Universidade Federal de Santa Maria
Curso de Ciência da Computação
Disciplina: Computação Gráfica
Primeiro Semestre de 2018
Prof. Cesar Tadeu Pozzer
Data: 29/03/2018

Trabalho 1 - Transformada Discreta de Cosseno

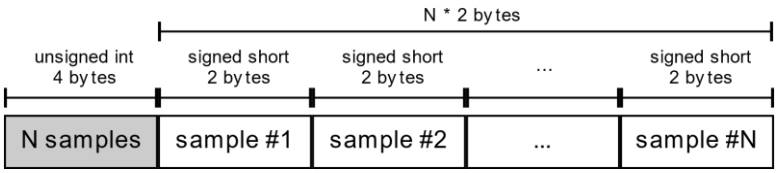
Ferramentas

Linguagem C++, utilizando a API Canvas2D (disponível no site da disciplina) e IDE Code::Blocks, compilando com MinGW (disponível na versão 17.12 da IDE Code::Blocks). **Não podem ser utilizadas bibliotecas auxiliares**. Não pode ser usada a API OpenGL.

Descrição

Desenvolva um programa que aplique a Transformada Discreta de Cosseno (DCT) e sua inversa (IDCT) em um sinal amostrado unidimensional.

As amostras do sinal terão um valor entre -100 e 100 e deverão ser lidas do arquivo binário de entrada **samples.dct** que é fornecido em anexo. Esse arquivo está estruturado com a quantidade de amostras (samples) no início, seguida pelos valores de cada amostra, conforme a figura a seguir.



O tamanho máximo da janela do programa deverá ser HD (1280x720). Caso o monitor utilizado pelo aluno não suporte essa resolução, poderá ser utilizada uma resolução menor seguindo a proporção 16:9.

Requisitos do programa:

1. Plotar um gráfico com tamanho mínimo 600x400 pixels. Sobre esse gráfico deverão ser plotados, utilizando-se cores diferentes, os seguintes dados:
   1.1. Os valores das amostras lidas do arquivo de entrada;
   1.2. Os valores reconstruídos após a aplicação da DCT e IDCT;
   1.3. A diferença entre os valores originais e os reconstruídos.

   Obs.: considerando que os valores das amostras representam apenas o eixo Y, os pontos plotados no gráfico deverão ser distribuídos igualmente ao longo do eixo X.

Exemplo de plotagem das amostras originais e reconstruídas

2. A área de plotagem e escala do gráfico devem se auto ajustar em função dos valores das amostras a serem exibidos e tamanho da tela.
3. Botões para CARREGAR, SALVAR e APLICAR a DCT e IDCT. O básico são 3 botões:
   3.1. CARREGAR: lê o arquivo de entrada e automaticamente plota os valores no gráfico;
   3.2. SALVAR: salva os valores reconstruídos sobrescrevendo o arquivo de entrada;
   3.3. APLICAR: aplica a DCT e a IDCT sobre as amostras e plota os valores reconstruídos e a diferença no gráfico.
4. Checkboxes para habilitar/desabilitar a plotagem de cada tipo de valores (originais, reconstruídos, diferenças).
5. Utilização simples.
6. Todo em tempo real: uma seleção implica na exibição do resultado sem precisar clicar em botões adicionais.
7. Visualmente bonito e bem organizado: organizar o espaço na tela para dispor os elementos da interface.
8. Didático. Criar componentes (botões, barras, sliders, etc) e interfaces amigáveis.

Extras (Para nota acima de 9.0):

1. (+1) Usar vetor de quantização.

2. (+1) Permitir ativar/desativar formas de onda ortogonais durante a reconstrução das amostras.

3. (+1) Parametrizar o tamanho dos blocos (e.g. 4, 8, 16, em 1D, ou 8x8, 16x16, 32x32, em 2D).

4. (+3) Carregar a imagem **lena128_24bits.bmp** (em anexo). Aplica a DCT e IDCT e exibir na tela as imagens de entrada e saída. Obs.: para ler o arquivo de entrada deverá ser utilizada, **exclusivamente**, a classe Bmp disponível no demo **gl_14_texture** (disponível no site da disciplina).

5. Mais ideias que acrescentem ao conteúdo explorado neste trabalho também poderão ser recompensadas.

O trabalho deve apresentar uma lista de instruções, explicando de forma como o usuário deve interagir com o programa. Enumere no início do código fonte (arquivo main.cpp) os quesitos que foram implementados.

Data e Formato de Entrega:

- Data: 16/04/2018, até as 23:59. Após esse prazo o trabalho será desconsiderado.

- No e-mail e no cabeçalho do arquivo, deve conter o nome completo do aluno. O arquivo deve ser enviado para pozzer3@gmail.com e bruno.t.nasc@gmail.com, com o subject “CG T1”.
- O programa deve ser enviado em um arquivo compactado ***fulano*.RAR** (*fulano* = login do aluno). Dentro deste arquivo deve haver um **diretório com o mesmo nome do arquivo** e, dentro deste diretório, os arquivos do trabalho. **Deve-se enviar somente**: código fonte, imagens e arquivos de áudio (quando existirem) e o projeto. Não devem ser enviadas libs, executáveis e DLLs em geral.

Critérios de Avaliação:

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação. Não comente por exemplo o que faz b++.
- README: incluir um arquivo “README.txt” contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras).
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- Legibilidade: nome de variáveis, estruturação do código.
- Clareza: facilidade de compreensão – evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- Funcionalidade: o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão a nota 0 (zero).