Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação

EEC1515 - VISÃO COMPUTACIONAL

RELATÓRIO

IMPLEMENTAÇÃO DE UM FILTRO DE SUAVIZAÇÃO SELETIVA

Autor: Luís Gabriel Pereira Condados

Professor orientador: Rafael Beserra Gomes

Natal-RN

2020

Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação

EEC1515 - VISÃO COMPUTACIONAL

RELATÓRIO

Relatório apresentado à disciplina de EEC1515- Visão Computacional, correspondente a 1º unidade do semestre 2020.2, sob orientação do **Prof. Rafael Beserra Gomes**.

Autor:Luís Gabriel Pereira Condados

Natal-RN

2020

Sumário

Lista de ilustrações

1 INTRODUÇÃO

Uma operação de pré-processamento muito comum em visão computacional é a de suavização, cujo principal objetivo é a atenuação do ruído presente na imagem através do borramento da mesma. Essa operação geralmente é feita por meio de convolução, exemplos de máscaras convolucionais usadas para suavização são: média e a Gaussiana. Porém dependendo do tamanho da máscara o borramento afeta fortemente a informação de bordas da imagem, tornando-as desfocadas.

O objetivo deste trabalho é apresentar uma solução para o problema da perca de informação de borda devido à forte suavização da imagem. Para isso será adotado uma abordagem de filtragem seletiva, que faz uso da informação do gradiente da imagem para determinar se uma determinada região da imagem é uma região de borda ou não, para com essa informação saber quando pode-se aplicar a suavização mais forte (máscara de dimensão maior) e quando não.

2 METODOLOGIA

Afim de preservar as bordas presentes na imagem é proposto neste trabalho uma técnica de suavização seletiva baseado no módulo do gradiente da imagem. A ideia é aplicar o filtro que causa o maior borramento apenas nos pixeis não pertencentes à nenhum borda, esse procedimento está ilustrado nas Figuras 2a e 2b. Para poder-se aplicar esse tipo de seleção é necessário ter-se o conhecimento de quais pixeis estão em alguma borda, neste trabalho isso é feito através da análise do módulo do grande da imagem, caso esse valor seja maior que um determinado limiar pré-determinado, o pixel é classificado como sendo ou não pixel de borda.

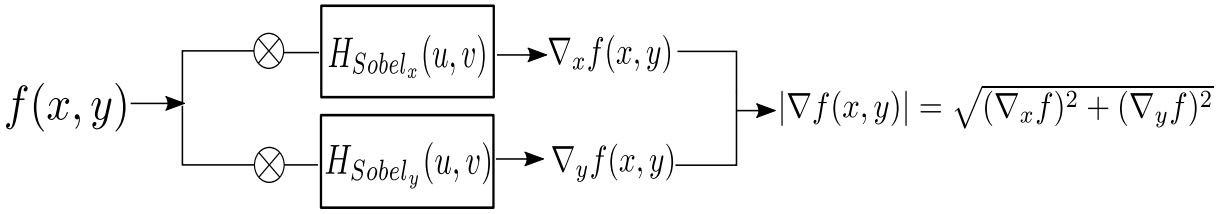


Figura 1 – Esquema de obtenção do módulo do gradiente de uma imagem $f(x, y)$.

A Figura 1 ilustra o procedimento utilizado para calcular o módulo do gradiente da imagem $f(x, y)$, onde $H_{Sobel_x}(u, v)$ e $H_{Sobel_y}(u, v)$ são os operadores de *Sobel* para a direção x e y respectivamente, suas máscaras podem ser vistas na Figura 3. A saída da operação de convolução de $f(x, y)$ com cada uma dessas máscaras, resulta nos equivalentes às derivadas parciais nos sentidos correspondentes, ou de outro modo, as componentes do gradiente de

$f(x, y)$. $|\nabla f(y, x)|$ é o módulo desse gradiente.

$$|\nabla f(x, y)| < threshold$$

$$f(x, y) \otimes \boxed{H_{hard}(u, v)} \longrightarrow g(x, y)$$

(a) Suavização forte para pixeis de não borda.

$$|\nabla f(x, y)| \geq threshold$$

$$f(x, y) \otimes \boxed{H_{soft}(u, v)} \longrightarrow g(x, y)$$

(b) Suavização fraca para pixeis de borda.

Figura 2 – Ilustração do suavizador seletivo.

$H_{Sobel_x}(u, v)$

-1	0	1
-2	0	2
-1	0	1

(a) Operador de *Sobel* na direção x

$H_{Sobel_y}(u, v)$

1	2	1
0	0	0
-1	-2	-1

(b) Operador de *Sobel* na direção y

Figura 3 – Operadores de *Sobel*

Para validar a proposta apresentada, foi testado em diferentes imagens a suavização seletiva, usando-se como suavizador forte um filtro de média de dimensão 10, e um filtro *Gaussiano* de dimensão 3(ver Figura 4) como suavizador fraco (os filtros foram escolhidos de forma a enfatizar o problema do desfoque da imagem) e a qualidade do resultado do processamento é medido por comparação visual entre as imagens fortemente suaviza, entrada e saída.

A implementação foi feita em *C++* e fez-se uso da biblioteca de visão computacional *OpenCV*(1) para facilitar a manipulação de imagens.

$$H_{Gaussian} = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \frac{1}{16}$$

(a) Filtro *Gaussiano* de dimensão 3.

$$H_{average} = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} \frac{1}{25}$$

(b) Filtro de média de dimensão 5.

Figura 4 – Filtros de suavização utilizados.

O programa recebe como argumentos de linha de comando o caminho da imagem que se deseja aplicar a suavização e o limiar usado no teste de pixel de borda.

3 RESULTADOS

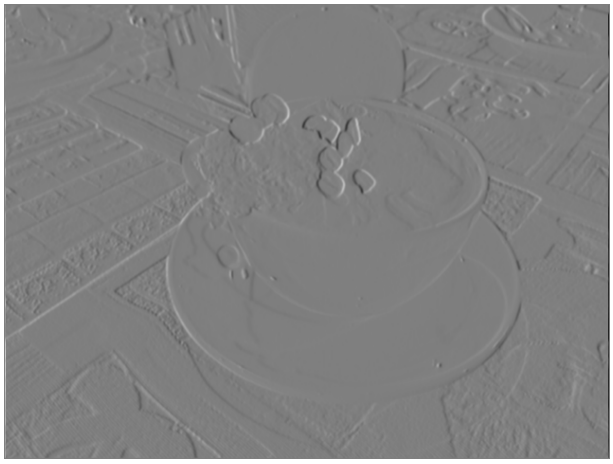
Para testar a implementação, foi utilizado algumas imagens de entrada e diferentes limiares, os resultados estão apresentados a seguir.

(a) Imagem de entrada.

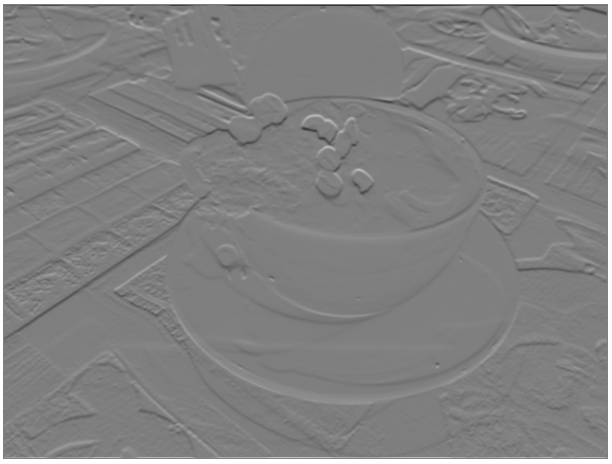
(b) Resultado da suavização seletiva.

(c) Imagem fortemente suavizada.

(d) Imagem levemente suavizada.



(e) Imagem gradiente na direção x.



(f) Imagem gradiente na direção y.

Figura 5 – Experimento 1.Imagem: café, $threshold = 10$.

(a) Imagem de entrada. (b) Resultado da suavização seletiva.

Figura 6 – Experimento 2. Imagem: café, $threshold = 100$.

(a) Imagem de entrada. (b) Resultado da suavização seletiva.

Figura 7 – Experimento 3. Imagem: café, $threshold = 1000$.

Os experimentos de 1 à 3 utilizaram a mesma imagem de entrada (café), por isso só é mostrado uma vez as imagens da suavização forte, suavização fraca e as imagens de gradiente, pois elas só dependem da imagem de entrada. Esses experimentos tiveram como diferença apenas o limiar. É perceptível que a medida que aumenta-se o limiar a imagem resultante torna-se mais desfocada, ou seja, perde-se mais as informações das bordas.

(a) Imagem de entrada.

(b) Resultado da suavização seletiva.

(c) Imagem fortemente suavizada.

(d) Imagem levemente suavizada.

(e) Imagem gradiente na direção x.

(f) Imagem gradiente na direção y.

Figura 8 – Experimento 5. Imagem: Totó, $threshold = 10$

(a) Imagem de entrada. (b) Resultado da suavização seletiva.

Figura 9 – Experimento 5: Imagem: Totó, $threshold = 100$

(a) Imagem de entrada. (b) Resultado da suavização seletiva.

Figura 10 – Experimento 6. Imagem: Totó, $threshold = 1000$

Os experimentos de 4 à 6 utilizaram a mesma imagem de entrada (Totó), por isso só foi mostrado uma vez as imagens da suavização forte, suavização fraca e as imagens de gradiente, pois elas só dependem da imagem de entrada. E assim como os experimentos anteriores, teve o parâmetro de limiar alterado, sendo eles 10, 100 e 1000. O mesmo comportamento dos experimentos com a imagem do *café*, é apresentado aqui, ou seja, a imagem perde a nitidez e aumenta o desfoco conforme aumenta-se o limiar.

4 CONCLUSÃO

Os resultados apresentados nos experimentos foram como o esperado. Ao se aumentar o limiar torna-se mais rigoroso o teste para classificar se um pixel será considerado como parte de uma borda, fazendo com que o mesmo seja filtrado com o suavizador mais forte. O caso contrário também ocorreu, o que quer dizer que se o limiar for baixo, o filtro tende

a usar mais a suavização leve.

Alguns efeitos extras, que podem ser indesejáveis, foram visíveis nas imagens de saída, foram eles: bordas que não haviam antes na imagem (observe as Figuras 5b e 8b por exemplo) e um efeito "ghost"/sombreado nas proximidades das bordas (por exemplo, as Figuras 7a e 10a). O primeiro efeito, o de borda, é causado devido a maneira como foi implementado a operação de convolução, a operação causa uma redução na dimensão na imagem de forma proporcional ao tamanho da máscara utilizada, isso é bem evidentes nas imagens resultante do borramento forte (ver Figura 8c como exemplo), pois é utilizado um filtro de dimensão 10, isso faz com que a imagem resultante da convolução tenha uma redução de 5pixeis nas suas bordas, porém para manter a resolução igual a entrada, esses pixeis faltantes são substituídos por pixeis pretos. Existem algumas maneiras de se tratar o efeito da redução no tamanho da imagem, uma maneira é aumentar a imagem com replicas dela mesma antes de realizar a convolução.

O outro efeito notável é o de sombreamento nas proximidades das bordas. Isso é devido à natureza do filtro de gradiente, pois ele realça toda a proximidade de uma região de borda e não apenas a borda em si, isso fez com algumas regiões nas proximidades também fossem classificadas como borda. Uma maneira que contornaria esse problema, seria a utilização do detector de bordas de *Canny*, pois consegue realça-las com precisão de 1 pixel.

Referências

1 ITSEEZ. *Open Source Computer Vision Library*. 2015. <https://github.com/itseez/opencv>. 2

5 ANEXO 1

Código fonte: <https://github.com/Gabriellgpc/ComputerVision_DIM0141/blob/master/unidade1/tarefa2/q1.cpp>