

# Trabalho Final

Fernando Nunes – 14/0039678 – Fábio Marques – 14/0039082 – Bruno Rodrigues – 14/0017658

Orientador: Bruno Macchiavello  
Universidade de Brasília  
Brasília, DF.

**Resumo** — Para o nosso trabalho final, precisamos fazer um programa que, dado uma imagem de entrada, devolva a mesma imagem, com o efeito de cartoon. Para isso, diversos processamentos vistos ao longo do semestre foram utilizados, seguindo um artigo de referência dado pelo professor.

**Palavras-chave** – Imagem, cartoon, processamento, filtro.

## I. INTRODUÇÃO

Este trabalho final de Disciplina foi feito em grupo e utilizando muitas das técnicas de processamento de imagens em baixo e médio nível aprendidos durante o semestre. A fim de uma completa compreensão do funcionamento dos processos envolvidos na adição do efeito cartoon a uma imagem, são necessários conhecimentos em filtragem de média e mediana, segmentação para a localização de bordas, operações morfológicas(dilatação), remoção de elementos a partir de um limiar mínimo de área, filtragem bilateral e quantização de cores.

## II. METODOLOGIA

Com o objetivo de obter o efeito desejado, o algoritmo foi dividido em três partes. Na parte A, utilizaremos algoritmos para obter as bordas da imagem original, transformada para níveis de cinza. Dessa forma, obteremos os traços característicos de um cartoon. Na parte B, utilizaremos algoritmos para tornar as cores mais uniformes, de modo a deixar as cores da imagem semelhantes a pintura de uma desenho. Na parte C, apenas reuniremos os resultados das partes A e B para obter o nosso efeito de cartoon.

### A. Parte A

**1) Filtro de Mediana:** Primeiramente, foi utilizado um filtro de mediana para deixar a imagem mais uniforme e retirar ruídos do tipo salt-and-pepper. Esse processamento tem uma função importante visto que auxilia em uma melhoria considerável na detecção das bordas da imagem, pois utilizaremos métodos de segmentação por transição que são sensíveis ao ruído. O filtro utilizado foi de tamanho 7x7, de modo que ele é grande o suficiente para suavizar os níveis de cinza, e não tão grande a ponto de descaracterizar as bordas. Para a aplicação desse filtro, foi usada a função `medfilt2` do MatLab.

**2) Detector de Bordas de Canny:** Reduzidos os ruídos, foi aplicado um detector de bordas de Canny, usando a função `edge`. Esse método tem a importante característica de não produzir bordas duplas (como um filtro Laplaciano), que será importante para um resultado efetivo no próximo passo. O limiar máximo utilizado foi de 0.15 e o limiar mínimo foi calculado pelo função `edge` do MatLab. Esse valor foi escolhido mediante testes com diversos tipos de imagem e apresentou o melhor resultado médio, avaliado pelos integrantes do grupo.

**3) Dilatação:** Nesta etapa, utilizaremos uma operação morfológica com o intuito de deixar as bordas mais espessas, utilizando a função `imdilate` do MatLab. A operação utilizada foi a dilatação, utilizando um elemento estruturante do tipo disco de tamanho 2x2, obtido a partir da função `strel`. Depois desse passo, teremos uma imagem com as bordas mais grossas e mais parecidas com as bordas de um cartoon. Sendo assim, podemos perceber que se existissem linhas duplas após o passo anterior, as bordas ficariam demasiadamente espessas e bordas próximas se juntariam.

**4) Filtragem das Bordas:** Ao final da parte A, utilizaremos a função `bwareaopen` para diminuir a quantidade de bordas detectadas e assim, diminuir a quantidade de áreas que ficaram sobrecarregadas com bordas, diminuindo a qualidade da nossa saída em estilo cartoon. Com essa função, conseguimos retirar qualquer borda cuja área ultrapassasse 200 pixels.

### B. Parte B

**1) Filtragem Bilateral:** O primeiro e principal fator no tratamento das cores, para se obter um efeito de cartoon na imagem, é a aplicação do filtro bilateral.

O filtro bilateral tem o objetivo de homogeneizar as cores da imagem, de forma que a mesma se afaste um pouco da realidade e adentre mais às características de uma imagem cartoonizada.

Esse filtro, em sua aplicação, se parece com uma convolução usando máscara gaussiana. O seu diferencial está nos pesos de cada pixel na hora da soma; quando um pixel adjacente tiver valor muito diferente do pixel central, ele vai ter um peso menor do que um pixel adjacente que tenha valor próximo ao central. Com isso, podemos nos tranquilizar quanto à preservação das mudanças bruscas de cor, o que não aconteceria se usássemos um filtro gaussiano.

Um dos problemas relacionados à execução do filtro bilateral é a sua complexidade computacional. Pelo fato de

estarmos rodando no matlab, aplicar tal filtro exigiu relativamente bastante tempo, e, por isso, foi utilizada a função `imresize` com fator 1/4 para reduzir a imagem e, assim, reduzir também o tempo de execução.

No artigo tomado por nós como referência, o filtro foi iterado 14 vezes em uma imagem, porém, no nosso caso, uma iteração nos deu bons resultados, ao passo que duas ou mais iterações não resultaram em imagens muito diferentes, talvez pelo fato de estarmos trabalhando com canais RGB, enquanto, em outras implementações que estudamos, foram usados canais como CIElab.

Nossa implementação foi baseada em diversas implementações encontradas, e em diferentes referências, que serão listadas no final deste relatório. Os limiares, por sua vez, foram determinados empiricamente, conforme testávamos a função implementada em várias entradas. O tamanho da máscara escolhido, baseado no artigo lido, foi 9 x 9.

**2) Filtro de Mediana:** Assim como na Parte A, usamos esse filtro para poder suavizar certos comportamentos não desejados na imagem. No caso do tratamento das bordas, esse foi usado para amenizar o ruído Salt and Pepper, e, na Parte B, esse filtro tem o objetivo de, ao suavizar novamente a imagem, reduzir os problemas trazidos pela interpolação, usada no final da filtragem bilateral para voltar a imagem ao seu tamanho original.

Além do motivo listado acima, o filtro de mediana também contribuiu um pouco com a estética cartoonizada da imagem.

**3) Processo de Quantização:** Por fim da parte de tratamento das cores, é realizada a quantização dos níveis de cor em cada canal da imagem, arredondando os valores para baixo, usando a função `floor()` do matlab.

Quanto menor a quantidade de níveis de cor, menos real a imagem ficava, favorecendo assim o fator cartoon, porém a qualidade visual piorava consideravelmente, ao passo que, quanto maior a quantidade de níveis, a imagem ficava visualmente melhor, porém era perdido o fator cartoon, o que não poderia acontecer de forma alguma.

Com isso, ao testar diversas quantizações, percebemos que a quantização com fator 16 (em cada canal) balanceava bem os dois quesitos, qualidade visual e aparência cartoon da imagem.

### C. Parte C

**1) Junção das Partes A e B:** Para completar esta parte, a única operação realizada foi a multiplicação, pixel a pixel, do negativo do resultado da parte A com o resultado da parte B. Assim, obtendo a imagem resultante das cores quantizadas sobrepostas sobre as bordas obtidas na parte A.

## III. RESULTADOS

Depois de testes com as mais variadas entradas, chegamos a conclusão de que nossa implementação do algoritmo de cartoonização foi bem sucedida, alcançando resultados muito bons em uma boa variedade de imagens.

É importante ressaltar que as avaliações realizadas sobre os resultados foram bastante subjetivas, fato que levou a algumas divergências sobre as imagens entre os integrantes do grupo. Entretanto, para a maioria das imagens testadas as opiniões foram semelhantes, indicando que o nosso programa alcançou um resultado satisfatório.

Além disso, a definição dos parâmetros das funções foram escolhidos por meio de métodos empíricos, nos quais, para diversas entradas, o resultado foi avaliado pelos membros do grupo até que uma convergência de valores foi encontrada. Ainda assim, os resultados não foram bons todas as imagens testadas. Em alguns casos os limiares empregados, principalmente na função de detecção de bordas, não eram os valores que tornariam o efeito o melhor possível, subjetivamente.



A imagem acima foi selecionada pelos integrantes do grupo como a imagem que obteve o melhor resultado. Podemos perceber que as bordas destacadas nessa imagem foram, em sua maioria, as bordas que melhor definem os contornos do carro. Além disso, apesar da existência de detalhes no fundo da imagem, apenas os contornos gerais foram delimitados, dando uma aparência bem cartoonizada para o resultado.

Com relação aos processamentos realizados com as cores, nota-se uma suavização dos níveis antes da quantização, de modo que as regiões ficaram compactadas sem uma variação brusca de cor ao longo dos objetos. Sendo assim, podemos concluir que essa imagem foi um caso bem sucedido dos processamentos realizados na parte B do algoritmo.



A imagem acima foi outro exemplo no qual o nosso programa foi muito bem-sucedido. Percebe-se que os contornos do leão ficaram nítidos e, principalmente, a juba ficou muito parecida com um desenho real, com contornos espaçados e verticais.

Além disso, a quantização das cores mais uma vez alcançou um excelente resultado, no qual as regiões ficaram bem delimitadas e uma certa textura parecida com um cartoon feito a mão. Sendo assim, mostrando mais um caso em que o algoritmo implementado foi satisfatório.

da quantização das cores. Podemos ver que o fundo foi suavizado sem perder as características que o diferenciam da grama, abaixo do cavalo. Além disso, as regiões quantizadas ao longo do corpo do cavalo conferiram um certo efeito de textura diferenciada.



Acima, vemos a imagem no qual o nosso programa apresentou o pior resultado. Nessa imagem, é nítido que foram destacadas mais bordas do que o necessário. Pode-se perceber isso principalmente ao longo do capuz e na face presentes na foto, onde diversos contornos não eram importantes para conferir ao efeito de cartoon para a imagem.

Ainda no caso dessa imagem, em que a supersegmentação prejudicou o efeito final, podemos perceber que quantização das cores realizadas na parte B do algoritmo foi suficiente para se alcançar os efeitos desejados no quesito das cores.



Já no caso dessa imagem, os resultados foram satisfatórios, mas não tão bons quanto nos exemplos anteriores. Os contornos do animal em si foram obtidos corretamente pela parte A do algoritmo. Entretanto os detalhes na grama ressaltados pelas bordas não eram desejados, fato que faz com que a imagem fuja da ideia do cartoon. Além disso, outros contornos não desejados podem ser encontrados na região da cabeça do cavalo, no qual diversas bordas foram salientadas sem necessidade.

Embora o resultado da detecção de bordas tenha sido apenas satisfatório, ele é compensado por um ótimo resultado

#### IV. MAIS RESULTADOS

Abaixo, encontramos mais resultados decorrentes da utilização do nosso programa. Podemos perceber resultados muito bons como nas duas primeiras imagens, mas também resultados intermediários como na terceira imagem.





## V. CONCLUSÃO

A implementação do objetivo do projeto, o qual era a adição do efeito cartoon a uma fotografia, foi bem sucedida. Foram utilizadas várias funções da ferramenta MatLab e uma somente teve de ser pesquisada para a sua confecção sendo ela a que realiza na imagem a filtragem bilateral. Para a criação da função anterior foi tomada como referência um tópico do fórum do MatLab listado nas referências, já que sua implementação e opiniões favoráveis foram consistentes e a partir desse modelo nossa função "colorBilateralFil" foi modelada.

As variâncias escolhidas para a Gaussiana das intensidade dos pixels e para a das distâncias, o tamanho do elemento estruturante para a dilatação das bordas, o tamanho das elementos a serem apagados pela função "bwareaopen" e o limiar para segmentação foram escolhidos de modo empírico sendo selecionados aqueles que levaram nos melhores resultados. Dessa forma, essas funções levaram a um processamento pouco robusto.

A segmentação foi feita com o algoritmo de Canny pelo fato dele resultar em uma imagem com a segmentação da original sem bordas duplas, além de ter oferecido um resultado estético melhor do que outros métodos para detecção de bordas.

A instrução do artigo de referência era a de realizar a filtragem bilateral diversas vezes, mas foi decidido pelo grupo que ela seria feita somente uma vez. Essa decisão foi tomada depois de comparar o resultado de uma única filtragem com o de 4 e 14 filtrações. As diferenças foram muito pequenas, basicamente na iluminação, porém a demora desse processo de filtrações múltiplas não foi a favor de sua viabilidade, portanto, com base no custo-benefício, foi decidida realização de uma única filtragem bilateral.

Mesmo nos resultados em que a cartoonização da imagem não foi esteticamente satisfatória, a imagem resultante da quantização das cores após a filtragem bilateral foi suficiente para uma avaliação estética muito boa. Esse resultado demonstra que a parte mais dependente das imagens de entrada e dos parâmetros das funções foi a segmentação.

## VI. REFERÊNCIAS

- [1] [https://en.wikipedia.org/wiki/Bilateral\\_filter](https://en.wikipedia.org/wiki/Bilateral_filter), acessado 18:00 hrs do dia 04/07/2015
- [2] <http://www.mathworks.com/matlabcentral/fileexchange/12191-bilateral-filtering/content/Bilateral%20Filtering/bfilter2.m>, acessado 20:00 hrs do dia 04/07/2015
- [3] Toonify: Cartoon Photo Effect Application. Dade, Kevin. Stanford University.
- [4] Slides presentes no moodle da disciplina Introdução ao Processamento de Imagens – Prof. Bruno Macchiavello