulações (ver [3]), um sistema linear de $n-2$ equações a n incógnitas. Para solucionar tal sistema, é necessário introduzir mais restrições de forma a gerar um sistema de n equações por n incógnitas. As Splines Naturais são obtidas quando forçamos as derivadas segundas nos pontos extremos da curva se anularem, isto é $S(x_1)'' = S(x_n)'' = 0$.

[3] Álgebra Linear com Aplicações – Anton Rorres. Bookman - Oitava edição. Páginas 384-386.

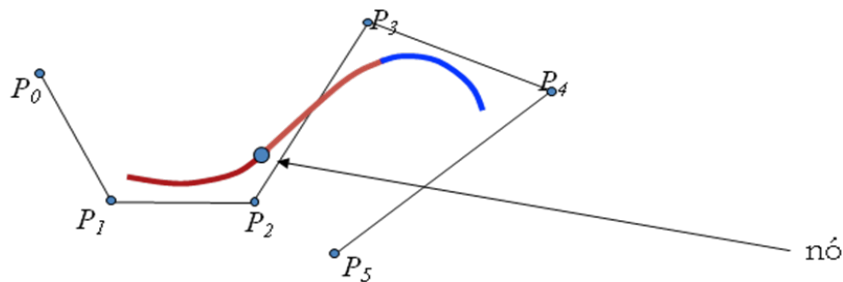
- 6) Explique as diferenças entre uma *B-Spline* e uma *Nurbs*. Qual a vantagem das *Nurbs* sobre as *B-Splines*? (1.25 ponto).

Uma B-Spline é uma curva completa polinomial por partes consistindo de um conjunto de $m-2$ segmentos de curva Q_3, Q_4, \dots, Q_m definidos por $m+1$ pontos de controle P_0, P_1, \dots, P_m , $m \geq 3$.

Cada segmento de curva é definido por quatro pontos de controle e quatro funções de mistura. Cada ponto de controle influencia somente quatro segmentos de curva.



Os pontos associados às junções de dois segmentos adjacentes são denominados nós. O valor de u correspondente a um nó é denominado valor do nó. Uma B-spline é dita uniforme se valores dos nós são igualmente espaçados no espaço do parâmetro u . Existem $m+5$ valores de nós em uma B-Spline uniforme.



A expressão que define um segmento de B-Spline é dada por

$$Q_i(u) = \sum_{k=0}^3 B_{i-3+k}(u) p_{i-3+k}, 0 \leq u \leq l \text{ e as bases de B-Spline cúbicas são dadas por:}$$

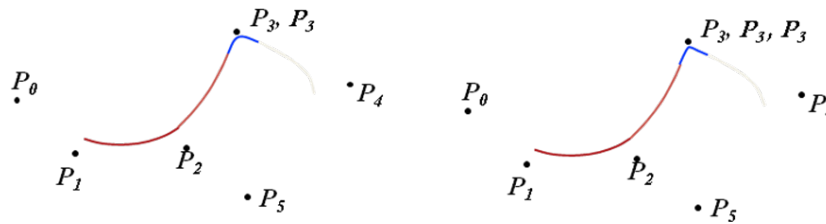
$$\begin{aligned} B_i(u) &= 1/6u^3 \\ B_{i-1}(u) &= 1/6(-3u^3 + 3u^2 + 3u + 1) \\ B_{i-2}(u) &= 1/6(3u^3 - 6u^2 + 4) \\ B_{i-3}(u) &= 1/6(1 - u^3) \end{aligned},$$

No caso geral, as bases de uma curva B-Spline podem ser determinadas através das equações recursivas de Cox-De-Boor conforme abaixo:

$$\begin{aligned} B_{i,1}(u) &= \begin{cases} 1, u_i \leq u < u_{i+1} \\ 0, \text{ caso contrário} \end{cases} \\ B_{i,2}(u) &= \frac{u - u_i}{u_{i+1} - u_i} B_{i,1}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} B_{i+1,1}(u) \\ B_{i,3}(u) &= \frac{u - u_i}{u_{i+2} - u_i} B_{i,2}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} B_{i+1,2}(u) \\ B_{i,4}(u) &= \frac{u - u_i}{u_{i+3} - u_i} B_{i,3}(u) + \frac{u_{i+4} - u}{u_{i+4} - u_{i+1}} B_{i+1,3}(u) \end{aligned}$$

B-Splines normalmente não interpolam seus pontos de controle. Entretanto, é possível alcançar tal efeito de duas formas:

- Replicando pontos de controle, o que pode causar perdas de continuidade na curva



- Aumentando a multiplicidade dos nós, o que não causa perda de continuidade, mas que gera curvas B-Spline não uniformes, já que o intervalo no espaço de parâmetros entre os valores de nós não é mais o mesmo.

NURBS são B-Splines não uniformes dadas pela razão de dois polinômios:

$$\frac{\sum_{i=0}^n P_i w_i B_{i,k}(u)}{\sum_{i=0}^n w_i B_{i,k}(u)}$$

Os valores w_i associados a cada ponto de controle são pesos que podem ser vistos como parâmetros extras. Os w_i afetam a curva apenas localmente. A curva é atraída para um ponto P_i se o w_i correspondente aumenta e é afastada de P_i se w_i diminui. Os w_i podem ser compreendidos como parâmetros que regem o acoplamento da curva aos pontos de controle.

Tanto as B-Splines quanto as Nurbs são curvas compostas por segmentos de curva definidos com base em pontos de controle e vetores de nós. Assim como em uma B-Spline não-uniforme, nas Nurbs (*Non Uniform Rational B-Spline*), o espaço entre os valores dos nós não é uniforme, podendo ter multiplicidade diferente de um.

O que de fato torna as Nurbs diferentes é a existência de um peso w_i , associado a cada ponto de controle P_i , que afeta a curva apenas localmente, assim como os pontos de controle, e que funciona como um fator de acoplamento. Quanto maior o valor de w_i ,

mais a curva se aproxima do ponto de controle P_i . A curva $P(t)$ de grau k é obtida através de uma equação envolvendo uma divisão pelo somatório das bases de B-Splines ponderadas pelos pesos, o que explica o porque da curva ser considerada uma curva racional.

- 7) O Inkscape (<https://inkscape.org/en/>) é um software para desenho vetorial. Uma das ferramentas do Inkscape permite renderizar árvores aleatórias bidimensionais. Um dos modos de gerar tais figuras é utilizar um sistema de reescrita chamado *L-Systems*. Faça uma pesquisa sobre *L-Systems* e sua relação com fractais e gramáticas (1.25 ponto).

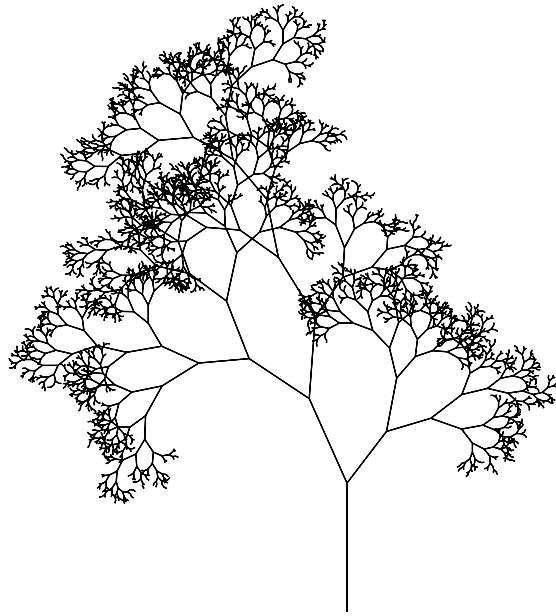


Figura gerada no Inkscape (<https://inkscape.org/en/>)

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 [4].

Um exemplo bem simples e de fácil compreensão, publicado em [4], 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: $(F \rightarrow F+F-F-F+F)$

 $n = 0:$

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

Outro exemplo, agora capaz de produzir um estrutura de ramificação semelhante a uma planta:

variáveis : X F

constantes : + - []

axioma : X

regra de produção : $(X \rightarrow F+[[X]-X]-F[-FX]+X), (F \rightarrow FF)$

angle : 25°

onde "+" significar girar a esquerda de 25 graus, "-" significa girar a direita de 25 graus, "[" significar armazenar o estado (posição) atual, "]" significa recupera o estado anterior. X neste caso não representa nenhuma ação de desenho e somente serve para controlar a evolução do processo.



Figura obtida de <https://commons.wikimedia.org/wiki/File:Fractal-plant.svg>. Licença [GNU Free Documentation License](#).

[4]: <https://en.wikipedia.org/wiki/L-system>. Acessado em 01-09-2018 às 19:00h.

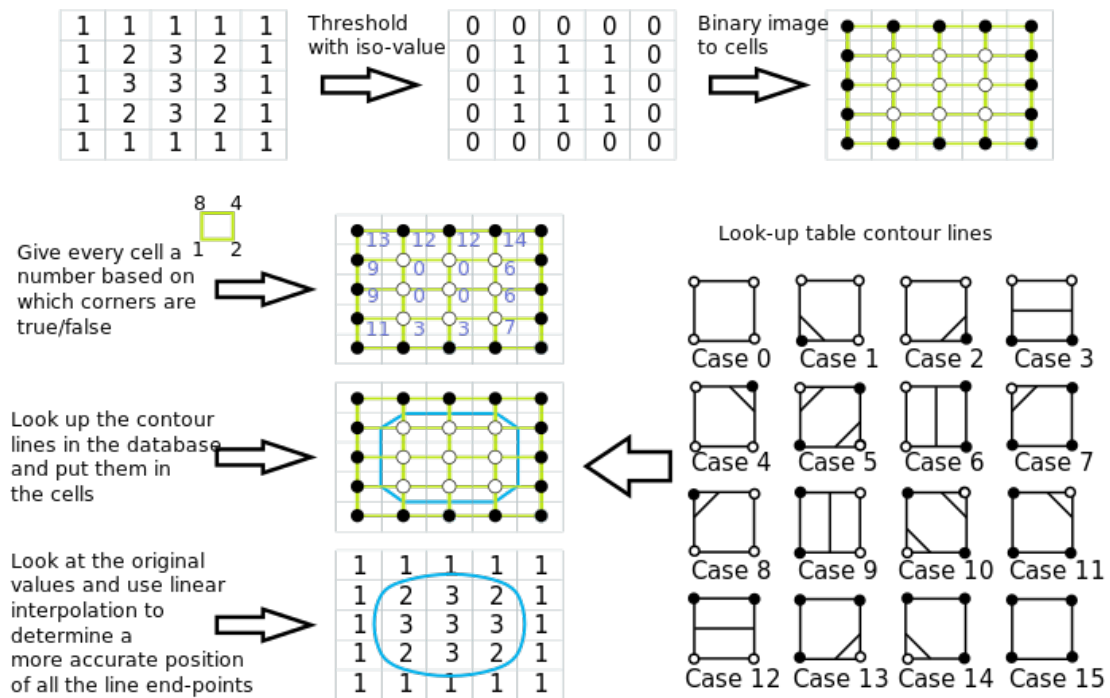
- 8) Explique o algoritmo *Marching Squares* e sua diferença em relação ao algoritmo para poligonização de curvas implícitas visto em aula (tópico objetos gráficos 2D) (1.25 ponto).

O *Marching Squares* é um algoritmo utilizado para poligonizar um isocontorno, isto é, uma curva implícita $F(x,y) = k$ em um campo escalar definido por valores $F(x,y)$. Observe que o caso $F(x,y) = 0$ é um caso particular de $F(x,y) = k$. Pode-se dizer que ele é a versão bidimensional do algoritmo *Marching Cubes*.

Seja um campo escalar $F(x,y)$, aproximado por um campo escalar discreto F_{ij} em um reticulado. O algoritmo processa cada uma das células do reticulado individualmente, verificando, para cada aresta da célula, se há troca de sinal da função $F_{ij}-k$ avaliada nos seus vértices.

A configuração de sinais determina a geometria de parte do contorno no interior da célula. O *Marching Squares* utiliza seguinte tabela de busca (*look-up table*) com 16 casos para determinar a topologia local da curva a ser extraída. Posteriormente, as posições onde a curva corta as arestas de uma célula são calculadas via um processo de interpolação, levando em conta o valor do campo nos vértices do reticulado

A figura abaixo, obtida de [5] sob a licença [Creative Commons Attribution-Share Alike 4.0 International](https://creativecommons.org/licenses/by-sa/4.0/), explica como funciona o processo. As três figuras superiores correspondem, respectivamente, ao modelo discreto do campo escalar F_{ij} (no caso uma imagem digital), ao modelo obtido após limiarização (todos os valores acima de 1 viram 1 e os abaixo viram 0) e finalmente a criação do reticulado. Na figura, o autor indica os pontos pretos como verdadeiro e os brancos como falso, o que seria equivalente a positivo (fora do contorno) ou negativo (dentro do contorno). De posse do reticulado cada célula é rotulada de acordo com o caso especificado pela tabela de casos. Finalmente, é especificada a topologia de toda as células com base nos rótulos e usado um processo de interpolação com os valores escalares originais para determinar a posição dos vértices da curva extraída.



A diferença do Marching Squares para o algoritmo de poligonização no livro Fundamento da Computação Gráfica de Jonas e Velho e, que foi apresentado no curso, é que o Marching Squares trabalha sobre um reticulado e não em uma triangulação. Esta diferença acarreta no fato de que o Marching Squares apresenta casos de ambiguidade que não ocorrem no algoritmo baseado em uma triangulação.

[5] https://commons.wikimedia.org/wiki/File:Marching_squares_algorithm.svg.
Acessado em 02/09/2019 as 11:46;