UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

GABRIEL BALOTIN ZEFERINO

# TÉCNICAS DE REMOÇÃO DE FUNDO EM IMAGENS ESTÉREO OBTIDAS A PARTIR DE CÂMERAS EM MOVIMENTO

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2018

GABRIEL BALOTIN ZEFERINO

# TÉCNICAS DE REMOÇÃO DE FUNDO EM IMAGENS ESTÉREO OBTIDAS A PARTIR DE CÂMERAS EM MOVIMENTO

Trabalho de Conclusão de Curso, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro da Computação.

Orientador: Prof. Dr. Pablo Gauterio Cavalcanti

PATO BRANCO

2018

**Ministério da Educação**
**Universidade Tecnológica Federal do Paraná**
**Câmpus Pato Branco**
**Departamento Acadêmico de Informática**
**Curso de Engenharia de Computação**

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

## TERMO DE APROVAÇÃO

Às 14 horas do dia 13 de dezembro de 2018, na sala V006 da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Pablo Gauterio Cavalcanti (orientador), Dalcimar Casanova e Marco Antonio de Castro Barbosa para avaliar o trabalho de conclusão de curso com o título **Técnicas de remoção de fundo em imagens estéreo obtidas a partir de câmeras em movimento**, do aluno **Gabriel Balotin Zeferino**, matrícula 01376535, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

_____
Prof. Pablo Gauterio Cavalcanti
Orientador (UTFPR)

_____
Prof. Dalcimar Casanova
(UTFPR)

_____
Prof. Marco Antonio de Castro Barbosa
(UTFPR)

_____
Profa. Beatriz Terezinha Borsoi
Coordenador de TCC

_____
Prof. Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

## RESUMO

ZEFERINO, Gabriel Balotin. Técnicas de remoção de fundo em imagens estéreo obtidas a partir de câmeras em movimento. 2018. 43f. Monografia (Trabalho de Conclusão de Curso 2) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

Normalmente, durante a aquisição de uma ou mais imagens, diversos são os conteúdos registrados além do objeto ou região de interesse. Uma aplicação de visão computacional pode ter seu foco específico, mas as imagens obtidas incluirão, além deste foco, diversos outros objetos ou texturas que estiverem posicionados na mesma cena. Partindo do princípio que o objeto ou região de interesse de uma cena está no *foreground* da imagem, segmentar os pixels desta imagem entre aqueles pertencentes ao fundo (*background*) e aqueles de fato pertencentes ao *foreground* torna-se uma tarefa fundamental para qualquer sistema de visão computacional. Neste trabalho, é analisada a abordagem de segmentação da imagem utilizando-se dos conceitos de visão estéreo, ou seja, obtendo imagens a partir de duas (ou mais) câmeras e, a partir do processamento das mesmas, identificar os pixels pertencentes aos objetos mais distantes das câmeras e, portanto, pertencentes ao fundo da cena. Além dos conceitos matemáticas de visão estéreo, são apresentados detalhes a respeito de pré-processamento das imagens, extração de pontos característicos, correspondência esparsa e outros algoritmos necessários para esta tarefa.

**Palavras-chave**: Visão estéreo, Extração de fundo, Processamento de Imagens.

## ABSTRACT

ZEFERINO, Gabriel Balotin. Background subtraction tools for stereo images obtained from moving cameras. 2018. 43f. Monografia (Trabalho de Conclusão de Curso 2) - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

Usually, during the acquisition of one or more images, several structures are recorded beyond the object or region of interest. A computer vision application may have its specific focus, but the images obtained will include, in addition to this focus, several other objects or textures that are in the same scene. Assuming that the object or region of interest is in the image foreground, to segment the pixels of this image between those belonging to the background and those really belonging to the foreground becomes a fundamental task for any computer vision system. In this work, the segmentation approach of the image is analyzed using the concepts of stereo vision, i.e. obtaining images from two (or more) cameras and, after processing them, identify the pixels belonging to the objects more distant from the cameras and consequently belonging to the background of the scene. In addition to the mathematical concepts of stereo vision, details are presented regarding image preprocessing, local descriptors, sparse matching and other algorithms required for this task.



**Keywords**: Stereo vision, Background subtraction, Image Processing.

## LISTA DE FIGURAS

**SUMÁRIO**

# 1 INTRODUÇÃO

Com o constante aumento da automatização de equipamentos, a aquisição de dados, utilizando sensores e câmeras, se torna maior a cada instante. A necessidade de criar técnicas para auxiliar e evitar erros nesta etapa é de grande importância para o processo.

Considerando a aquisição de dados a partir de imagem (ou imagens), uma das etapas primordiais é dada pela segmentação, ou seja, um procedimento em que a imagem é separada em subpartes baseada na região ou objeto de interesse (GONZALEZ; WOODS, 2010).

Normalmente, a região ou objeto de interesse é parte do *foreground* da imagem, subpartes que possuem menor distância em relação à câmera do que a maior parte do ambiente sendo fotografado ou filmado. As demais subpartes (i.e., aquelas distantes da câmera) são consideradas *background*, e a sua correta remoção é tarefa importante para o processo de segmentação da imagem. Para esta etapa, então, existem diversos métodos já desenvolvidos, sendo o modelo de *Mixture of gaussian* (MOG) o mais popular (BOULMERKA; ALLILI, 2015).

Porém, ao se adquirir imagens de cenas reais, muitos problemas que podem dificultar esta etapa de remoção de *background* se tornam evidentes, como: mudança de iluminação, camuflagem, câmera *jitter*, *background* dinâmico, sombreamento, etc (BRAHAM *et al.*, 2017). Se considerarmos, ainda, que as câmeras podem estar em movimento, um dos maiores desafios encontrados é o *background* dinâmico, pois pode haver mudança na direção das câmeras em cada um dos *frames* subsequentes, gerando diversos problemas para a precisa remoção.

Diversas técnicas são combinadas a métodos clássicos para minimizar os problemas encontrados em cenas reais, dentre estes a profundidade. Uma das maneiras usuais para a obtenção da profundidade é a simulação da visão humana (VIEIRA *et al.*, 2017). Nesse tipo de simulação, o mapa de profundidade é geralmente obtido a partir de duas imagens bidimensionais (visão estéreo) (VIEIRA *et al.*, 2017).

## 1.1 OBJETIVOS

Nesta sessão, será apresentado o objetivo geral e objetivos específicos que o trabalho proposto deverá cumprir.

### 1.1.1 OBJETIVO GERAL

- Implementar um método de remoção de *background* mesclando técnicas clássicas com as características extraídas de imagens estéreo.

### 1.1.2 OBJETIVOS ESPECÍFICOS

- Modelar um ambiente filmado a partir de um sistema de câmeras estéreo em movimento e, a partir disto, eliminar os objetos mais distantes (ou seja, remoção de *background*).
- Realizar experimentos com filtros lineares e não-lineares para pré-processamento das imagens.
- Comparar resultados obtidos com a aplicação de diversas técnicas de extração de pontos característicos, a fim de identificar o que possui melhor robustez para *matching* entre pontos em cenas reais.
- Identificar os principais problemas do método desenvolvido quando aplicado às câmeras em movimento para ambientes reais e realizar experimentos visando comparar possíveis soluções.

## 1.2 JUSTIFICATIVA

Diversas são as técnicas de remoção de fundo já desenvolvidas e que geralmente obtém bons resultados para câmeras simples (não-estéreo) e estáticas. Entretanto, estas técnicas costumam apresentar significativos defeitos se aplicadas à cenas reais, principalmente se considerarmos o fundo dinâmico resultante de câmeras em movimento.

A remoção de fundo em cenas reais com fundo dinâmico pode ainda ser afetada por diversos problemas, como sombreamento, mudança de iluminação, ruídos, ajustes automáticos de câmera (balanço de branco, contraste, etc). Estes problemas afetam significativamente a metodologia clássica, mas combinando técnicas

pré-processamento, extração de características e reconstrução estéreo, foi possível adquirir respostas aceitáveis para o problema proposto, além de perceber diversos problemas no método e expor possíveis soluções para melhoria do mesmo.

# 2 REFERENCIAL TEÓRICO

Nesta capítulo serão apresentados todos os métodos necessários para as etapas de pré-processamento, sendo, transformações de intensidade e filtragem espacial. Assim como, fundamentos básicos da visão estéreo e técnicas necessárias para seu desenvolvimento, como as especificações de extração de pontos característicos e a denotação por correspondência esparsa. Por fim, serão apresentados os princípios básicos de extração de fundo e o método clássico, mistura de gaussianas (MOG) que será utilizado como padrão para comparação dos resultados.

## 2.1 TRANSFORMAÇÕES DE INTENSIDADE E FILTRAGEM ESPACIAL

Existem diversas formas de manipular imagens digitais, estas divididas em dois grupos: domínio espacial e domínio da transformada (GONZALEZ; WOODS, 2010). As manipulação de transformas, geralmente, utilizam transformada de Fourier discreta, para a representação da imagem, e em cima desta, aplicam filtros no domínio da frequência. Na etapa espacial serão apresentadas todas as técnicas que derivam os métodos espaciais, que se referem às transformações que são aplicadas diretamente às imagens, ou seja, na manipulação direta dos *pixels*.

O processamento espacial pode ser categorizado em: transformações de intensidade e filtragem espacial. Sendo que a primeira é aplicada individualmente nos *pixels* da imagem para manipulação de contraste e limiarização. Por outro lado, a filtragem espacial lida com operações que atuam nas vizinhanças de cada *pixels* (GONZALEZ; WOODS, 2010).

### 2.1.1 PROCESSAMENTO DE HISTOGRAMA

Histograma é a representação gráfica em colunas de um conjunto de dados, estes assumindo valores de distribuição variados, dividido em diversas classes. Várias técnicas de processamento no domínio espacial são baseadas em histogramas que podem ser utilizados para realçar imagens, alterar contraste e outras operações. Além dessa utilização, histogramas fornecem informações estatísticas sobre a imagem, bastante úteis em outras aplicações de processamento de imagem, como compreensão

e segmentação (GONZALEZ; WOODS, 2010).

Pode-se observar que imagens escuras possuem em seu histograma uma concentração de *pixels* em valores baixos, enquanto imagens claras possuem suas concentrações nos valores altos (GONZALEZ; WOODS, 2010). Uma imagem de baixo contraste possui seu histograma estreito, geralmente, concentrado no centro da imagem. Por sua vez, imagens de alto contraste possuem uma distribuição que tende a ocupar todos os valores possíveis de *pixels* sendo distribuídas de maneira uniforme (GONZALEZ; WOODS, 2010). Essas características podem ser observadas na Figura 1.

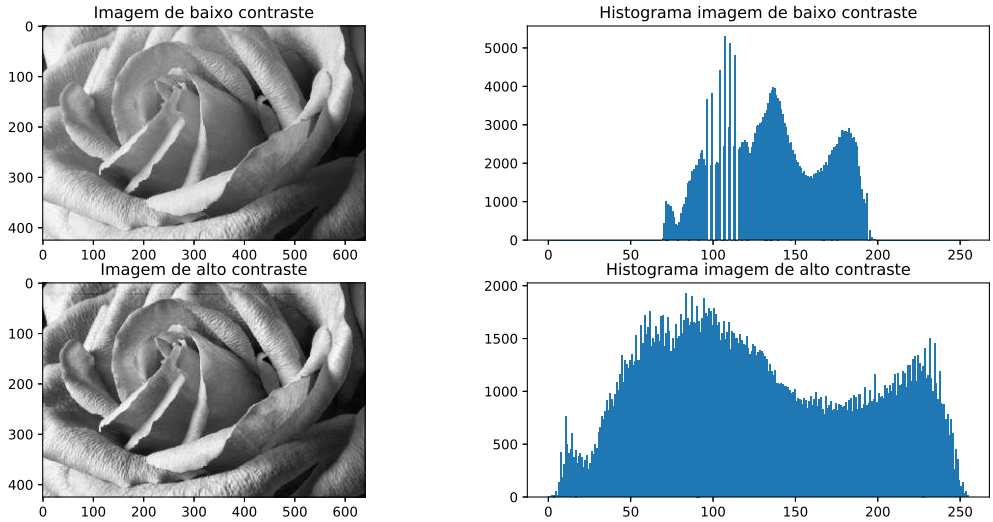


**Figura 1: Dois tipos básicos de imagens: alto e baixo contraste e seus respectivos histogramas**

### 2.1.1.1 EQUALIZAÇÃO DE HISTOGRAMA

Para imagens digitais, a equalização de histograma pode ser decomposta de forma discreta, desta forma, podem ser utilizadas probabilidades (oriundas dos valores do histograma) e somatórios (GONZALEZ; WOODS, 2010). Assim a probabilidade de ocorrência de uma intensidade $r_k$ é expressa da seguinte forma:

$$p_r(r_k) = \frac{n_k}{MN} \quad K = 0, 1, 2...L-1 \tag{1}$$

sendo que $MN$ é o numero total de *pixels* da imagem, $n_k$ é o numero de *pixels* com intensidade $r_k$ e L é o numero é o número de intensidades possíveis na imagem. Podendo, assim, modelar a seguinte equação, que descreve a equalização discreta

de um histograma:

$$s_k = (L-1)\sum_{j=0}^{k} p_r(r_j) \quad k = 0, 1, 2...L-1 \tag{2}$$

Assim, podemos utilizar a equação para processar cada pixel da imagem, sendo $r_k$ o valor original na imagem a ser processada e o valor de saída $s_k$ o correspondente pós-processamento que substituirá o valor de $r_k$ (GONZALEZ; WOODS, 2010). A Figura 2 demonstra a utilização da equalização do histograma para uma imagem.

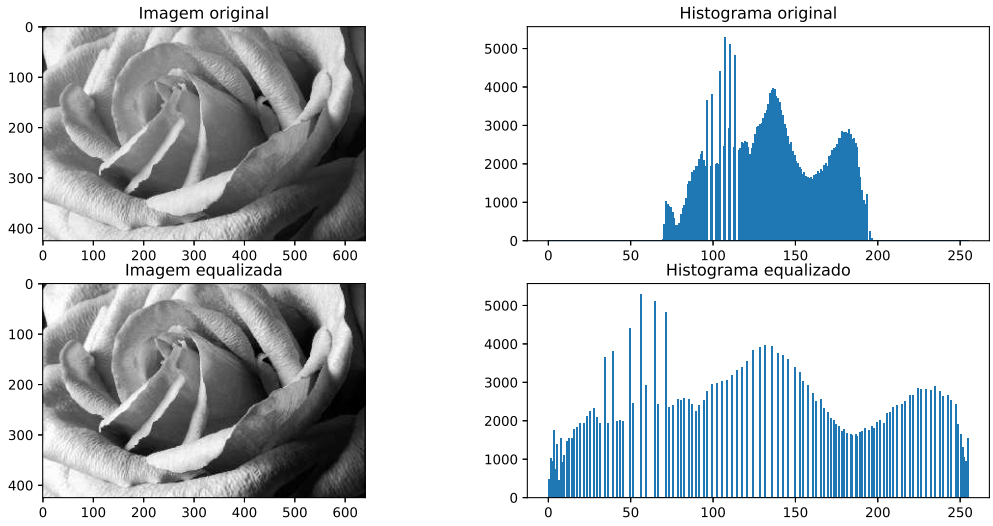


**Figura 2: Demonstração de utilização de equalização de histograma. Fonte: Autoria própria**

### 2.1.2 FUNDAMENTOS DE FILTRAGEM ESPACIAL

A filtragem espacial consiste em fazer uma operação em uma pequena área sobre a localização de um *pixel*, denominada máscara, assim criando um novo *pixel* com as coordenadas do centro da área e cujo valor é o resultado da operação de filtragem (GONZALEZ; WOODS, 2010). Se a operação utilizada sobre os *pixels* da imagem for linear, o filtro é chamado de *filtro espacial linear*. Caso contrário, o filtro é chamado de *filtro espacial não-linear*.

Este fundamento pode ser aplicado utilizando dois conceitos: convolução ou correlação (GONZALEZ; WOODS, 2010). Desta forma, a filtragem espacial com correlação transita a máscara pela imagem e calcula a soma do produto dos pesos pelos respectivos valores em cada posição. Já a filtragem baseada pela convolução segue o mesmo princípio da correlação, porém a máscara deve ser rotacionada em $180^o$.

Para a representação de uma máscara $S$ para um filtro genérico $M$x$N$, respectivamente os valores de linhas e colunas da máscara, com valores de peso $w$ (GONZALEZ; WOODS, 2010), pode-se representar a máscara da forma:

$$S = \sum_{k=1}^{MN} w_k z_k \tag{3}$$

$$M = \mathbf{w}^T \mathbf{z} \tag{4}$$

Para filtros espaciais lineares, é, muitas vezes, utilizada uma função de duas variáveis para a obtenção dos pesos $w$. Para a geração dessa máscara é feita uma amostragem ao redor de seu centro e ajustada a dispersão de acordo com o tamanho desejado (GONZALEZ; WOODS, 2010).

A geração de filtros não-lineares utiliza operações não definidas como funções, desta forma, seus filtros dependem da dimensão do filtro a ser projetado e a operação desejada para a máscara (GONZALEZ; WOODS, 2010). Um exemplo simples é a criação de um filtro de *mediana* com dimensão de $5$x$5$, esta sendo uma estatística não-linear.

### 2.1.3 FILTROS DE SUAVIZAÇÃO

Em diversos casos, imagens digitais são suscetíveis a ruídos ou erros de captura, nestes casos é utilizado filtro de suavização, que geralmente está associados à remoção de ruído e borramento (GONZALEZ; WOODS, 2010). Este tipo de filtro está dividido em lineares, sendo derivados de operações lineares e funções, e não-lineares, que são adquiridos de funções estatísticas.

#### 2.1.3.1 FILTROS LINEARES

Filtros de suavização lineares são os que utilizam a operação de correlação ou convolução, definindo uma máscara de $M$x$N$ derivada de uma função linear para ponderamento do vetor $w$. Os filtros lineares geralmente são chamados por *filtros de média* e também por *filtros passa-baixa* (GONZALEZ; WOODS, 2010). Sendo expresso da seguinte maneira:

$$R = \frac{1}{MN} \sum_{i=1}^{MN} w_i \tag{5}$$

Dentre as várias funções utilizadas para ponderamento, a função gaussiana é a mais popular, sendo possível a criação de um filtro do tamanho desejado variando os parâmetros da função (PETROU; BOSDOGIANNI, 1999), definida da seguinte maneira:

$$h(x, y) = e^{\frac{x^2+y^2}{2\sigma^2}} \tag{6}$$

sendo que $\sigma$ é o desvio padrão e os valores $x$ e $y$ são inteiros dispersos no centro da gaussiana para geração da amostragem na máscara. A Figura 3 demonstra a utilização de diversos filtros baseados na gaussiana, variando $\sigma$.

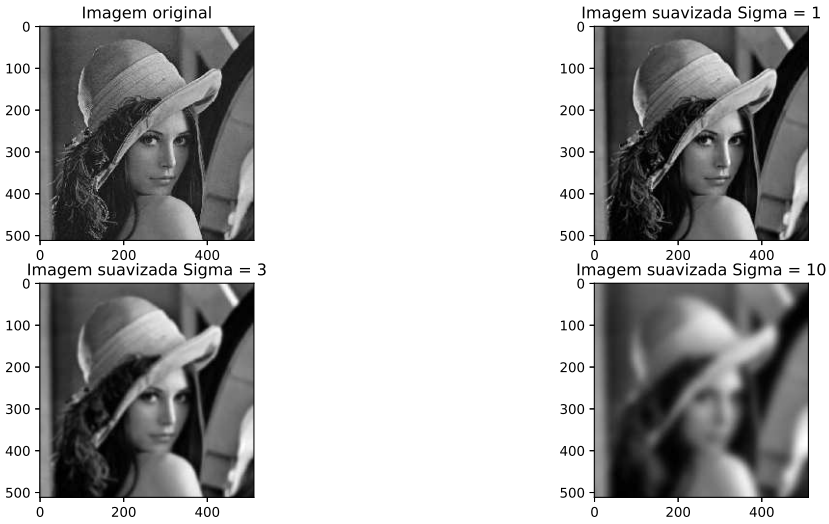


**Figura 3: Demonstração de filtro de suavização gaussiano com sigma: 1, 3 e 10. Fonte: Autoria própria**

### 2.1.3.2 FILTROS ESTATÍSTICOS

São filtros espaciais não-lineares, cuja resposta se baseia na ordenação dos elementos e aplicação de um operação estatística (GONZALEZ; WOODS, 2010). Dada uma imagem, o filtro será passado por todos os *pixels* redefinindo seu valor baseado na operação e o tamanho da área.

O filtro estatístico mais utilizado é o da mediana (GONZALEZ; WOODS, 2010), uma vez que este é robusto contra ruídos aleatórios e apresenta uma suavização menor que os apresentados por filtros lineares. A Figura 4 mostra o resultado da aplicação do filtro da mediana para uma imagem que apresenta ruído aleatório. Porém, o filtro da mediana não é o único deste modelo a ser utilizado, pois em algumas aplicações são utilizadas as operações de máximo e mínimo, o primeiro para a detecção de pontos claros na imagem e o segundo para os pontos escuros (GONZALEZ; WOODS,

2010).

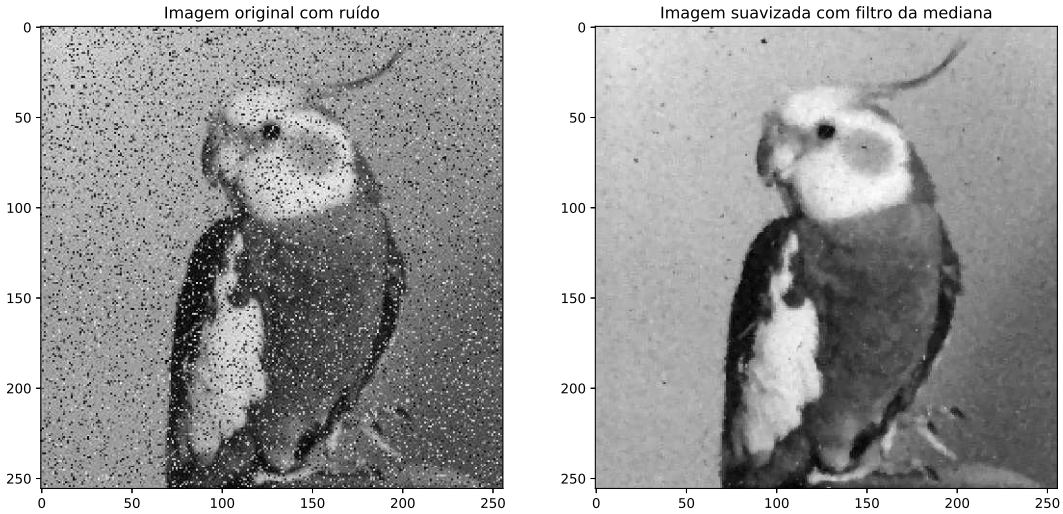


**Figura 4: Demonstração da utilização do filtro da mediana em uma imagem com ruído aleatório. Fonte: Autoria própria**

### 2.1.4 FILTROS DE AGUÇAMENTO

Para melhorar a nitidez de uma imagem, geralmente são utilizados filtros de *aguçamento*, para destacar bordas e detalhes finos da imagem, aumentando a diferença entre os *pixels* vizinhos (GONZALEZ; WOODS, 2010). Este filtro é análogo à derivação da imagem, sendo os valores de bordas e ruídos ampliados e regiões homogêneas atenuadas (GONZALEZ; WOODS, 2010).

Filtros de aguçamentos são baseados na primeira e segunda derivada (GONZALEZ; WOODS 2010), em que máscaras são derivadas destas operações e utilizadas no princípio de correlação ou convolução, o mesmo ocorre com os filtros de suavização linear.
A derivada de uma função digital pode ser obtida a partir da diferença entre os valores discretos da função (GONZALEZ; WOODS, 2010), esta podendo ser definida como:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{7}$$

Da mesma forma, pode-se descrever a segunda derivada como a diferença entre os valores vizinhos da primeira derivada, sendo definida como:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{8}$$

#### 2.1.4.1 FILTRO LAPLACIANO

O operador isótropo (invariantes a rotação) mais simples é o laplaciano que para uma imagem definida como $f(x, y)$ é definido como:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \tag{9}$$

Desta forma, é possível expressar o laplaciano discreto da seguinte forma:

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \tag{10}$$

Assim pode-se definir um filtro $3$x$3$ laplaciano, um filtro linear com operador diferencial, que realça descontinuidades, como bordas e ruídos, e atenua áreas com regiões uniformes (GONZALEZ; WOODS, 2010). Podendo assim, derivar um filtro de aguçamento baseado no filtro laplaciano, definido como:

$$g(x, y) = f(x, y) + \nabla^2 f(x, y) \tag{11}$$

em que $f(x, y)$ é a imagem de entrada e $g(x, y)$ é a resposta do filtro de aguçamento baseado no laplaciano. A Figura 5 demonstra a utilização de um filtro de aguçamento $3$x$3$ baseado no gradiente da segunda derivada.

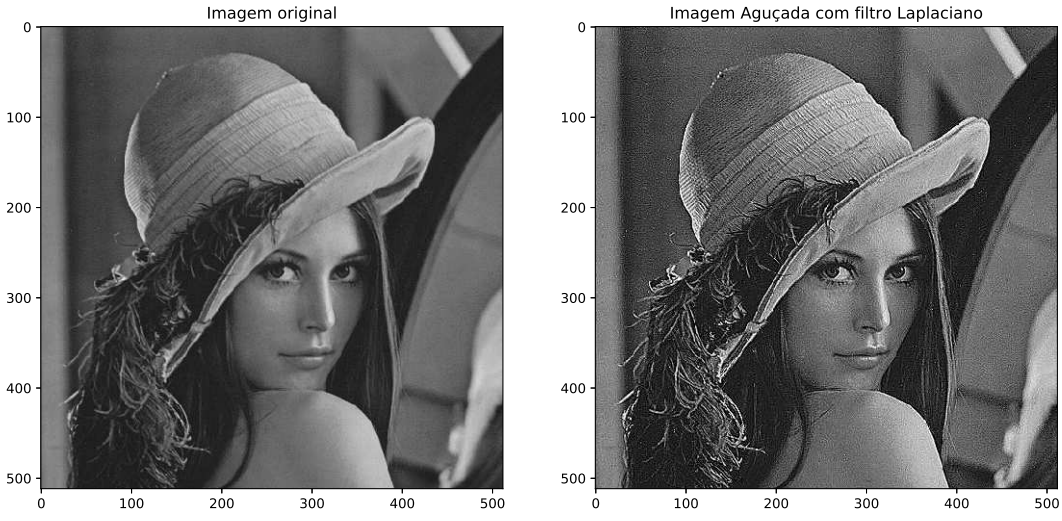


**Figura 5: Demonstração da utilização do filtro de Laplace para aguçamento. Fonte: Autoria própria**

#### 2.1.4.2 FILTRO DO GRADIENTE

Para processamento de imagens, as derivadas de primeira ordem da uma função devem ser implementadas a partir da magnitude do gradiente (GONZALEZ; WOODS, 2010), sendo que o gradiente de uma imagem descrita como $f(x,y)$ é definido como o vetor coluna, da seguinte forma:

$$\bigtriangledown f(x,y) = grad(f(x,y)) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} \tag{12}$$

Partindo da equação do gradiente, pode-se expressar a magnitude de $\bigtriangledown f(x,y)$ como $m(x,y)$, em que:

$$m(x,y) = \left| \bigtriangledown f(x,y) \right| = \sqrt{g_x^2 + g_y^2} \tag{13}$$

Desta forma, pode-se utilizar os termos das derivadas parciais $g_x$ e $g_y$ para formar o *operadores gradientes diagonais de Roberts* (GONZALEZ; WOODS, 2010), representado da seguinte forma:

$$\begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}, \begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix} \tag{14}$$

Porém, máscaras de tamanho pares são de difícil implementação, por não serem simétricas (GONZALEZ; WOODS, 2010), por essa razão as máscaras para derivadas parciais são implementadas em regiões de $3$x$3$, sendo o operador mais populares são os *operadores de Sobel* (GONZALEZ; WOODS, 2010), que representam as derivadas parciais na direção $x$ ou $y$, as máscaras respectivamente são:

$$\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}, \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \tag{15}$$

Na Figura 6, foi aplicado o filtro de Sobel para obter o resultado das derivadas parciais em $x$ e $y$, pode-se notar que pelos valores positivos e negativos da máscara, a resposta do mesmo é disposto entre -1 a 1, em que quanto maior o valor absoluto da resposta mais intensa é a borda em questão. Similarmente aos filtros de aguçamento baseados no Laplaciano, o realce em relação as primeiras derivadas pode ser representado como:

$$g(x,y) = f(x,y) + \left| \bigtriangledown f(x,y) \right| \tag{16}$$

Em que $g(x,y)$ será a resposta do filtro de aguçamento e $f(x,y)$ a imagem original.

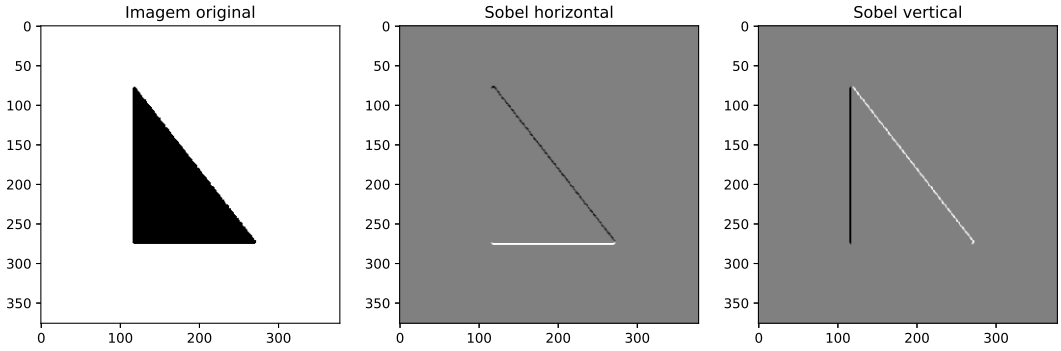


**Figura 6: Aplicação do filtro de Sobel para derivadas parciais em relação à $x$ ou $y$. Fonte: Autoria própria**

## 2.2 VISÃO ESTÉREO

Correspondência estéreo é o processo em que duas ou mais câmeras são utilizadas para a estimação de um modelo 3D, em que *pixels* da imagem em 2D são convertidos em profundidades 3D. Esses modelos geralmente utilizam modelagem esparsa para a recriação de modelos volumétricos ou superficiais (SZELISKI, 2011). A visão estéreo é uma das aplicações do *multi-view imaging* , em que duas câmeras observam uma cena com apenas um deslocamento horizontal entre elas, podendo, assim, modelar a profundidade do ambiente (SOLEM, 2012).

No início, a correspondência estéreo era utilizada para fotogrametria na reconstrução topológica em mapas de elevação (SZELISKI, 2011). Porém, diversas são as pesquisas em desenvolvimento, contendo, ainda, pesquisas em fotogrametrias (com a utilização de imagens aéreas), visão computacional incluindo modelagem da visão humana, navegação robótica, interpolação de imagem baseada em renderização, entre outras (SZELISKI, 2011).

Sendo duas imagens retificadas, pode-se assumir que para encontrar a correspondência de um *pixel* basta percorrer a linha da imagem (SOLEM, 2012). Se encontrado o pixel correspondente, a profundidade Z pode ser expressa como:

$$Z = \frac{fb}{x_l - x_r} \tag{17}$$

em que $f$ é a distância focal da câmera, $b$ é a distância entre o centro das câmeras e $x_l$

e $x_r$ são as coordenadas no eixo $X$ dos correspondentes *pixels* da direita e esquerda da imagem (SOLEM, 2012).

Reconstrução estéreo é um problema clássico na área de visão computacional (SOLEM, 2012), pois a profundidade deve ser estimada em cada *pixel* da imagem. Para solucionar esse problema existem diversos algoritmos e uma técnica que pode ser utilizada é a correspondência esparsa.

### 2.2.1 CORRESPONDÊNCIA ESPARSA

A correspondência esparsa é constituída por duas etapas, sendo a primeira responsável por extrair pontos característicos da imagem, utilizando operador de interesse ou detectores de borda (SZELISKI, 2011). A segunda etapa, também denominada correspondência estéreo ou *matching*, faz a extração de características de um bloco em torno do *pixel* de interesse, seguida por um processo de busca de correspondência nas demais imagens.

As limitações da correspondência esparsa são parcialmente devidas aos recursos computacionais, pois a necessidade de produzir respostas dos algoritmos de *matching* com grande precisão aumenta sua complexidade (SOLEM, 2012), já que correspondência esparsa é suscetível a problemas como mudança de iluminação, câmera *jitter*, entre outros.

Para diminuir o custo computacional, Zhang e Shan (2001) propõe como primeira etapa que poucos pontos sejam extraídos, desde que sejam pontos de alta confiabilidade e que possam ser utilizados para a geração de novos pontos para os demais correspondências.

Entretanto, se desconsideramos a necessidade de processamento em tempo real, o mais aceito é que sejam extraídos e utilizados tantos pontos quanto possível (inclusive, se todos pontos da imagem são utilizada, a correspondência é dita densa), normalmente aumentando a precisão do processo (SZELISKI, 2011).

### 2.2.2 DESCRITORES LOCAIS E *MATCHING*

Extração de características e *matching* são etapas fundamentais para diversas aplicações de visão computacional (SZELISKI, 2011). Quando é definida a aplicação utilizado correspondência esparsa, a primeira etapa é a extração de pontos de interesse nas imagens a serem comparadas. Estes pontos são obtidos a partir de características dos *pixels*, como bordas (que são caracterizadas pela sua intensidade e

orientação), cantos (que são descritos por um conjunto de *pixels* composto pelos seus vizinhos em uma região a sua volta) (SZELISKI, 2011), ou qualquer outra característica que seja desejada.

Inúmeros são os detectores de cantos propostos na literatura, como SIFT, Harris e outros (SOLEM, 2012). O método de Harris é bastante simples e de baixa custo computacional, baseado-se em encontrar pontos que possuam mais de uma borda intensa. Por estas características e, normalmente, obter resultados satisfatórios, será utilizado neste trabalho.

#### 2.2.2.1 DETECTOR DE CANTOS DE HARRIS

Para definir uma borda, deve-se primeiramente descrever os gradientes da imagem (HARRIS; STEPHENS, 1988), que são aproximadamente:

$$X = I * (-1, 0, 1) \approx \frac{\partial I}{\partial x} \tag{18}$$

$$Y = I * (-1, 0, 1)^T \approx \frac{\partial I}{\partial y} \tag{19}$$

Após calcular os gradientes, pode-se definir uma matriz $M_I$ (SOLEM, 2012), em que $\lambda_1 = |\nabla I|^2$ e $\lambda_2 = 0$, representada da seguinte forma:

$$M_I = \nabla I \nabla I^T = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{20}$$

A partir da matriz $M_I$, pode-se obter matriz de Harris, definida como a seguinte operação:

$$\overline{M_I} = W * M_I \tag{21}$$

em que $\overline{M_I}$ é a matriz de Harris e $w$ é uma máscara de peso, que depende da região de interesse (HARRIS; STEPHENS, 1988). Após construída, pode-se analisar os valores da matriz, que sao dispostos nos 3 seguintes grupos:

- Se $\lambda_1$ e $\lambda_2$ são ambos valores elevados e positivos, então o ponto de interesse é um canto.
- Se $\lambda_1$ é um valor elevado e $\lambda_2 \approx 0$, então o ponto de interesse é uma borda.
- $\lambda_1$ e $\lambda_2$ são valores próximos a 0, então o ponto de interesse se encontra em uma região uniforme.

Como em muitas vezes, apenas a detecção da borda não é suficiente, deve-se

definir uma maneira de quantificar a qualidade de uma borda ou canto. Assim, Harris e Stephens (1988), introduziram o indicador

$$R = Det(\overline{M_I}) - k \ \ trace(\overline{M_I}) \tag{22}$$

assim um *pixel* de canto pode ser selecionado a partir dos valores em que $R$ são máximos e *pixels* de borda são os valores em que $R$ são mínimos (HARRIS; STEPHENS, 1988). Desta forma, pode-se aplicar *thresold* altos ou baixos, para detecção de pontos de interesse na imagem.

Na Figura 7 pode-se observar a aplicação do detector de Harris. A imagem da direita é a resposta do detector de cantos de Harris, em que *pixels* cinzas podem ser considerados regiões homogêneas, os pontos mais escuros as bordas e os *pixels* brancos são os cantos.

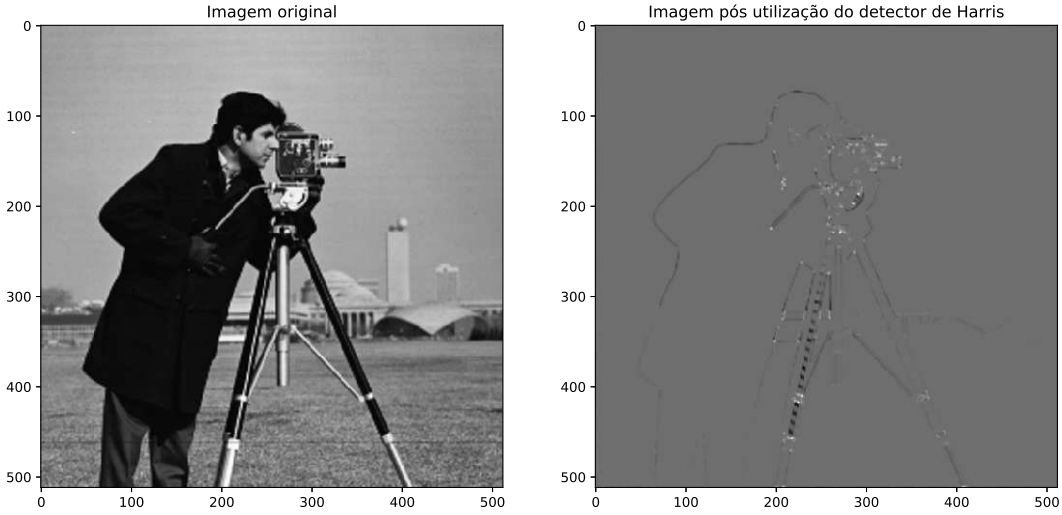


**Figura 7: Aplicação do detector de cantos de Harris com $\sigma$ = 1. Fonte: Autoria própria**

### 2.2.2.2 CORRESPONDÊNCIA ENTRE PONTOS

Após a extração de pontos característicos, é necessário fazer a comparação entre os pontos selecionados para encontrar o correspondente do mesmo em outra imagem (SOLEM, 2012).

Um ponto de interesse, geralmente, é expresso como um vetor de características, estas que devem representar a vizinhança em torno do mesmo (SOLEM, 2012). Sendo que quanto melhor for a representação dessa vizinhança, mais preciso será a correspondência entre dois pontos.

Usualmente a representação das características de pontos provenientes de extratores de cantos, como o método de Harris, é feito a partir de uma combinação dos valores de *pixels* vizinhos do ponto de interesse. Estes conjuntos de valores serão comparados com a utilização de métodos de cálculo de diferenças, como: correlação cruzada normalizada ou soma do quadrado das diferenças (SOLEM, 2012).

Geralmente, a correspondência entre dois conjuntos de pontos, definido como o pixel de interesse e seus vizinhos, $I_1(X)$ e $I_2(x)$ é definido como

$$c(I_1, I_2) = \sum_x f(I_1(x), I_2(x)) \tag{23}$$

em que a função $f$ depende do método de correlação a ser utilizado (SOLEM, 2012). No caso da correção normalizada, esta é definida como a seguinte operação

$$ncc(I_1, I_2) = \frac{1}{N-1} \sum_x \frac{(I_1(x) - \mu_1)}{\sigma_1} . \frac{(I_2(x) - \mu_2)}{\sigma_2} \tag{24}$$

em que $N$ é o numero de pontos do conjunto, $\mu_1$ e $\mu_2$ são as respectivas médias dos mesmos, e $\sigma_1$ e $\sigma_2$ são seus desvios padrão. Por subtrair a media e dividir pelo desvio padrão, o método se torna robusto a mudanças de luminosidade (SOLEM, 2012).

## 2.3 EXTRAÇÃO DE FUNDO

Extração de fundo pode ser utilizada em diversas aplicações, exemplos simples são: a detecção de movimento em vídeos de vigilância, captura por movimento e multimídia. O modelo mais simples para estimar o fundo é afirmar que tudo que se move entre os *frames* é caracterizado como se não fosse fundo (BOUWMANS *et al.*, 2008). Porém, esta técnica está sujeita à diversos problemas, como: mudança de iluminação, introdução ou remoção de objetos da cena, entre outros.

Existem diversas classificações de modelagem de fundo, podendo ser: modelagem básica de fundo, modelagem estatística de fundo, modelagem *fuzzy* de fundo, entre outras (BOUWMANS *et al.*, 2008). O modelo mais utilizado é a mistura de gaussianas (MOG), sendo um modelo baseado em estatística, obtendo resultados que não comprometem significativamente sua robustez em situações críticas ou de restrições.

### 2.3.1 MISTURA DE GAUSSIANAS

Inicialmente, deve-se categorizar cada pixel pela sua intensidade no espaço de cores RGB, determinando a probabilidade deste ser fundo (BOUWMANS *et al.*, 2008),

a partir da formula multidimensional:

$$P(x_t) = \sum_{i=0}^{k} w_{i,t}\eta(x_t, \mu_{i,t}, \sigma_{i,t}) \tag{25}$$

em que o parâmetro $k$ é o número de distribuições, $w_{i,t}$é o peso associado à i-ésima gaussiana em um tempo $t$, $\mu_{i,t}$ é a média, com o desvio padrão $\sigma_{i,t}$. $\eta$ é a função de densidade gaussiana:

$$\eta(x_t, \mu, \sigma) = \frac{1}{(2\pi)^{\frac{1}{2}} |\sigma|^{\frac{1}{2}}} e^{\frac{-1}{2}(x_t-\mu)\sigma^{-1}(x_t-\mu)} \tag{26}$$

Por motivos computacionais, pode-se assumir que as componentes RGB são independentes e possuem a mesma variância. Por esta razão, a matriz de covariância é definida como:

$$\sum_{i,t} = \sigma_{i,t}^2 I \tag{27}$$

Assim, pode-se inicializar diferentes misturas de gaussianas, variando seus parâmetros (como: o número de gaussianas, o peso $w_(i,t)$, a média $\mu_{i,t}$ e a matriz de covariância $\sum_{i,t}$), categorizando, desta forma, cada *pixel* por uma mistura de $k$ gaussianas (BOUWMANS *et al.*, 2008). Dentre as $k$ gaussianas, as primeiras $T$ (limiar previamente definido) são consideradas *background*, enquanto as demais são *foreground*.

Desta forma, após definidos os parâmetros das misturas de gaussianas, em um *frame* com o tempo $t + 1$, é realizado um teste de correspondência em cada *pixel* (BOUWMANS *et al.*, 2008). Para cada um deles, procura-se uma correspondência na distribuição gaussiana. Então, pode ocorrer duas situações:

(i) É encontrado pelo menos uma correspondência nas $k$ gaussianas. Neste caso, se a gaussiana é classificada como *background*, o *pixel* será classificado como *background*, caso contrário, ele é classificado como *foreground*.

(ii) Não é encontrada nenhuma correspondência nas $k$ gaussianas. Neste caso, o pixel é classificado como *foreground*

Os parâmetros das gaussianas são atualizados a cada tempo $t$, permitindo que a modelagem do *background* se adapte a mudanças sutis das imagens.

#### 2.3.1.1 CARACTERÍSTICAS DE PROFUNDIDADE

Apesar da simples utilização dos valores RGB de cada pixel ser a mais comum entre aplicações com MOG, é possível a combinação destas intensidades com valor da disparidade (fator inversamente proporcional a profundidade, porém sem compensação) ou mesmo a profundidade. A utilização dessa característica extra tem a vantagem de diminuir problemas encontrados na remoção de fundo, como camuflagem (BOUWMANS *et al.*, 2008). Porém, outras limitações são encontradas, como a necessidade de que o ambiente a ser processado possua uma textura miníma para confiabilidade de correspondências.

O método proposto por Gordon *et al.* (1999), por exemplo, utiliza do mesmo princípio da mistura de gaussianas clássica, apenas considerando um aprimoramento na técnica com a utilização de uma característica extra, a disparidade, como parâmetro de ponderamento. De forma similar, Harville *et al.* (2001) propuseram a utilização de profundidade como característica adicional, mantendo o princípio de funcionamento do MOG.

## 2.4 SUPERPIXELS

Superpixel é o conceito de agrupamento de *pixel*, baseando-se em algum parâmetro, com a finalidade de criação de estruturas de *pixels* (ACHANTA *et al.*, 2012). As propriedades esperadas por esse métodos são:

- Deve aderir bem as bordas da imagem;
- Reduz a complexidade computacional das aplicações, sendo um método de rápida execução, com baixo custo de memória e de fácil uso;
- Quando utilizado para segmentação, tende a aumentar a velocidade de execução e a qualidade dos resultados

### 2.4.1 SLIC

SLIC (*Simple linear iterative clustering*) é um método de criação de *superpixels* (ACHANTA *et al.*, 2012), sendo uma adaptação do algoritmo *k-means* possuindo apenas duas distinções:

- O número de cálculos de distância na otimização é drasticamente reduzido, limitando o espaço de busca à região proporcional. Isso reduz a complexidade

para ser linear ao número de pixels N da imagem, sem levar em consideração a quantidade de *clusters*.

- A medida de distância é calculado ponderando os valores do espaço de cor e a proximidade dos pixels, proporcionando um controle sobre o tamanho dos *superpixels*.

O algoritmo SLIC necessita de $k$ centros, inicialmente uniformemente distribuídos entre as coordenadas $x$ e $y$ da imagem. Utilizando o espaço de cores CIELAB, cada centro é definido como $C_i = [l_i, a_i, b_i, x_i, y_i]^T$ e todos *pixels* da imagem devem ser atribuídos a um centro, sendo estipulado que o mesmo pertencerá ao centro mais próximo. Uma vez atribuídos os *pixels* aos centros, os mesmo são recalculados e recebem novos conjuntos de *pixels*. Este processo se repete até um critério de parada (ACHANTA *et al.*, 2012). O algoritmo completo é o seguinte:

**Algoritmo 2.1:** SLIC

**Entrada:** $MAT, WID, TH$
**Saída:** $PontosSelecionados$

**início**
  Inicializa-se os k centros $C_k = [l_i, a_i, b_i, x_i, y_i]^T$
  Move o centro do agrupamento para o menor gradiente da vizinhança $3x3$
  Cria-se um vetor $e(i)$ = -1 com $i$ sendo o número de pixel
  Cria-se um vetor $d(i)$ = $\infty$ com $i$ sendo o número de pixel
  **repita**
    **para** *cada centro* $C_k$ **faça**
      **para** *cada pixel* $i$ *em volta da região* $C_k$ **faça**
        Compute a distância $D$ entre $C_k$ e $i$
        **se** $D < d(i)$ **então**
          $d(i) = D$
          $e(i) = k$
      Computar novos centros
      Computar erro residual $E$
  **até** *Limiar* $>= E$;

A Figura 8 é a aplicação do SLIC, deve-se notar a divisão da subárea gerada pela utilização do algoritmo.

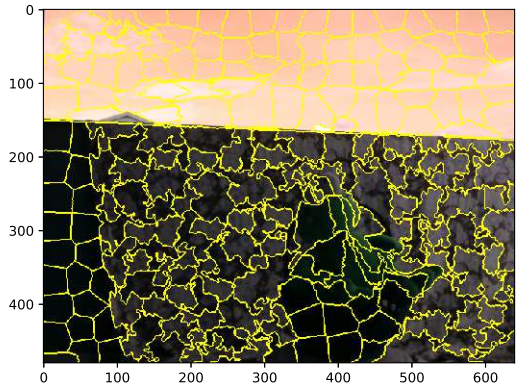


**Figura 8: Aplicação do SLIC. Fonte: Autoria própria**

# 3 METODOLOGIA

Neste capítulo, será apresentado a sequência de métodos do sistema proposto, sendo apresentado detalhadamente cada uma destas etapas . O diagrama de blocos da Figura 9 apresenta as etapas do método utilizado no desenvolvimento deste trabalho. A entrada é uma dupla de imagens e a saída serão duas imagens binárias com os respectivos *foreground* e *background*.

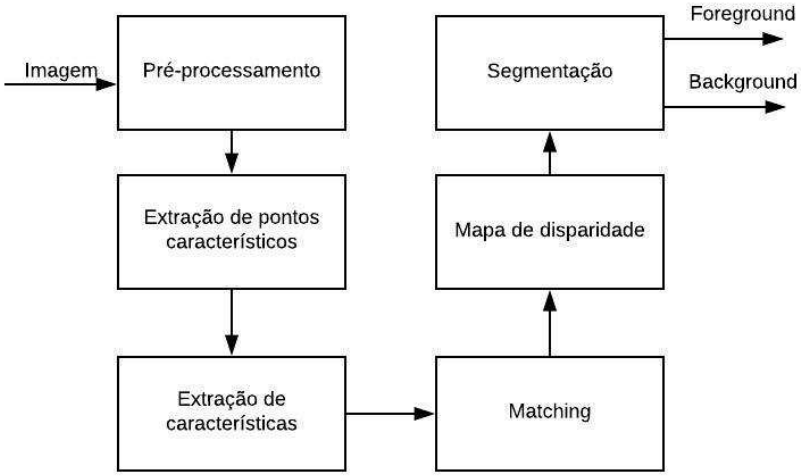


**Figura 9: Diagrama de blocos do processo a ser implementado**

As imagens de entrada foram adquiridas por duas câmeras, modelo *Logitech* c270, com resolução de 640x480 $pixels^2$, dispostas de forma paralela em cima de uma base móvel em movimento sistemático, sendo que quando há movimento na câmera tanto o *foreground* quanto *background* sofrem alterações, utilizando o microcontrolador MSP430G2553. A Figura 10 apresenta o protótipo desenvolvido para esta aplicação.

A figura 11 é a aquisição de quatro *frames* subsequentes e demonstra o efeito de câmeras em movimento. Pode-se notar que todos os objetos, *foreground* e *background*, são alterados com o movimento das câmeras, sendo que algumas partes possuem mudanças de posições, enquanto outras são removidas do quadro em questão.


**Figura 10: Protótipo para aquisição de vídeo**

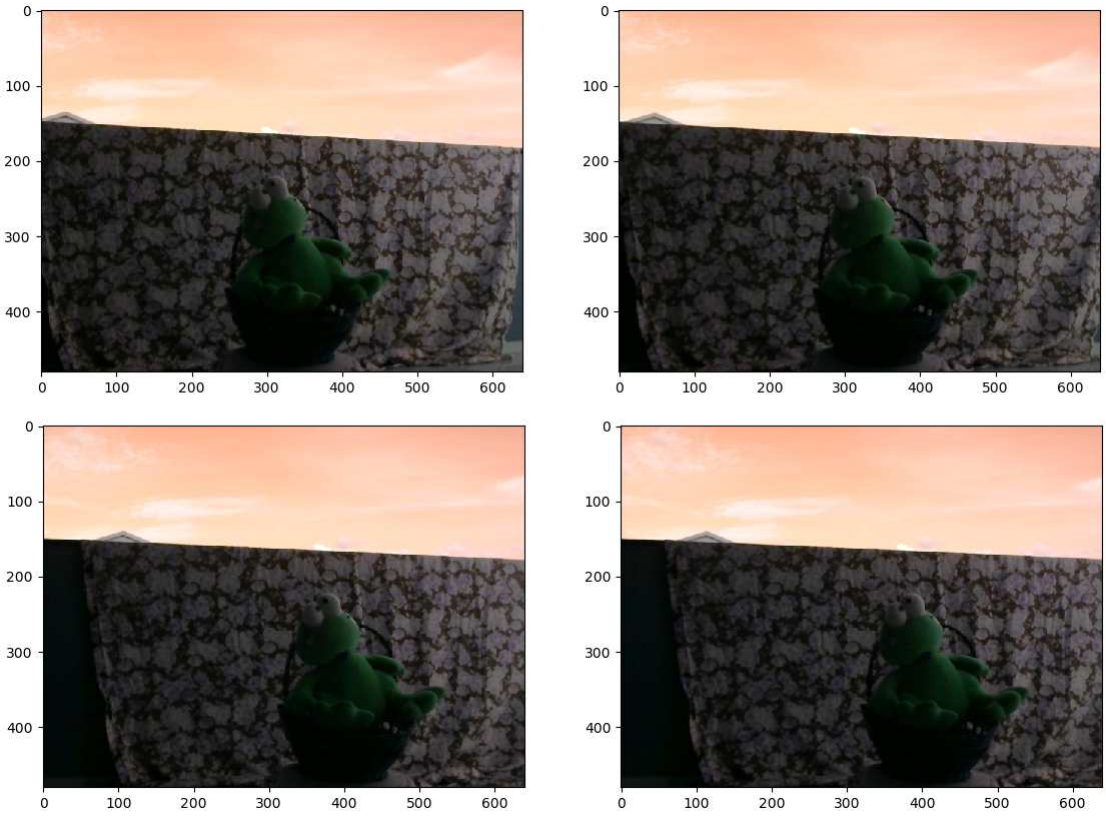

**Figura 11: *Frames* subsequentes com câmeras em movimento**

## 3.1 EXTRAÇÃO DE PONTOS DE INTERESSE

A primeira etapa para a criação de um mapa de disparidade baseado em correspondência esparsa é a utilização de um descritor que quantifica quais *pixels* possuem características que os diferenciem um dos outros. Para isto, foi escolhido o descritor de cantos de Harris. Entretanto, ao invés da Eq.( 22), foi utilizada a equação

a seguir.

$$f(x,y) = \frac{det(M(x,y))}{trace(M(x,y))} \tag{28}$$

Da mesma forma que a equação originalmente proposta, esta equação é capaz de diferenciar para cada pixel $(x, y)$ quais os que representam cantos, possuindo valores mais elevados da função e que, portanto, serão nossos *pixels* de interesse para realizar a correspondência esparsa.

A Figura 12 apresenta a matriz resposta do descritor de cantos de Harris utilizado, onde pode ser observado que poucos pontos apresentam valores elevados e, portanto, a grande maioria dos *pixels* serão descartados nesta etapa.

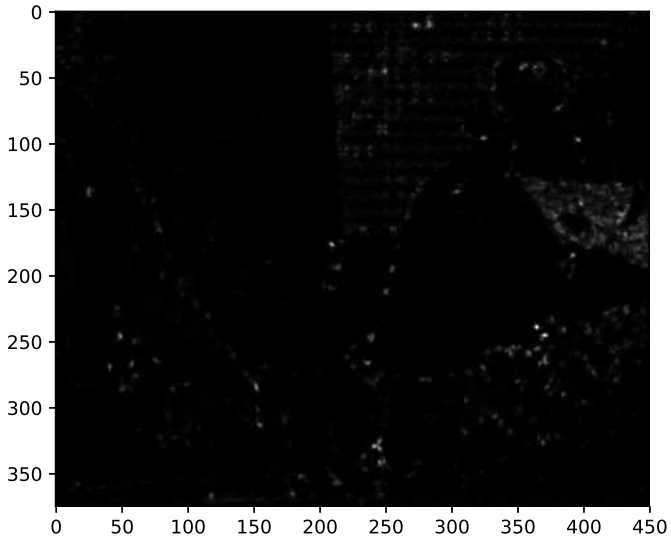


**Figura 12: Resposta ao descritor de cantos de Harris utilizado. Fonte: Autoria própria**

Após a utilização do descritor, foi necessário a seleção dos *pixels* que possuíssem os melhores descritores. Tendo em mente a necessidade de não selecionar diversos *pixels* próximos uns aos outros, pois isso poderia causar conflito no momento da correspondência já que seus valores seriam deveras parecido. Desta forma, para a seleção dos pontos, foram utilizados dois parâmetros: um limiar $TH$, para desprezar os pontos com valores baixos no descritor de Harris; e uma janela de distância mínima entre os *pixels*, $WID$, para que cada ponto selecionado estivesse a pelo menos $WID$ *pixels* de distância do próximo.

Abaixo será apresentado o pseudo-código que representa a função de seleção de pontos de interesse, baseando-se em um limiar ($TH$), uma janela mínima ($WID$) e

a matriz de descritores de Harris ($MAT$).

**Algoritmo 3.1:** ExtratorDePontosHarris

**Entrada:** $MAT, WID, TH$

**Saída:** $PontosSelecionados$

1 **início**

2 $Coords = ArgSort(MAT > TH)$

3 $PontosSelecionados = []$

4 **para** *cada* $Coord \in Coords$ **faça**

5 Adiciona $Coord$ a $PontosSelecionados$

6 $RemoveVizinhos(Coords, WID)$

7 **retorna** $PontosSelecionados$

Neste algoritmo, tem-se a entrada dos parâmetros $MAT$, representação da matriz de descritores de Harris para todos *pixels* da imagem, $WID$, janela mínima entre *pixels*, e $TH$, o limiar mínimo que deve ser considerado na Matriz. A partir disto, é selecionado em ordem decrescente do maior valor da matriz ao menor, limitado pelo $TH$, e a cada ponto selecionado todos seus vizinhos em um raio de $WID$ são removidos da possível inserção. A Figura 13, é o resultado da seleção de pontos para uma imagem exemplo.

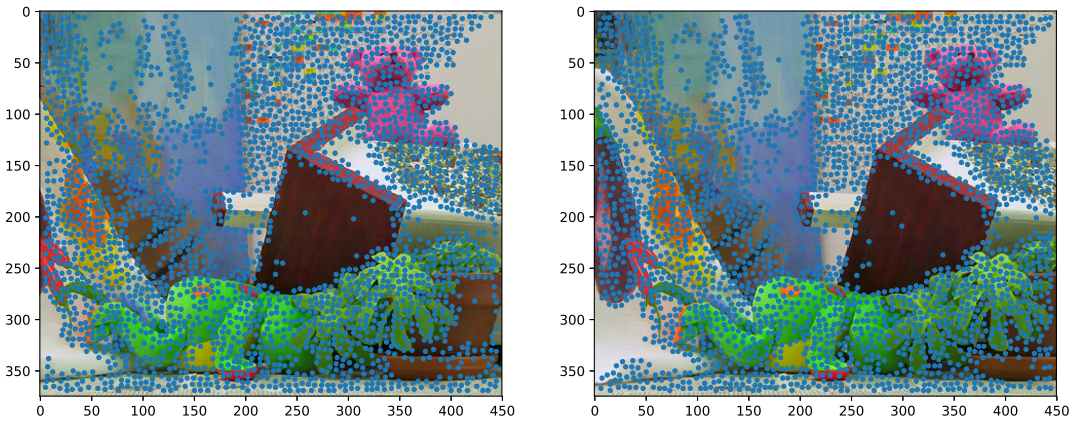


**Figura 13: Pontos selecionados a partir da matriz de descritores de Harris em um par de imagens estéreo. Fonte: Autoria própria**

Na Figura 12, podemos observar como os pontos são selecionados: cada ponto na imagem representa um pixel selecionado naquele local, sendo que é possível observar que áreas homogêneas não possuem *pixels* selecionados, por ocorrência do

limiar utilizado.

## 3.2 CORRESPONDÊNCIA ENTRE PIXELS

Após a seleção de pontos, há uma necessidade de achar o seu correspondente na imagem oposta. Por esta razão, é necessário expressar as características de cada pixel, representando-o como um vetor de características e utilizando um método para calcular se há, ou não, um correspondente. Para a representação de cada pixel selecionado, foi criado um vetor característica composto pelos valores RGB, obtidos a partir de uma janela de $10x10$ centrada no pixel.

Com os valores de representação extraídos, foi necessária uma métrica para selecionar os correspondentes entre as imagens. Definiu-se a utilização da correlação cruzada normalizada, apresentada no Capítulo 2, em cima dos vetores de características, juntamente a algumas restrições para melhorar o desempenho e resposta da seleção de pontos (SOLEM, 2012) (HIRSCHMÜLLER *et al.*, 2002).

A primeira restrição se refere ao raio de busca de um pixel correspondente na imagem oposta. Partindo da ideia de que as imagens são paralelas, os *pixels* deveriam:

- Possuir valores $x_0$ e $x_1$ parecidos, pois não devem possuir deslocamento no eixo $X$.
- Possuir $y_1 >= y_0$, pois há um deslocamento na imagem da direita.

Outra otimização aplicada, é desprezar *pixels* que os seus dois maiores valores de correspondência sejam parecidos, pois caso dois ou mais valores de correspondência sejam parecidos, estamos lidando com áreas homogêneas que devem ser desprezadas pela grande ocorrência de erros no calculo de disparidade.

A ultima restrição aplicada foi a correspondência dupla, em que são eliminadas correspondências que não formem duplas simétricas. Por exemplo, se um pixel A da imagem esquerda teve sua mais alta correspondência calculada para um pixel B da imagem direita, assumimos que este pixel B também deve obter a mais alta correspondência no pixel A. Portanto, a correspondência não se torna válida se é definida entre os *pixels* apenas em um sentido.

Partindo de todos os pontos selecionados na Figura 12, podemos ver o resultado obtido pelo método de correspondência na Figura 14. Observa-se a diminuição

significativa dos pontos, já que muitos foram eliminados por não possuírem correspondência na imagem oposta.

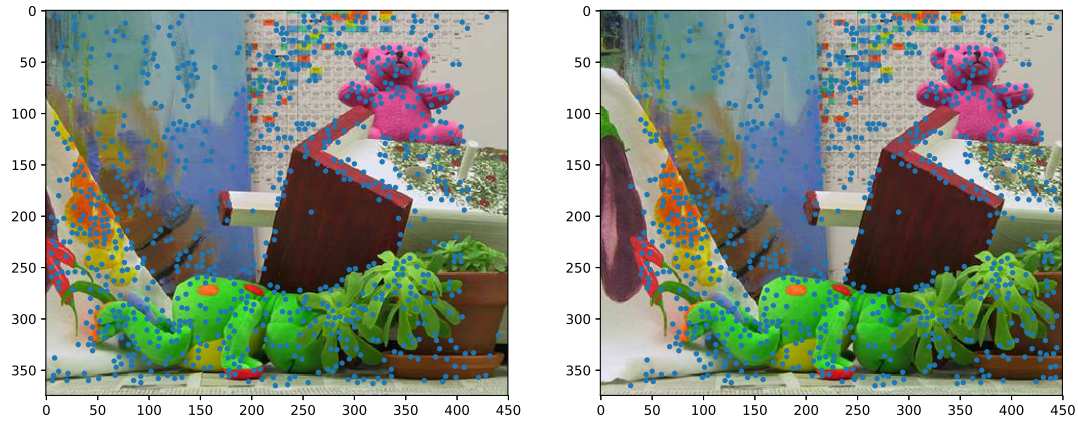


**Figura 14: Par de imagens com os *pixels* que possuem correspondência. Fonte: Autoria própria**

## 3.3 SEGMENTAÇÃO

Com os *pixels* correspondentes selecionados, a próxima etapa é o cálculo de disparidade e consequente segmentação da imagem baseada em distância. Uma vez que um pixel $A$ já foi computado como correspondente ao pixel $B$, a disparidade entre eles é dada pela diferença entre suas coordenadas $x$. Dois *pixels* serão menos díspares quando mais distantes das câmeras na cena, enquanto *pixels* mais díspares (e, portanto, mais próximos das câmeras) terão esta diferença maior.

Porém, há uma necessidade anterior que é espalhar os valores de disparidade para os *pixels* vizinhos dos pontos. A Figura 15 representa a escassez de pontos a terem sua disparidade computada e a consequente necessidade da utilização de um método que faça essa distribuição.

Considerando que cada ponto na Figura 15 representa um pixel, pode-se notar que para uma imagem com milhares de *pixels*, menos que 1000 destes possuem valores já definidos.

Desta forma, será utilizado o conceito de super-pixel para a criação de regiões similares. O algoritmo para a criação destas áreas foi o SLIC (ver capítulo 2), sendo que a resposta do mesmo é mostrado na Figura 16, subdividido por áreas em

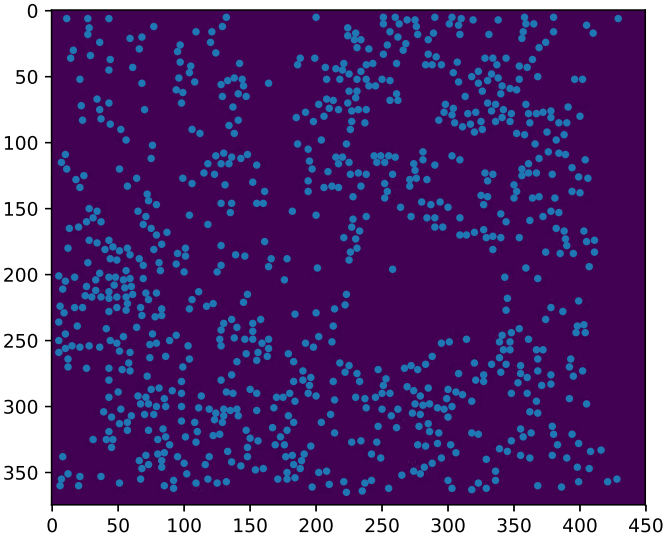


**Figura 15: Localização de pontos de disparidade. Fonte: Autoria própria**

tonalidades diferentes.

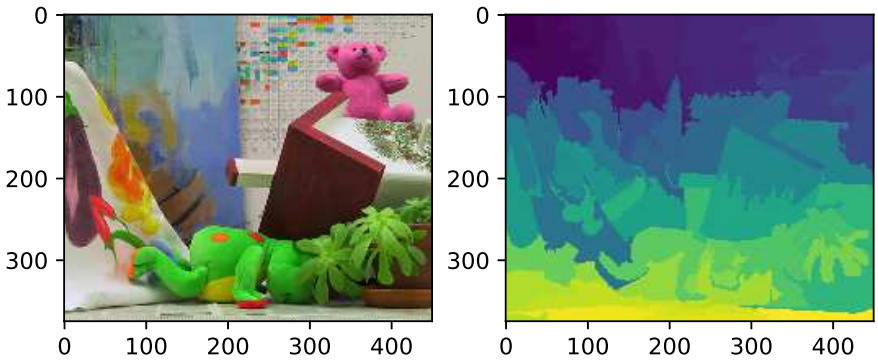


**Figura 16: Subdivisão gerada pela utilização do SLIC. Fonte: Autoria própria**

Com a subdivisão em regiões realizada, passamos a atribuição de valores para cada uma delas. Cada região receberá o valor da mediana da disparidade dentre os seus *pixels*, visando a diminuição de danos por valores atípicos ou ruídos. A Figura 17 apresenta o resultado da utilização do SLIC juntamente com a atribuição dos

valores de área.

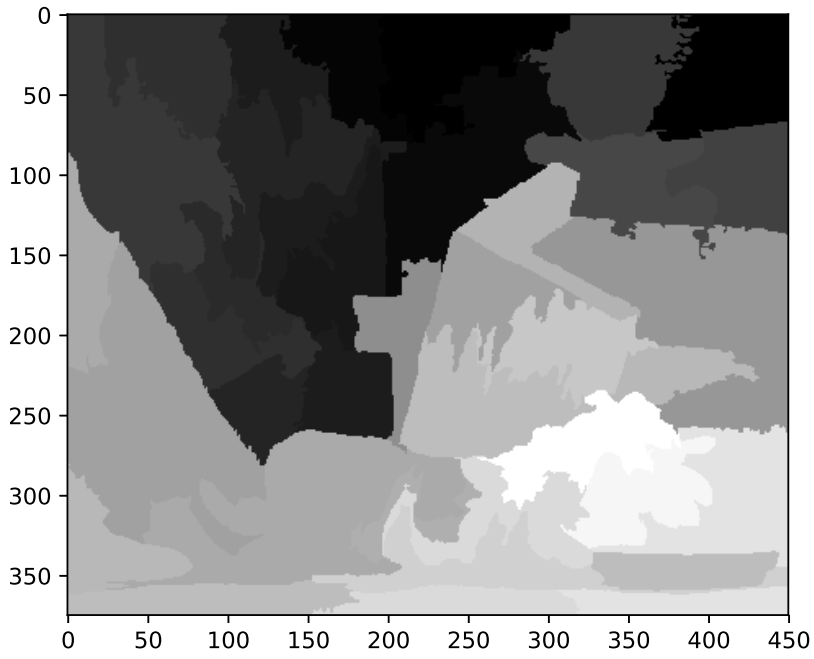


**Figura 17: Representação do mapa de disparidade com SLIC. Fonte: Autoria própria**

Com os valores atribuídos a todas regiões, é possível que os *pixels* assumam dois valores:

- Valor $X >= 0$, caso exista pelo menos um valor de disparidade na área,
- Atribui-se valor $X = -1$, caso não exista nenhuma valor de disparidade na área.

Assim, é possível diferenciar as áreas que possuem problemas no cálculo de disparidade e desprezar as mesmas na geração do histograma para segmentar a imagem resultante.

A seguir, é possível gerar um histograma da imagem com os valores de disparidade por área. A Figura 18 apresenta o histograma gerado para a imagem exemplo, sendo possível a visualização de um valor que tem potencial de segmentar a imagem, ou seja, separar o *background* do *foreground*.

A partir do histograma, é possível fazer a segmentação da imagem. A Figura 19 apresenta o resultado final, com um limiar fixo $25$ capaz de separar os dois modos do histograma. A região em branco representa o *foreground*, enquanto preto representa o *background* da imagem. Ainda importante considerar que, se desejável, o limiar pode ser obtido com a utilização de uma função específica para este fim

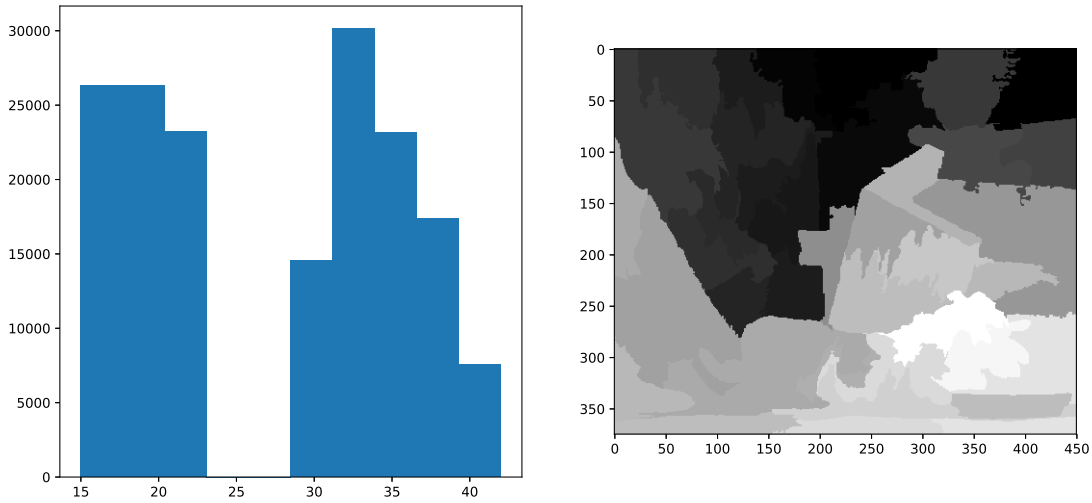


**Figura 18: Histograma obtido a partir do mapa de disparidade gerado pelo SLIC. Fonte: Autoria própria**

(cálculo do limiar de Otsu) (OTSU, 1979).

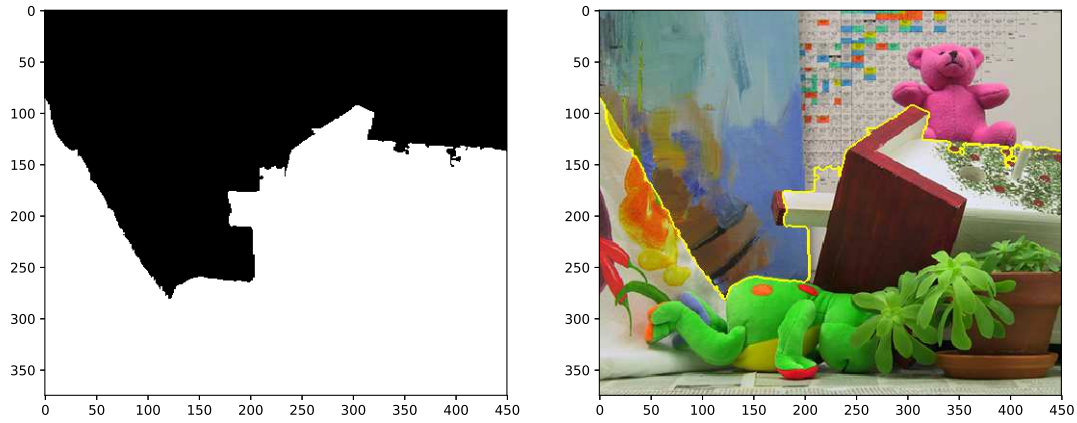


**Figura 19: Resultado da segmentação do mapa de disparidade. Fonte: Autoria própria**

# 4 RESULTADOS E DISCUSSÕES

Todo trabalho realizado foi implementado utilizando-se a IDE *Scientific Python Development Environment* (SPYDER). Os experimentos foram executados em um equipamento Asus X555UB com processador Intel(R) Core(TM) i5-6200U em 2,30Ghz e 8,00 GB de memória (RAM).

Durante os experimentos, foram identificados diversos problemas que impossibilitaram melhores resultados. Porém, com as restrições implementadas na etapa de correspondência pode-se notar melhoria significativa nos resultados, tanto no tempo de execução quanto na melhoria do resultado.

Para fins de comparação, a Figura 20 apresenta um exemplo de resultado sem as restrições implementadas. Podemos notar uma significante piora (em relação ao resultado apresentado na Figura 19) nas áreas de segmentação, em que cinza significa *background*, branco representa *foreground* e o preto são todas as áreas indeterminadas.

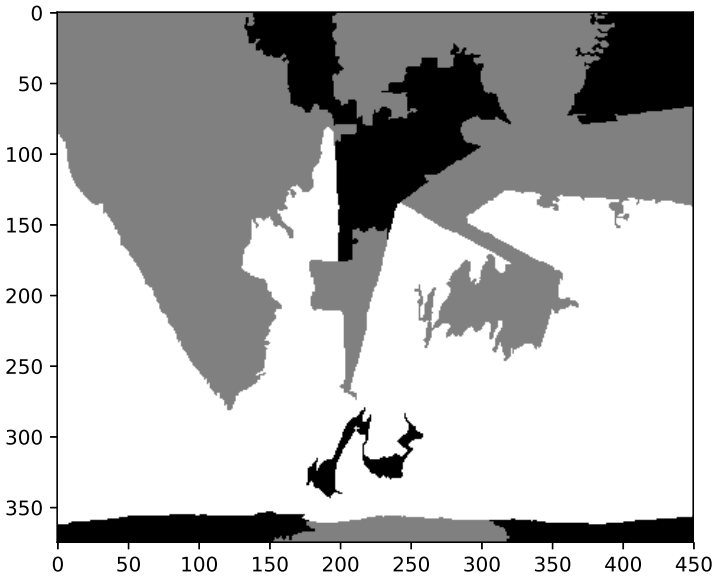


**Figura 20: Resultado da segmentação do mapa de disparidade sem restrições. Fonte: Autoria própria**

Além da significante melhoria no resultado, o tempo de execução do sistema

foi altamente reduzido com a utilização das restrições. Com a restrição de área como dois *pixels* de erro, o tempo de execução foi de 45 segundos, enquanto o tempo de execução sem restrição ultrapassou os 65 minutos (3900 segundos). Esses resultados refletem a necessidade da utilização de restrições das buscas na etapa de correspondência e a necessidade de aplicar outras restrições específicas para cada aplicação.

Conforme apresentado até então, os resultados foram considerados bastante satisfatórios para imagens em condições ideais. Entretanto, quando da aquisição de imagens em experimentos idealizados para este projeto, os resultados obtidos se mostraram inconsistentes, pois os mesmos possuíam uma grande variação na qualidade da resposta ao sistema. Os principais problemas encontrados foram:

- Dificuldade de correspondência nas áreas homogêneas, criando diversas áreas indefinidas;
- Utilização do espaço de cor RGB se mostrou pouco confiável para representação dos *pixels*;
- Dificuldade de restringir a área de busca nos *frames* do vídeo pela tremulação ao movimentar as câmeras;
- A utilização de *superpixel* em imagens reais se mostrou variante, gerando subáreas diferentes para *frames* similares;
- Problemas gerados pela mudança de ambiente, devido à grande quantidade de limiares fixos;

A Figura 21 é um ótimo exemplo para apontar algum dos erros citados acima. As duas imagens à esquerda foram obtidas a partir do protótipo em *frames* subsequentes, e as imagens da direita são os resultados obtidos após o processamento do sistema proposto, em que branco representa *foreground*, cinza significa *background* e as áreas em preto são regiões indefinidas.

Inicialmente, pode-se notar que a área superior da imagem (por se tratar de uma região homogênea) gerou dois problemas: criação de regiões indefinidas e erro de segmentação causado pelo erro na correspondência entre os *pixels*. Ainda que grande parte do fundo tenha sido segmentado corretamente, devido a textura do mesmo, pode-se também notar uma falha de segmentação gerada pela utilização do SLIC: apesar dos *frames* serem subsequentes, o SLIC apresentou resultados inconsistentes entre si e, por consequência, os mapas de disparidade são bastante diferentes.

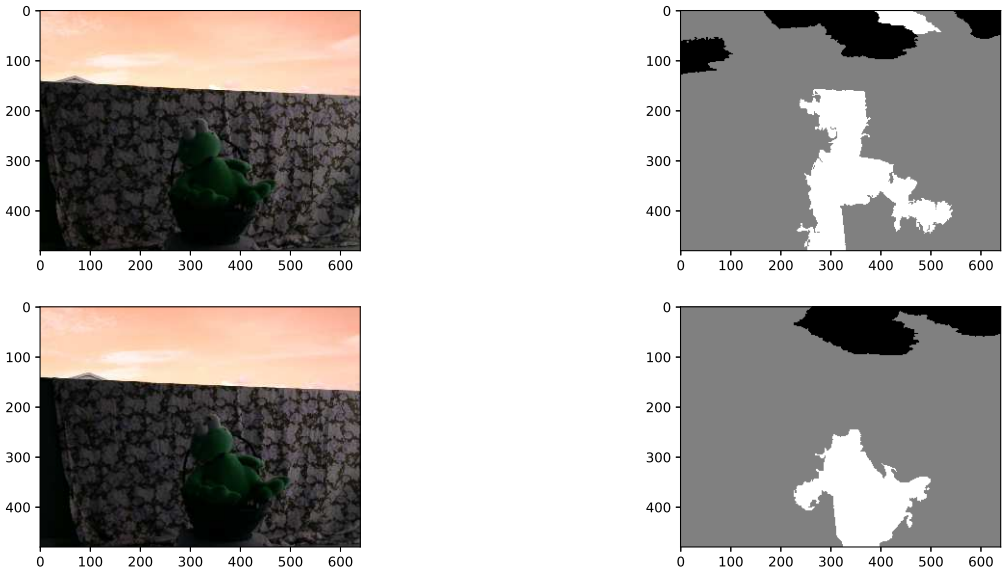


**Figura 21: Resultado da segmentação do mapa de disparidade para dois *frames* distintos. Fonte: Autoria própria**

Para melhorar o resultado nesses vídeos, seria necessário achar uma forma de melhor representação de *pixels* que pudesse resultar em melhor correspondência dos mesmos, sendo possível extrair informações extras e criar restrições mais robustas. Além disto, os problemas gerados pelas áreas indefinidas necessitariam de uma solução de contorno.

# 5 CONCLUSÃO

Neste trabalho, foi desenvolvida uma aplicação de visão computacional em que foi aplicada visão estéreo para a remoção do fundo (*background*) de vídeos, em que foi necessário a aplicação de conceitos de filtragem espacial, extração de pontos de interesse, extração de características dos *pixels*, correspondência esparsa, *super-pixel*, entre outros.

Foram implementadas técnicas para melhorar o desempenho em etapas essenciais, como a restrição de áreas de busca referente a correspondência esparsa e a utilização do algoritmo SLIC (*Simple Linear Iterative Clustering*) para a criação de subáreas para a dispersão do mapa de disparidade, estas sendo classificadas em 3 grupos: *background*, *foreground* e áreas indefinidas.

Os resultados preliminares se mostraram eficientes. Porém, em casos obtidos pelo protótipo proposto, os resultados não se mostraram totalmente confiáveis, possuindo erros de segmentação.

Em trabalho futuros, pode-se explorar novas técnicas de correspondência esparsa, desde a extração de características até a utilização de novas funções para o cálculo da mesma, a criação de situações de contorno para regiões indefinidas (áreas homogêneas da imagem) e a criação de um protótipo menos suscetível a tremulação nas câmeras, uma vez que este fator atrapalha na criação de regras de restrição da busca de candidatos para correspondência.

# REFERÊNCIAS

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SÃ$\frac{1}{4}$SSTRUNK, S. Slic superpixels compared to state-of-the-art superpixel methods. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 11, p. 2274–2282, Nov 2012. ISSN 0162-8828.

BOULMERKA, A.; ALLILI, M. S. Background modeling in videos revisited using finite mixtures of generalized gaussians and spatial information. In: **2015 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2015. p. 3660–3664.

BOUWMANS, Thierry; BAF, Fida El; VACHON, Bertrand. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. **Recent Patents on Computer Science**, Bentham Science Publishers, v. 1, n. 3, p. 219–237, nov. 2008.

BRAHAM, M.; PIÃ©RARD, S.; DROOGENBROECK, M. Van. Semantic background subtraction. In: **2017 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2017. p. 4552–4556.

GONZALEZ, R.C.; WOODS, R.E. **Processamento Digital De Imagens**. [S.l.]: ADDISON WESLEY BRA, 2010. ISBN 9788576054016.

GORDON, G.; DARRELL, T.; HARVILLE, M.; WOODFILL, J. Background estimation and removal based on range and color. In: **Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)**. [S.l.: s.n.], 1999. v. 2, p. 464 Vol. 2. ISSN 1063-6919.

HARRIS, Chris; STEPHENS, Mike. A combined corner and edge detector. In: **In Proc. of Fourth Alvey Vision Conference**. [S.l.: s.n.], 1988. p. 147–151.

HARVILLE, Michael; GORDON, Gaile G.; WOODFILL, John. Foreground segmentation using adaptive mixture models in color and depth. In: **IEEE Workshop on Detection and Recognition of Events in Video**. [S.l.: s.n.], 2001.

HIRSCHMÜLLER, Heiko; INNOCENT, Peter R; GARIBALDI, Jon. Real-time correlation-based stereo vision with reduced border errors. **International Journal of Computer Vision**, Springer, v. 47, n. 1-3, p. 229–246, 2002.

OTSU, Nobuyuki. A threshold selection method from gray-level histograms. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 9, n. 1, p. 62–66, 1979.

PETROU, Maria; BOSDOGIANNI, Panagiota. **Image Processing : The Fundamentals**. [S.l.]: Wiley, 1999.

SOLEM, Jan Erick. **Programming computer vision with Python**. Beijing; Cambridge; Sebastopol [etc.]: O'Reilly, 2012. ISBN 9781449316549 1449316549.

SZELISKI, Richard. **Computer vision algorithms and applications**. London; New York: Springer, 2011.

VIEIRA, G. d. S.; SOARES, F. A. A. M. N.; LAUREANO, G. T.; SOUSA, N. M. de; OLIVEIRA, J. G. A.; PARREIRA, R. T.; FERREIRA, J. C.; COSTA, R. M. da. Stereo vision methods: From development to the evaluation of disparity maps. In: **2017 Workshop of Computer Vision (WVC)**. [S.l.: s.n.], 2017. p. 132–137.

ZHANG, Zhengyou; SHAN, Ying. A progressive scheme for stereo matching. In: **Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments**. London, UK, UK: Springer-Verlag, 2001. (SMILE '00), p. 68–85. ISBN 3-540-41845-8.