

Trabalho de EC212/EC215/C209

Durante o semestre, veremos várias técnicas de processamento de imagens, áudios e vídeos, muitas delas amplamente utilizadas em visão computacional e computação gráfica. Em realidade, essas técnicas normalmente são executadas com bibliotecas já estabelecidas e conhecidas, nas quais muitas operações já estão abstraídas em classes e funções. Temos bibliotecas de visão computacional como OpenCV, de aprendizado de máquina como PyTorch ou TensorFlow ou síntese de imagens como OpenGL que simplificam muito o trabalho do programador.

Alguns tópicos comuns dentro da multimídia incluem:

- Alinhamento de imagens (image alignment);
- Codificação e decodificação de arquivos:
 - JPEG
 - OBJ
 - TGA
 - WAV
 - etc.
- Compressão e descompressão de dados (data compression and decompression):
 - LZS
 - Fractal Compression
 - Wavelet Compression
 - etc.
- Detecção de objetos (object detection);
- Estimativa de movimento (feature tracking ou motion estimation);
- Reconhecimento óptico de caracteres (optical character recognition);
- Remoção de ruídos (noise removal);
- Segmentação de imagens (image segmentation);
- Subtração de fundo (background subtraction);
- Template matching; Preenchimento de regiões (boundary fill)
- Algoritmos para desenho de linha (line drawing algorithm)
- Técnicas de antisserrilhamento (anti-aliasing)
- Remoção de linhas ou faces escondidas (hidden-line / hidden-face removal)
- Algoritmos de iluminação global
- Equalização e filtragem de áudio
- Efeitos sonoros (wah-wah, phaser, vibrato, flanger, chorus, echo, tremolo, vibrato, etc)
- Dentre outros.

Escolha um dos tópicos acima (ou proponha um novo para o monitor) e resolva-o na linguagem de programação de sua escolha, podendo utilizar bibliotecas caso necessário (desde que não tornem a solução muito trivial). Crie um vídeo de 8 a 15 minutos explicando a solução e um código que resolve o problema escolhido.

No vídeo, deve ser explicado:

- Uma introdução geral do tópico escolhido;

- Uma introdução geral da técnica escolhida (sugestões de tópicos: Por quem foi criado? Quando? Por quê? Como funciona?)
- Apresentação do algoritmo (sugestões de tópicos: Como funciona? Qual é a teoria por trás dele?);
- Conclusões (sugestões de tópicos: Quais são os pontos fortes da técnica? E os fracos? Como poderia ser melhorado? Seria viável implementar junto com outra técnica?)

As equipes devem ser formadas por **no máximo 3 alunos**. Um dos alunos deverá compartilhar a tela do computador e os mesmos podem realizar a apresentação por áudio.

Tema do Trabalho:

O tema do trabalho, assim como os membros da equipe (nomes e matrículas) devem ser entregues até o dia **10 de Março** na tarefa do Teams para aprovação.

Entrega:

Coloque o vídeo em uma plataforma na nuvem como Google Drive, OneDrive ou qualquer outra plataforma semelhante, e envie na tarefa do Teams os links para que eu tenha acesso a eles (antes de enviarem verifiquem se o compartilhamento está sendo realizado de forma adequada).

O código desenvolvido deve ser colocado no GitHub junto com instruções em como executar o código e alguns testes. O link para o GitHub deve ser colocado na tarefa também.

Data de Entrega:

O trabalho deverá ser entregue até dia **14 de Abril** na tarefa do Teams.

Avaliação:

50% da nota será a apresentação (clareza na explicação, eloquência, domínio do assunto apresentado) e 50% no desenvolvimento e funcionamento (funcionamento do algoritmo, complexidade da solução e qualidade do software final e documentação).

Obs: Como alguns destes problemas podem ser resolvidos de forma muito trivial com auxílio de bibliotecas, espera-se que as implementações tenham certa complexidade. Por exemplo: codificação & decodificação e compressão & descompressão de dados devem ser implementados com o NumPy ou semelhante de forma que o algoritmo seja implementado, para evitar códigos que são trivialmente resolvidos. Atentem-se à complexidade da solução.