

# Trabalho Final

## Tema 1: Imagem cartoon

Lucas Mariano Carvalho – 16/0133661

Departamento de Ciências da Computação - CIC  
Universidade de Brasília  
Brasília - DF, Brasil  
lmarianocarvalho@hotmail.com

Evandro Thalles Vale de Castro – 16/0119286

Departamento de Ciências da Computação - CIC  
Universidade de Brasília  
Brasília - DF, Brasil  
evandro-tvc@hotmail.com

**Abstrato** — Este documento tem o intuito de resumir e apresentar o trabalho final de introdução ao processamento de imagens, é feito uma breve introdução teórica ao assunto, apresentado a metodologia utilizada para resolver o pedido, a amostragem dos resultados obtidos e uma conclusão final sobre o trabalho.

**Palavras-chave** — Detecção de bordas; Filtragem Bilateral; Suavização; Mediana; Filtro; Canny;

### I. RESUMO

O trabalho final pretendido pela disciplina de introdução ao processamento de imagens visa colocar em prática tudo o que já foi estudado em sala. Para isso foi pedido que se faça o processamento de imagens em uma imagem real a fim de obter uma imagem “cartoon” que basicamente é uma imagem na forma de desenho. Para que tal processo ocorra é necessário a aplicação de várias técnicas que serão explicadas melhor no decorrer do relatório, mais pode-se citar duas técnicas principais utilizadas que são a detecção de bordas de uma imagem e a sua filtragem bilateral.

### II. INTRODUÇÃO TEÓRICA

#### A. Detecção de Bordas

As propriedades dos objetos, tais como as características geométricas e físicas, são passadas à imagem pois ocasionam variações nos tons de cinza da imagem. Dessa forma, para se detectar e extrair informações dos objetos, muitas técnicas de processamento de imagens são utilizadas, dentre elas a detecção de bordas. Dependendo do fim a que se destina, a detecção de bordas pode ser tida como um fim ou como um pré-processamento para passos subsequentes. De qualquer forma, para que sejam obtidos os resultados desejados, é necessário que a estratégia de detecção de bordas seja eficiente e confiável. A fim de que as variações dos tons de cinza sejam detectadas (bordas) é necessário diferenciar a imagem.

Existem várias formas para se fazer isso: pelo método de Canny, pela diferença gaussiana, dentre outros. Optamos pela detecção por Canny, pois a técnica por Canny, é a técnica mais eficiente e atinge os resultados da maneira que gostaríamos, único ponto negativo de tal metodologia é que exige muito custo de memória.

O detector de bordas de Canny tem como uma de suas principais características a dualidade entre a detecção e a precisão de localização das bordas. Isto é, quanto mais se privilegia a precisão de localização, menor é a razão sinal/ruído e, consequentemente, a detecção se torna cada vez mais sensível aos detalhes espúrios da imagem, e vice-versa. Dessa forma, para cada imagem é necessário a escolha de um valor particular para o fator de escala do núcleo de convolução Gaussiano, o que normalmente envolve tentativa e erro e uma dose de experiência do operador.

Este método utiliza os conceitos de primeira e da segunda derivada de uma maneira muito eficaz. A sua aplicação é clássica é a abordagem gradiente para detecção de bordas na presença de ruído branco ou gaussiano, mas também incorpora elementos da abordagem laplaciano.

A seguir será explanado os procedimentos usados no algoritmo de canny, de quais as operações que ele utiliza para realizar as detecções das bordas:

- 1) ler a imagem de entrada;
- 2) Criar uma máscara gaussiana unidimensional G para para convoluir com I.O fator de escala desta gaussiana é um parâmetro do detector de bordas;  
Criar uma máscara Gaussiana unidimensional Gaussiana para convoluir com I. O fator de escala desta Gaussiana é um parâmetro do detector de bordas;
- 3) Criar uma máscara unidimensional para a derivada primeira da Gaussiana, nas direções x(linha) e y (coluna) , denominando-as de G<sub>x</sub> e G<sub>y</sub>;
- 4) Convoluir a imagem I com G ao longo das linhas, gerando a imagem I<sub>x</sub>, e, analogamente, ao longo da colunas para gerar I<sub>y</sub>;
- 5) Convoluir I<sub>x</sub> com G<sub>x</sub> para I<sub>x</sub> que é o resultado da convolução da imagem I com a primeira derivada da gaussiana ao longo das linhas. Analogamente convoluir I<sub>y</sub> com G<sub>y</sub> para gerar I<sub>y</sub>;
- 6) A magnitude do gradiente é calculada em cada pixel nas formas (x,y) na forma que segue:

$$M(x,y) = [ (I_x'(x,y))^2 + (I_y'(x,y))^2 ]^{1/2}$$

Ou seja, simplesmente a norma do gradiente, que é a distância

7) Supressão não Máxima, tratando-se da busca de máximos locais de  $M(x,y)$  nas direções transversais às bordas, reduzindo num mapa de bordas afinadas - isto é, bordas com espessura de 1 pixel.

### B. Filtro Bilateral

O principal objetivo do filtro bilateral é suavizar as zonas com menor contraste da imagem enquanto não afeta os limites de maior contraste, reduzindo o ruído presente na imagem enquanto preserva seus contornos. O filtro bilateral, considera dois pixels próximos um do outro se ocupam posições espacialmente próximas, como também, se de acordo com uma escala de cores e tonalidades, eles possuem alguma similaridade.

### C. Amostragem

O processo de amostragem nada mais é que a obtenção de amostras de um sinal contínuo, em instantes de tempo igualmente espaçados. No entanto um certo cuidado deve ser tomado na definição da frequência com a qual as amostras são obtidas. Se tal frequência for muito lenta, a posterior reconstrução do sinal pode não ser mais possível.

### D. Dilatação (operações morfológicas)

Uma importante operação que será utilizada também para definir alguns detalhes das bordas, é o processo de dilatação, operação morfológica de extrema importância que será explanado agora.

Utilizando um elemento estruturante onde todos os valores são positivos para efetuar uma dilatação numa imagem em níveis de cinza, obtém-se como imagem resultante uma imagem com detalhes escuros atenuados e detalhes claros acentuados. A operação de dilatação em tons de cinza fornece como saída uma imagem com as mesmas dimensões da imagem de entrada e pixels com valores de 0 a 255, para o caso de uma imagem original com 256 níveis de cinza. Isso significa que o algoritmo para aplicação de dilatação em imagens em tons de cinza segue os seguintes passos:

- I. O elemento estruturante é posicionado no início da imagem a ser aplicada a dilatação;
- II. É feita a comparação da soma das intensidades dos pixels da imagem original com as intensidades dos pixels correspondentes da matriz do elemento estruturante;
- III. O pixel correspondente da imagem de saída terá a intensidade da maior soma desta comparação. Caso a maior soma ultrapassar 255 o valor assumido para a intensidade do pixel correspondente será 255.
- IV. A matriz do elemento estruturante é posicionada no elemento seguinte da matriz que representa a imagem original, retornando-se ao passo de número. Caso o último elemento da matriz da imagem original já tenha sido visitado pela matriz do elemento estruturante, o processo de dilatação está concluído.

### E. Filtro de Mediana

É um filtro que tem por objetivo de suavizar imagem, é de extrema utilidade para casos onde quer preservar as bordas, que para o nosso caso, é o ponto chave para para não somente manter a integridade das bordas, mas também remover certos ruídos presentes na figura. Vale ressaltar que sua resposta se baseia na organização dos pixels.

## III. METODOLOGIA

Com o propósito de simplificar o entendimento de cada algoritmo utilizado para chegar no filtro “cartoon” os métodos utilizados foram separados em duas partes principais sendo elas a quantização de cores e a detecção das bordas de uma imagem de entrada, os dois serão explicados a seguir:

Para deixar o processo mais didático faremos o passo a passo até chegar na imagem final cartoon.



Figura 1. Exemplo de *imagem de entrada*.

### A. Filtragem Bilateral

Já com a imagem de entrada o primeiro passo a ser feito é aplicar o filtro bilateral para que ocorra uma suavização da imagem e um breve borramento. Mas por quê borrar seria bom? Às vezes é necessário borrar um pouco a imagem antes de aplicar o filtro: isso alisa pequenas trepidações e previne falsas bordas. E por isso, quer se evitar o máximo possível de bordas falsas já que o ponto chave do nosso projeto é determinar muito bem as bordas e realizar um processo de segmentação correto. Após isso, ao aplicar esse efeito tanto de suavizar, quanto de borrar aplicou-se o filtro de mediana para diminuir os ruídos presentes na imagem, tal ação foi necessário para tornar o passo seguinte de detecção de bordas o mais preciso possível. Depois, aplicou o processo de dilatação, primeiramente usamos um elemento estruturante de  $4 \times 4$ , mas como o processo de dilatação dá uma certa “engordada”, ocorrerá um processo de preenchimento de espaço muito exacerbado do esperado, ou seja, as bordas começariam a invadir espaço que não pertence a ela, então adotamos um elemento estruturante de  $2 \times 2$  que melhor satisfaz os nossos resultados.

E por fim, para remover pequenas regiões que foram reconhecidas como bordas, mas não eram, aplicou-se uma função que contornava tal situação para que as bordas fiquem

o mais nítidas e precisas o possível. Novamente ressaltando que uns dos principais detalhes que o grupo teve que se preocupar é a integridade das bordas, pois sem elas o processo de cartonização seria um tanto quanto inconveniente de se realizar.



Figura 2. Imagem obtida a partir do filtro bilateral aplicado em cada componente RGB da imagem de entrada.

### B. Detecção das Bordas

Foi feito também a detecção das bordas da imagem de entrada utilizando o método “edge Canny”, para a aplicação deste algoritmo tivemos que transformar a imagem de entrada RGB para os níveis de cinza na qual o valor de cada pixel é uma única amostra de um espaço de cores, isso tudo é por causa da definição de borda que é a mudança de nível de cinza quando ocorre alguma descontinuidade na intensidade, ou quando o gradiente da imagem sofre alguma variação abrupta. E, como pelo método de canny baseia-se em funções que utilizem derivadas, há um termo em específico que a gente pode utilizar a nosso favor que é o gradiente, isto é, quanto mais o pixel é bem localizado, menor será a razão entre ruído/sinal, assim terá o contornos das bordas mais suaves e nítidas para visualizar. Como o código “edge canny” é bastante recorrente e famoso, usamos um função própria do matlab que é a edge. Com isso, basta agora pegar esses filtros criados, que são as imagens aplicadas o filtro bilateral e a imagem aplicada com o edge canny, a após isso, juntar ambos para gerar a imagem final. Esse processo é o final do algoritmo, que por sinal foi extremamente trabalhoso, pegamos como base em vários artigos encontrados na internet e implementamos, na forma empírica, ou seja, tentativa e erro.



Figura 3. Imagem obtida através do algoritmo de detecção de bordas canny

E por final após a obtenção das duas imagens, uma com a do filtro bilateral e a outra com as bordas, é feito o processo de junção das duas imagens obtidas fazendo a diminuição da imagem com bordas em cada componente RGB da imagem filtrada bilateral, resultando assim em uma imagem final com o aspecto de desenho (cartoon).

## IV. RESULTADOS

Para a obtenção dos resultados presentes neste relatório foram utilizadas 35 imagens com diferentes características e a diversificação de tamanhos e qualidades, os resultados obtidos serão mostrados a seguir e será explicado as limitações percebidas e ressaltar as qualidades na aplicação do filtro cartoon.

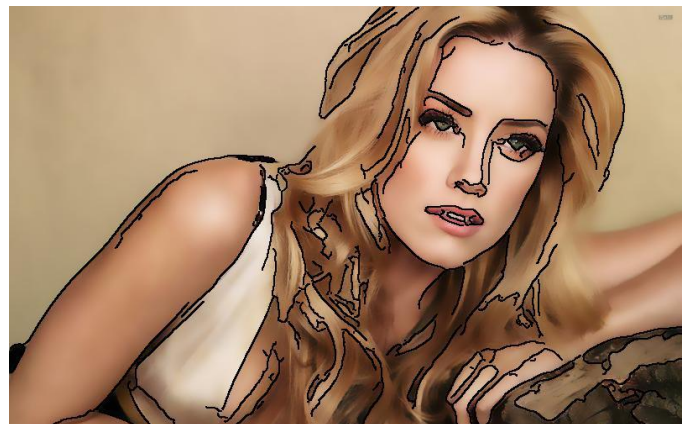


Figura 4. Imagem resultado da imagem de entrada mostrada na figura 1.





Figura 5. Imagem resultado filtro Cartoon.



Figura 6. Imagem resultado filtro Cartoon.



Figura 7. Imagem resultado de uma imagem de pequena dimensão (menor que 400x800 x 400x800).



Figura 8. Imagem resultado de baixa qualidade apesar da dimensão entre 400x800 x 400x800.

O resultado obtido na figura 4, 5 e 6 mostra que realmente ocorreu a “cartoonização” (deixar com aspecto de desenho) das imagens, devido aos traços bem segmentados e as cores realçadas da imagem.

Pode-se perceber que esta técnica de “animar” as imagens possui algumas limitações, como:

- 1) Quanto menor é a dimensão da imagem, mais grossos e menos suaves os contornos da imagem ficavam.
- 2) Quanto muito maior a dimensão da imagem, após a realização da cartoonização a figura gerada não sofria muita alteração, apenas um certo borramento nos detalhes da mesma não aparentando ser do tipo desenho.



Figura 9. Imagem resultado de uma imagem com uma dimensão maior que dimensão entre 400x800 x 400x800..

Como visto apesar dessas limitações o processo de cartoonizar foi muito bem-sucedido em imagens do tipo retrato e em imagens com dimensões entre 400x800 até 400x800, sendo os resultados mostrados acima os piores casos em que o filtro pode gerar. Foi analisado também a aplicação de outros filtros, como o laplaciano, assim como a utilização de outras técnicas de detecção de borda como o DoG (Diferença Gaussiana).

Dado as limitações do filtro e as suas qualidades foi feito uma tabela mostrando todos os casos testados:

	Corretos	Falsos positivos	Falsos negativos	Total de testes
Dimensão Grande	7 (43,75%)	6 (37,50%)	3 (18,75%)	16
Dimensão Pequena	5 (55,55%)	1 (11,11%)	3 (33,33%)	9
Dimensão 400x800	10 (58,82%)	5 (29,41%)	2 (11,76%)	17

### V. CONCLUSÃO

Algoritmos de filtragem de imagens como o do cartoon têm ganhado cada vez mais aplicações de seu uso em processamento de imagens, sendo bastante utilizado em

redes sociais como Instagram, Snapchat, Facebook, Whatsapp, etc. O método apresentado no relatório é de baixa complexidade, seu objetivo é apenas representar um filtro que já existe e é muito utilizado feito apenas para o intuito de adquirir conhecimento.

O código final mostrou-se bastante eficiente partir dos resultados obtidos, pode-se concluir que o algoritmo de cartoonização cumpriu com seu objetivo apesar de algumas limitações do mesmo para alguns tipos específicos de imagem.

### VI. REFERÊNCIAS

[1] A. Hertzmann, "Painterly Rendering with Curved Brush Strokes of Multiple Sizes," Proc. ACM SIGGRAPH '98, pp. 453-460, 1998.

[2] Dade, Kevin, Toonify: Cartoon photo effect aplicattion. Stanford, EUA: Departament of Eletrical Engineering, Stanford University.

[3] Calonder, Michael, et al. Brief: Binary robust independent elementary features. Computer Vision-ECCV 2010. Springer Berlin Heildelberg, 2010. 778-772..