Universidade Federal de Santa Maria
Prof. Cesar Tadeu Pozzer
Data: 23/06/2022
Nome: Bento Borges Schirmer

3.8

Prova 1 – Computação Gráfica

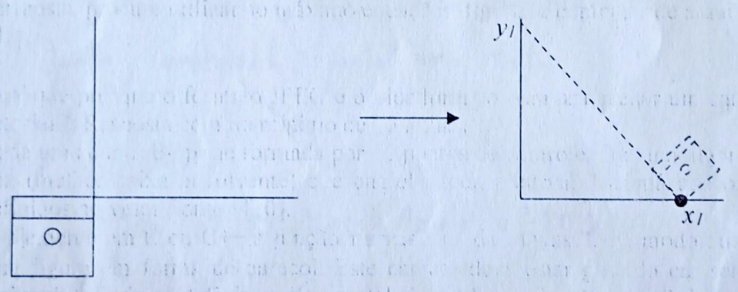
Em cada resposta, procure utilizar ao máximo equações, figuras e comentar de assuntos que sejam diretamente relacionados.

8 1. Explique por que o formato JPEG é o pior formato para armazenar um cartaz que possui texto e figuras vetoriais? Resposta com no mínimo de 10 linhas.

6 2. Dada uma curva B-spline formada por 10 pontos de controle, faça um algoritmo para calcular o bounding box (melhor caixa envolvente) que engloba toda a curva. Assuma que os pontos de controle já estão definidos no vetor Ponto v[10].

0 3. Implemente em C ou C++ a função `render()` da canvas2D de modo que seja gerado uma figura em forma de caracol. Este caracol deve estar girando em sentido horário (animado). Pode-se definir variáveis globais. Utilize a função point() da canvas2D para desenho.

1 4. Tem-se um quadrado de diagonal d e lado l, como mostrado na figura a esquerda. Descreva uma concatenação de matrizes de transformação M (Não precisa multiplicar), que ao multiplicar P'=MP gere a configuração final mostrada na figura pontilhada. A figura final está alinhada com a linha pontilhada. A figura final tem lado 1/3 do lado original. Desenhe a posição do quadrado a cada transformação.



1.

Gráficos vetoriais têm resolução infinita e numa imagem apresentam bordas acentuadas, isto é, a transição é brusca entre um pixel e outro. Uma foto possui bordas suaves, pois o sensor da câmera capta de modo limitado a riqueza de detalhes do mundo real.

JPEG opera em blocos de 8x8 pixels e aplica para cada canal de cor uma função ortogonal Discrete-Cosine Function (DCT), que passa os valores originais para o campo da frequência, onde todas as frequências são uma distância para a do canto superior esquerdo. Numa foto, as frequências são baixas por causa das bordas, gráficos vetoriais causam frequências altíssimas. Ao quantizar os blocos 8x8, a maioria das frequências da foto viram zero, dão boa compressão e visualmente causam pouca perda de qualidade. A quantização com gráficos vetoriais borra tudo, falha em reduzir a entropia e não ocasiona tanta compressão quanto uma foto.

2.

```
BBox BBox_of_BSpline (Ponto v[10])
{
  Ponto pixel_coords[] = BSpline_project_pixels(v);

  Ponto min = {Infinity, Infinity};
  Ponto max = {-Infinity, -Infinity};

  Foreach (Ponto coord of pixel_coords)
  {
    min = Min(min, coord);
    max = Max(max, coord);
  }

  return (BBox){min, max};
}
```

Bento Borges Schirmer
23/06/2022

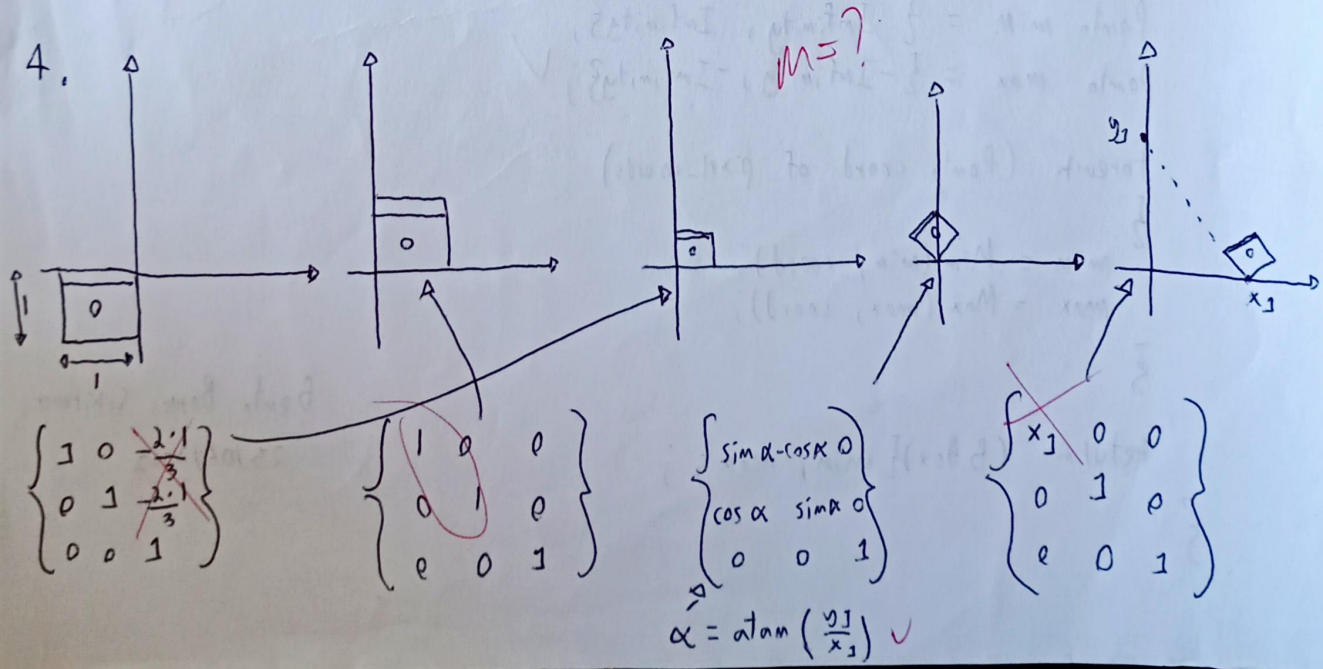
3.

```
#include <math.h>
#include <time.h>
#include "gl_canvas2d.h"
struct p2 { double r; double t; }; //polar point, radius and angle.
struct v2 { double x; double y; };

~~struct v2 from~~
struct v2 p2_to_v2 (struct p2);
double    delta (void);

void render (void)
{
  cv_translate ( screenWidth/2, screenHeight/2 );
  struct p2 p = { 1.0, -M_PI * delta (void) };
  for (int i = 0; i < 1000000; i++)
  {
    cv_point ( p2_to_v2 (p));
    p.r *= 1.3;
    p.t += M_PI * 2.0/360.0;
  }
}
```

4.



$$\begin{Bmatrix} 1 & 0 & -\frac{2.1}{3} \\ 0 & 1 & -\frac{1.1}{3} \\ 0 & 0 & 1 \end{Bmatrix} \quad \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad \begin{Bmatrix} \sin\alpha & -\cos\alpha & 0 \\ \cos\alpha & \sin\alpha & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad \begin{Bmatrix} x_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix}$$

$$\alpha = \operatorname{atan}\left(\frac{y_1}{x_1}\right)$$