

Relatório do Projeto Demonstrativo 4: Detecção de Bordas

Aluno: Jessé Barreto de Barros

Matrícula: 17/0067033

Sistemas Mecatrônicos - PPMEC-UnB

Disciplina: Visão Computacional

Turma: A

Data: 20/05/2017

1. Objetivos

O principal objetivo deste projeto demonstrativo é a exploração dos aspectos envolvidos no desenvolvimento e na utilização de algoritmos para a detecção de bordas em imagens e a análise comparativa entre eles.

2. Introdução

2.1. Detectores de Bordas

Conforme [1], na visão computacional as bordas podem ser identificadas como regiões de contorno que podem ser geradas por: a) diferenças bruscas no nível de brilho entre diferentes objetos, e.g. um objeto com cor clara sobre um fundo escuro (Figura ??); b) um objeto com regiões com diferentes reflectâncias e cores, e.g. as listras de uma zebra; c) diferenças nos formatos da superfície de um objeto que geram diferenças de brilho.

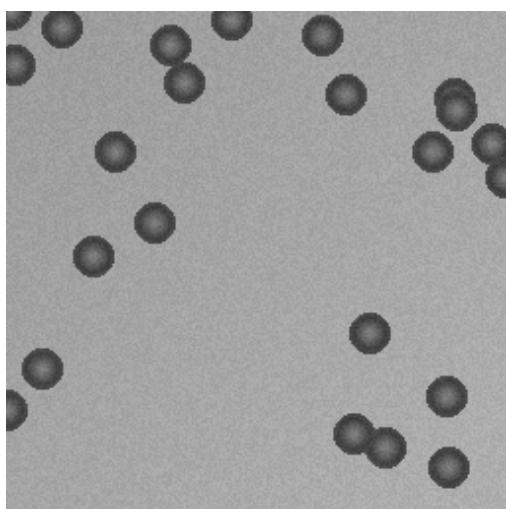


Figura 1: Círculos cujas bordas são de fácil identificação.
(Extraído das notas de aula.)

Uma forma de identificar as mudanças drásticas de brilho presentes nas bordas é a magnitude do gradiente de brilho na região da borda. Essa abordagem é utilizada pelas técnicas de Sobel, Laplaciano e Canny.

2.1.1 Sobel

O operador de Sobel é uma técnica para identificar as bordas em uma imagem que funciona como um operação de diferenciação discreta que calcula um valor aproximado do gradiente da intensidade luminosa da imagem através da convolução da imagem com um *kernel* que funciona como um filtro para obtenção do gradiente nas direções horizontal e vertical, nas equações 1 e 2 há a representação 3x3 dessas janelas.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (1)$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

2.1.2 Laplace

O operador Laplaciano é uma técnica baseada em um operador derivativo de segunda ordem para identificar pontos de máximo no gradiente da função de intensidade da imagem. O operador laplaciano nas imagens funciona a partir da convolução de um *kernel* pela imagem. Um exemplo de template de um laplaciano de tamanho 3x3 está representado na equação 3.

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (3)$$

2.1.3 Canny

O operador de Canny funciona como um detector para uma maior variedade de imagens e que na verdade é um operador de diferenciação de primeira ordem sobre um template de suavização gaussiana. O operador de Canny possui os seguintes estágios:

1. Suavização da imagem através de uma gaussiana.
2. Calculo da magnitude e direção do gradiente para cada pixel da imagem.
3. Limiarização da magnitude do gradiente calculado através de dois limiares para supressão de bordas "fracas".

3. Metodologia

O projeto possui o seguintes requisitos:

- **Requisito 1:** A partir de referências da biblioteca OpenCV desenvolva uma aplicação que a partir de uma imagem de entrada compute os três tipos de detectores de bordas e com os resultados dos detectores de bordas aplique uma técnica de limiarização para obter imagens binárias com as bordas dos objetos presentes na imagem. Para cada saída dos detectores de bordas salve em disco os resultados.
- **Requisito 2:** Execute a aplicação desenvolvida no requisito 1 para as imagens presentes na pasta *srcs*.
- **Requisito 3:** Compare os resultados das bordas obtidas com as respectivas imagens de *Ground Truth* fornecidas.
- **Requisito 4:** Para cada imagem de saída dos detectores de bordas propostos e utilizando as imagens de *Ground Truth* realize o cálculo de uma pontuação para a comparação dos resultados de cada detector, dado que as bordas possuem um valor de pixel nulo. O cálculo dessa pontuação funciona dada a equação 4. Crie uma tabela da pontuação de cada detector e cada imagem e as compare.

$$Score = \frac{Hits}{Hits + Misses} \quad (4)$$

Dado que: a) *Hits* - é a quantidade de pixels idênticos entre a imagem de *Ground Truth* e a imagem binária de saída do detector de bordas. b) *Misses* - é a quantidade de pixels diferentes entre a a imagem de *Ground Truth* e a imagem binária de saída do detector de bordas.

A aplicação implementada em C++ utilizando o OpenCV abre todas as imagens presentes na pasta *srcs* e

computa as suas bordas através dos três detectores de bordas: a) Sobel; b) Canny; c) Laplace. Com os resultados é utilizado uma operação de limiarização para binarizar as saídas dos detectores e realçar as bordas dados um valor limiar de intensidade dos pixels de 100. Depois salva cada uma das imagens em uma pasta chamada *dsts* e calcula a pontuação dos detectores da imagem.

Para cada imagem de entrada a saída é apresentada em tela e a aplicação aguarda o usuário pressionar qualquer tecla do teclado para continuar.

4. Resultados

Dada as imagens fonte disponibilizadas presentes na Figura 2 e as imagens utilizadas como *Ground Truth* presentes na Figura 3.

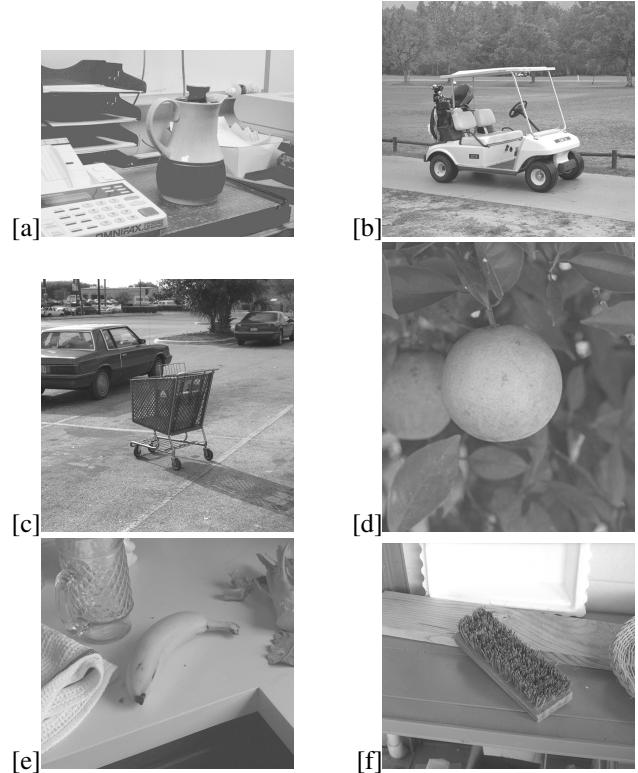


Figura 2: Imagens utilizadas na aplicação. (a) Imagem 46, (b) Imagem 140, (c) Imagem 208, (d) Imagem 212, (e) Imagem 217, (f) Imagem 221.

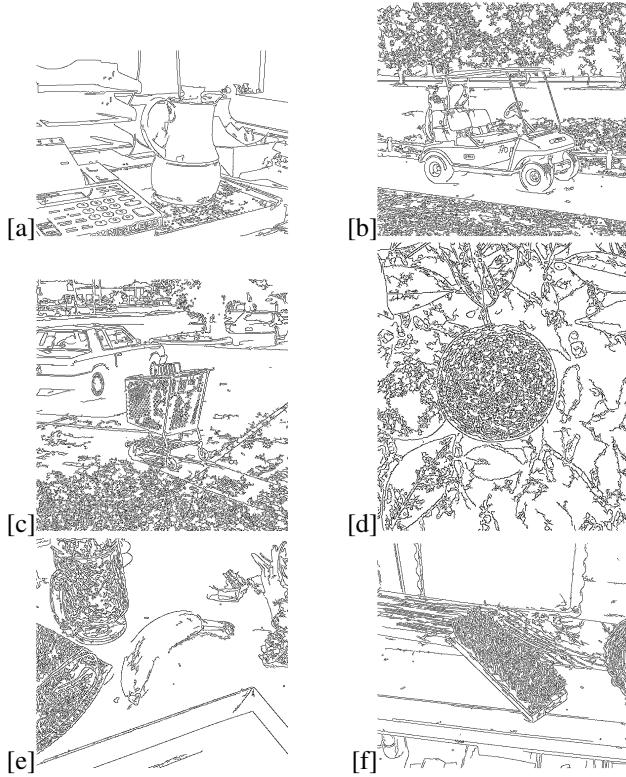


Figura 3: Imagens utilizadas na aplicação como *Ground Truth*. (a) Imagem 46, (b) Imagem 140, (c) Imagem 208, (d) Imagem 212, (e) Imagem 217, (f) Imagem 221.

Os resultados para o detector de Sobel estão presentes na Figura 4.

Os resultados para o detector Laplace estão presentes na Figura 5.

Os resultados para o detector Canny estão presentes na Figura 6.

Na tabela 1 está presente as pontuações, em porcentagem, de cada detector de bordas para cada imagem se comparados com as imagens de *Ground Truth*.

Imagen	46	140	208	212	217	221
Sobel	92.70%	83.40%	84.98%	82.61%	90.16%	91.02%
Canny	96.26%	87.42%	89.82%	83.04%	91.68%	92.85%
Laplace	86.69%	73.01%	66.19%	82.16%	87.92%	85.43%

Tabela 1: Tabela com as pontuações de cada técnica de detecção de bordas.

5. Discussão e Conclusões

O projeto demonstrativo concluiu os objetos propostos e com esse projeto foi possível aprender mais sobre as técnicas disponíveis para a detecção de bordas em imagens e as suas características individuais.

Percebeu-se que o limiar escolhido para a binarização

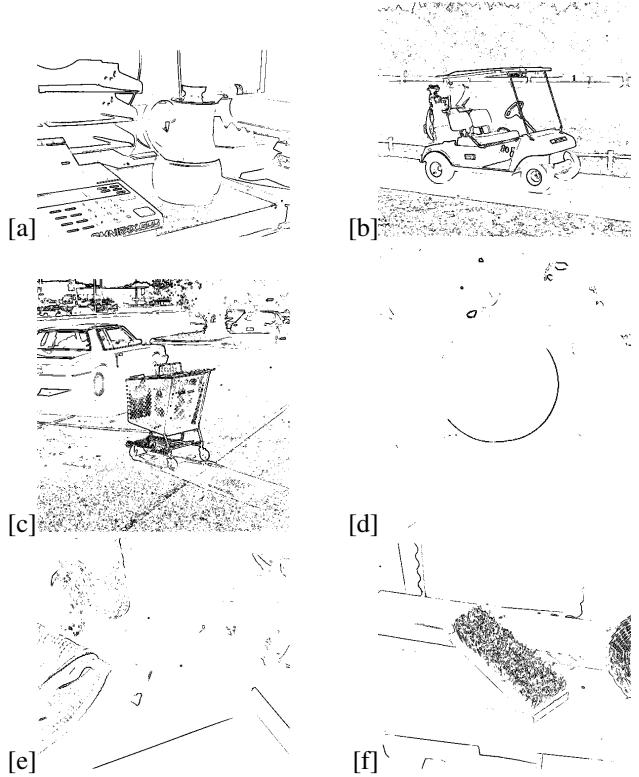


Figura 4: Imagens obtidas com a aplicação do operador de Sobel. a. (a) Imagem 46, (b) Imagem 140, (c) Imagem 208, (d) Imagem 212, (e) Imagem 217, (f) Imagem 221.

influencia a quantidade de ruídos na imagem de bordas. Isso se deve devido a "passagem" de bordas "fracas", ou seja, bordas cujo valor absoluto dos seus gradientes não é alto e, com exceção da imagem 212 ??, reduz a pontuação das técnicas de detecção de bordas.

Todos os detectores de Sobel, de Laplace e de Canny recebem como parâmetro o tamanho da janela que será utilizado na convolução. Janelas maiores reduzem a sensibilidade aos ruídos, porém deixam de identificar bordas com magnitudes menores de gradiente. Isso reduz a quantidade de bordas identificadas na imagem de bordas. Janelas de *template* menores são mais sensíveis a ruídos, mas identificam mais bordas.

A escolha do tamanho da janela do *template* depende do conteúdo da imagem e, geralmente, acaba sendo escolhida caso a caso.

O Laplaciano por ser um operador que funciona com derivadas de segunda ordem é o mais sensível de todos. Esse detector consegue encontrar muitas bordas porém é muito sensível aos ruídos. Isso explica o porquê que a média da pontuação do detector de Laplace possui a pior média entre os detectores de bordas.

O operador de Canny possui os melhores resultados e o

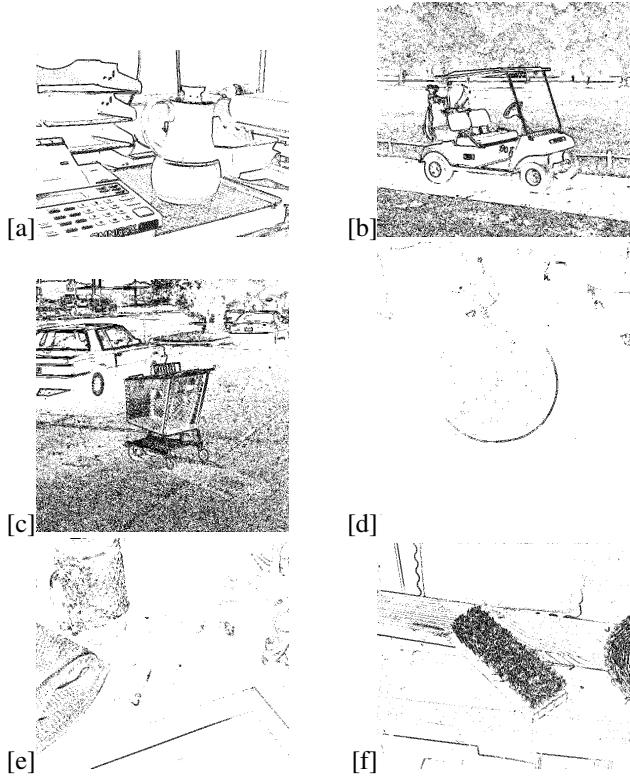


Figura 5: Imagens obtidas com o detector de bordas Laplace. (a) Imagem 46, (b) Imagem 140, (c) Imagem 208, (d) Imagem 212, (e) Imagem 217, (f) Imagem 221..

valor dos limiares utilizados foram de 100 e 200.

6. Bibliografia

Referências

- [1] D. A. Forsyth and J. Ponce. A modern approach.

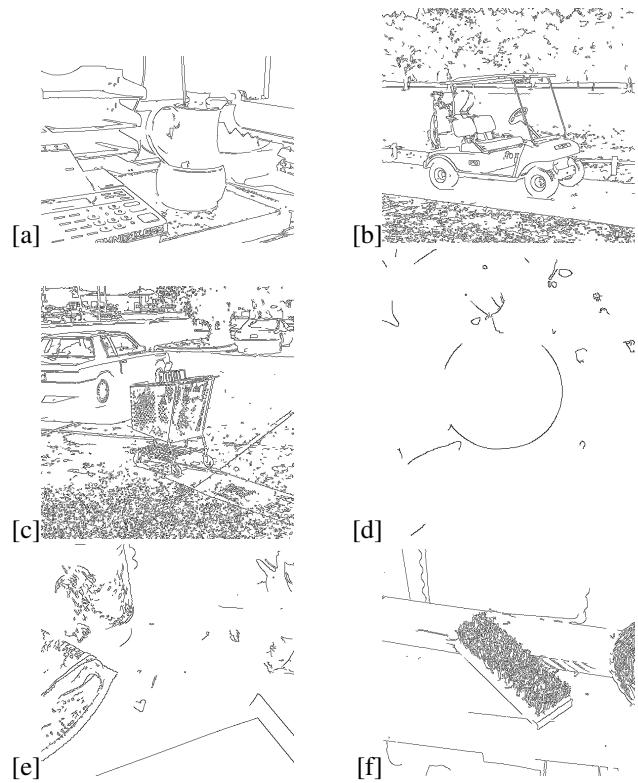


Figura 6: Imagens obtidas com o detector de Canny. (a) Imagem 46, (b) Imagem 140, (c) Imagem 208, (d) Imagem 212, (e) Imagem 217, (f) Imagem 221.