



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Matheus André Soares

**Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de  
Supermercado**

Araranguá  
2021

Matheus André Soares

**Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de Supermercado**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Antônio Carlos Sobieranski, Dr.

Araranguá

2021

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Soares, Matheus André

Ajuste de Perspectiva Automático Aplicado em Imagens de  
Gôndolas de Supermercado / Matheus André Soares ;  
orientador, Antônio Carlos Sobieranski, 2021.

36 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Engenharia de Computação, Araranguá, 2021.

Inclui referências.

1. Engenharia de Computação. 2. Ajuste de Perspectiva.  
3. Pontos de fuga. 4. Visão Computacional. 5. Gôndolas de  
Supermercado. I. Sobieranski, Antônio Carlos. II.  
Universidade Federal de Santa Catarina. Graduação em  
Engenharia de Computação. III. Título.

Matheus André Soares

**Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de Supermercado**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 17 de maio de 2021.

---

Prof. Fabrício De Oliveira Ourique, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Antônio Carlos Sobieranski, Dr.  
Orientador

---

Prof. Anderson Luiz Fernandes Perez, Dr  
Avaliador  
Universidade Federal de Santa Catarina

---

Luís Eduardo Ramos de Carvalho, Dr.  
Avaliador  
Involves Tecnologia

---

Prof. Fábio Rodrigues de la Rocha, Dr.  
Avaliador Suplente  
Universidade Federal de Santa Catarina



# Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de Supermercado

## *Automatic Perspective Adjustment Applied to Images of Supermarket Shelves*

Matheus André Soares \*      Antônio Carlos Sobieranski †

2021, Maio

### Resumo

Grande parte dos produtos presentes nas gôndolas de estabelecimentos comerciais não são notadas pelos consumidores. Por esse motivo, a indústria necessita criar e monitorar as estratégias utilizadas nos pontos de venda para aumentar o alcance das ações de marketing. Isso pode ser planejado utilizando fotos da gôndola dos locais, obtidas por promotores de vendas, e processadas a posteriori por métodos de processamento digital de imagens e reconhecimento de padrões. Entretanto, caso não for possível capturar uma imagem perpendicular ao plano frontal, pode haver perda de informações dos produtos apresentados na mesma. Nesse contexto, o trabalho em questão apresenta a modelagem de uma solução computacional totalmente automática cujo objetivo é realizar a correção da perspectiva presente em imagens de gôndolas de supermercados. Utilizando métodos computacionais já consolidados na literatura, baseados em pontos de fuga, foi possível obter um alinhamento adequado em 94% das imagens ajustadas e, em um *dataset* de validação, comprovou-se que as fotos corrigidas ficaram mais semelhantes às suas fotos frontais em todos os casos testados.

**Palavras-chaves:** Visão Computacional. Ajuste de Perspectiva. Pontos de Fuga. Gôndolas de supermercados.

---

\*matheus.soares@grad.ufsc.br

†a.sobieranski@ufsc.br

# Ajuste de Perspectiva Automático Aplicado em Imagens de Gôndolas de Supermercado

## *Automatic Perspective Adjustment Applied to Images of Supermarket Shelves*

Matheus André Soares <sup>\*</sup>      Antônio Carlos Sobieranski <sup>†</sup>

2021, Maio

### Abstract

Most of the products present on the shelves of commercial establishments are not noticed by consumers. For this reason, the industry needs to create and monitor the strategies used at points of sale to increase the reach of marketing actions. This can be planned using photos of the shelf, obtained by sales promoters and processed a posteriori by digital image processing and pattern recognition methods. However, if it is not possible to capture an image perpendicular to the frontal plane, there may be loss of information about the products presented on the shelf. In this context, the current study presents the modeling of a fully automatic computational solution whose goal is to perform the correction of the perspective inserted in supermarket shelf images. Using computational methods already consolidated in the literature, based on vanishing points, it was possible to obtain a proper alignment in 94% of the adjusted images and, in a validation *dataset*, it was verified that the corrected photos were more similar to their frontal photos in all cases tested.

**Key-words:** Computer Vision. Perspective Adjustment. Vanishing Points. Supermarket shelves.

---

<sup>\*</sup>matheus.soares@grad.ufsc.br

<sup>†</sup>a.sobieranski@ufsc.br

# 1 Introdução

De acordo com o relatório sobre o comportamento dos compradores, também conhecidos como *shoppers*, disponibilizado pela Nielsen<sup>1</sup>, empresa especializada em pesquisas sobre marketing, aproximadamente 70% das decisões de compras são feitas na frente da gôndola do supermercado e menos de 40% dos produtos expostos são notados (NIELSEN, 2015). Com essa informação, juntamente com o fato de que a indústria necessita cumprir com seus objetivos de venda, pode-se inferir que é necessário que toda cadeia de produção, distribuição e venda de produtos mantenha uma cooperação mútua para aumentar o alcance de suas ações de marketing. Para tal, uma análise profunda sobre todas as etapas de compra pode ser necessária, com informações sobre o público alvo, área de atuação, organização da gôndola, exposição no ponto de venda, precificação, ações de *merchandising*, assim como monitoramento constante sobre a evolução e impacto de cada uma dessas métricas (ALMEIDA et al., 2012).

Para realizar a análise supracitada, um dos recursos utilizados é a atuação de promotores e divulgadores diretamente no ponto de venda. Essa equipe tem como principal atividade garantir a manutenção adequada das mercadorias na gôndola e coletar informações pertinentes sobre a situação dos produtos no local para manter um histórico da situação dos mesmos (ARBACHE et al., 2011). Através dos dados coletados, a equipe de marketing e vendas consegue direcionar suas ações com foco em aproveitar oportunidades e ganhar novos consumidores (KOTLER; KELLER, 2013).

No âmbito da geração e consumo de dados, a fotografia se tornou fundamental no dia a dia das pessoas, servindo como fonte de lembrança, informação e registro da situação retratada (BATISTA, 2014). Como consequência, ao tirar uma foto do ponto de venda em uma visita, o promotor consegue fazer um registro histórico da situação dos produtos na gôndola, possibilitando que o time de Inteligência de Negócios realize uma coleta de informações mais específicas da situação fotografada (ARBACHE et al., 2011). Entretanto, enquanto a coleta de informações textuais pode possuir protocolos bem estabelecidos e mais facilitados, bastando apenas preencher campos predeterminados, as imagens podem não apresentar todos os dados necessários caso a pessoa que realizou a fotografia não cumpra todos os protocolos estabelecidos (LACERDA, 2012). Para o caso de fotos de gôndolas, o protocolo básico é tirar fotos de um modo frontal à prateleira para obter um registro claro de todos os produtos expostos. Na ocorrência de algum impedimento em cumprir esse procedimento, faz-se necessário realizar uma retificação na perspectiva inserida na imagem, deixando o plano da gôndola paralelo ao da foto.

Distorções na perspectiva da foto são causadas pela posição da câmera em relação ao objeto retratado na imagem. A sua principal característica é a transformação de linhas paralelas no plano real em linhas que convergem em um ponto, chamado de ponto de fuga, no plano da imagem. Os métodos clássicos dos algoritmos que realizam a correção na perspectiva são baseados em encontrar esses pontos e restaurar o paralelismo dessas retas (SANTANA-CEDRÉS et al., 2017). Algumas soluções são focadas em retificar imagens de documentos (ÁVILA; LINS, 2005)(TAKEZAWA; HASEGAWA; TABBONE, 2016) e outras focadas em corrigir a perspectiva de cenas que envolvam construções humanas (TSIRONIS et al., 2017)(LEE et al., 2012). Essas soluções têm se mostrado bastante efetivas no âmbito de seus focos de pesquisa, porém carecem de testes e desenvolvimento centrados em fotos de gôndolas de supermercados, uma vez que estas possuem características próprias que podem invalidar os métodos já propostos.

---

<sup>1</sup> <https://www.nielsen.com/br/pt/>

Com o objetivo de melhorar o aspecto da imagem para facilitar a visualização de todos os produtos expostos, assim como para deixar as fotos em uma conformidade mínima ao protocolo, o presente trabalho aborda o desenvolvimento de uma ferramenta capaz de remover as distorções de perspectiva inseridas em fotos de gôndolas de supermercado, tiradas por promotores de vendas. A utilização de pontos de fuga tem se mostrado extremamente promissora para a solução do problema abordado, uma vez que é característica inevitável de imagens em perspectiva, sendo uma técnica utilizada na presente pesquisa. Dessa forma, com a abordagem proposta foi possível obter um ajuste satisfatório de 94% das imagens, causando grandes distorções ou dificultando a utilização do restante das mesmas. Além disso, a partir de um conjunto de imagens de validação, confirmou-se uma maior semelhança da foto ajustada à fotos frontais se comparada com a foto angulada. Os resultados experimentais demonstram a viabilidade da técnica utilizada, trazendo mais tecnologia para maximizar o aproveitamento das fotos tiradas pelos promotores e aumentar a adequação das mesmas ao protocolo básico discutido anteriormente.

## 2 Trabalhos Correlatos

Uma revisão da literatura revelou que existem diversos trabalhos que abordam o tema ajuste de perspectiva automático. De uma maneira geral, três classes comuns puderam ser verificadas, sendo que podem ser classificadas com base no insumo utilizado para realizar a retificação da perspectiva. A primeira delas é mais comum, efetua a detecção de pontos de fuga através das linhas encontradas na imagem; a segunda busca encontrar alguma forma geométrica conhecida, mas distorcida, e remover essa deformação inserida; a última baseia-se no alinhamento de objetos e/ou linhas para encontrar o ângulo dos mesmos e realizar o ajuste. Dos métodos encontrados que se baseiam na detecção de pontos de fuga, destacam-se os trabalhos de:

- Chaudhury, DiVerdi e Ioffe em 2014, que utilizou um modelo baseado em RANSAC, para encontrar os melhores pontos de fuga e então utilizar uma matriz que desloca esses pontos para o infinito, chamada de  $H_P$ , retificando as linhas paralelas da imagem. Após esse processo, é realizada uma rotação 2D na imagem para alinhar o ponto de fuga vertical com o eixo Y da imagem, utilizando uma matriz de transformação isométrica, com vetor de translação nulo, o que, segundo os autores, pode aperfeiçoar a qualidade visual da imagem. Testes foram realizados em 2199 fotos randômicas obtidas por usuários da antiga rede social Google+, sendo que apenas 15,7% foram corrigidas de forma satisfatória, 1233 imagens tiveram ângulos incompatíveis entre os pontos de fuga encontrados e as outras tiveram muitas distorções (CHAUDHURY; DIVERDI; IOFFE, 2014).
- Santana-Cedres e demais autores em 2017, cuja proposta é corrigir automaticamente distorções ópticas e de perspectiva em imagens. Os autores primeiramente estimam as linhas distorcidas e o respectivo modelo que gera o efeito de lente. Após corrigir essa distorção, é utilizado um sistema de votação através de um histograma para encontrar as direções dominantes das linhas corrigidas e então calcular um ou dois pontos de fuga, utilizando a matriz de homografia  $H$  para realizar a retificação da perspectiva inserida (SANTANA-CEDRÉS et al., 2017).
- Yang, Fang e Tang em 2018, onde discutiram sobre a implementação de um algoritmo que utiliza pontos fuga para modificar a perspectiva de fotos de estradas e criar uma

visão aérea das mesmas. O objetivo principal dos autores é facilitar o processamento das informações adquiridas por câmeras em carros autônomos, aplicando a técnica de mapeamento por perspectiva inversa (*Inverse Perspective Mapping* - IPM) para criar uma visão superior da rodovia. Foi utilizado o mapa de regiões de saliência da imagem como informações primárias para um sistema de votação com o intuito de descobrir qual linha candidata corresponde ao melhor ponto de fuga. Após isso, foi modificada a perspectiva utilizando equações de mapeamento de pontos entre planos (YANG; FANG; TANG, 2018).

Os trabalhos que realizam a detecção de formas geométricas, apesar de não serem muito flexíveis de acordo com o cenário apresentado, ainda são bastante úteis naqueles casos em que as imagens em análise possuem algum padrão e poucos planos presentes. Os artigos em destaque são de:

- Kiran e Murali em 2013, que utilizaram erosão e dilatação da imagem em questão para encontrar quatro cantos formando um plano com formato de losango e transformá-lo em um plano retangular. Os autores aplicaram o operador Sobel para encontrar as bordas e então realizou-se as operações citadas anteriormente após binarizar a foto, calculando os cantos através dos pontos mínimos e máximos da imagem resultante. Os autores deixam claro que o resultado final pode conter distorções de escala, uma vez que o plano encontrado apenas é mapeado para um retângulo (KIRAN; MURALI, 2013).
- Li e outros autores no ano de 2017, apresentando uma abordagem semelhante a descrita no item anterior, mas buscando retificar círculos que se tornaram elipses devido à distorção de perspectiva inserida. O artigo apresenta uma solução específica para fotos de medidores de energia tiradas por robôs em subestações de energia com áreas de difícil acesso, com o objetivo de melhorar o reconhecimento das medições mostradas no equipamentos. Os autores realizam a binarização da imagem e a detecção de bordas através do algoritmo Canny, e então encontram elipses através da Transformada de Hough para finalmente realizar o ajuste na perspectiva (LI et al., 2017).
- Mantel e outros autores em 2020, que buscavam criar um método para estimar o ângulo e corrigir a perspectiva de imagens em eletroluminescência de painéis fotovoltaicos obtidas por *drones*. São apresentados dois algoritmos para alcançar tal objetivo: encontrar os quatro cantos do painel e encontrar os cantos de cada uma das células do painel. Para isso, a imagem foi normalizada e foi aplicado um filtro Sobel na mesma para então encontrar intersecções entre linhas em ambos os métodos propostos. Após isso, utilizou-se a biblioteca OpenCV<sup>2</sup> para realizar o cálculo da matriz de homografia através dos pontos encontrados (MANTEL et al., 2020).

Por outro lado, as pesquisas que utilizam orientação de objetos dependem da repetição dos mesmos para encontrar um padrão no ângulo formados por eles. Por esse motivo, esse método é mais utilizado em imagens de documentos, uma vez que possuem letras que se repetem com uma angulação semelhante. Assim sendo, destacam-se os trabalhos de:

---

<sup>2</sup> Open Source Computer Vision Library (<https://opencv.org>)

- Lu, Chen e Ko em 2005, cujo propósito era ajustar a perspectiva de fotos de documentos em geral, com o objetivo de melhorar a precisão no contexto de reconhecimento de caracteres - OCR (*Optical Character Recognition*). Para tal, foram utilizadas as características morfológicas das letras para calcular a orientação predominante do documento e então realizar a correção. Experimentos realizados pelos autores confirmaram que a técnica consegue lidar de forma satisfatória com diversos tipos de documentos, podendo conter fórmulas ou imagens também (LU; CHEN; KO, 2005).
- Golpardaz e Nezamabadi-Pour em 2011, que buscam retificar fotos de documentos escritos no idioma persa. O método utilizado é baseado em duas etapas, onde a primeira é uma etapa iterativa que busca ajustar a angulação da imagem através da minimização do cálculo da entropia da projeção da imagem ao longo do eixo horizontal. Em seguida, a perspectiva da imagem do documento é removida através da média dos ângulos formados pela conexão das letras em cada uma das linhas (GOLPARDAZ; NEZAMABADI-POUR, 2011).
- Li e outros autores em 2020, criando um *framework*, nomeado *Perspective Plane Program Induction*, ou P3I. Esse programa tem como objetivo receber como entrada uma imagem de objetos repetidos e obter uma visão holística da cena representada (sendo capaz de considerar texturas locais e a estrutura global da cena), além de construir também um simples programa que reproduz a estrutura repetitiva. Para corrigir a perspectiva, o algoritmo estima a posição da câmera a partir da recorrência dos objetos e das suas posições na imagem, fazendo então uma correção projetada para manter a coerência da foto (LI et al., 2020).

Através da análise dos métodos encontrados na literatura, percebe-se que aqueles baseados em formas geométricas e angulação de objetos em cena necessitam de fotos obtidas de um ambiente mais controlado, contendo formas geométricas preestabelecidas ou poucos planos. O mesmo não ocorre com a técnica baseada em pontos de fuga, sendo mais genérica a diversos ambientes, necessitando apenas de linhas paralelas que se tornaram convergentes na cena em análise. Uma vez que não foram encontrados métodos específicos para o contexto abordado neste trabalho (retificação de fotografias de gôndolas de supermercado), testes e adaptações às técnicas já propostas podem ser necessárias para validar a sua eficácia nas imagens de interesse. Por ser mais generalista, o procedimento que utiliza pontos de fuga se torna mais atraente para ser utilizado com o intuito de retificar as fotos em estudo, sendo o método escolhido para ser abordado no presente trabalho.

## 3 Fundamentação Teórica

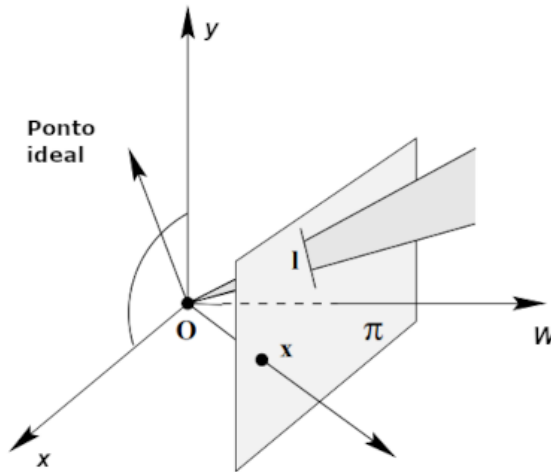
### 3.1 Representação de Pontos e Retas

Uma linha pode ser representada pela equação  $ax + by + c = 0$  ou, de uma forma simplificada, por um vetor do tipo  $(a, b, c)$ . Diferentes valores de coeficientes geram diferentes retas, mas as retas  $(a, b, c)$  e  $(ka, kb, kc)$  são as mesmas para valores não nulos de  $k$ , onde essa constante representa um múltiplo (HARTLEY; ZISSERMAN, 2004).

Um ponto  $\mathbf{P} = (x, y)$  pertence à reta formada pelo vetor  $\mathbf{l} = (a, b, c)$  se e, somente se,  $ax + by + c = 0$  ou se o produto escalar  $(x, y, 1) \cdot (a, b, c)$  forem iguais à zero. A reta passando por dois pontos  $\mathbf{P}$  e  $\mathbf{Q}$  é dada pelo produto vetorial  $\mathbf{P} \times \mathbf{Q}$  (LIEBOWITZ, 2001).

O espaço projetivo é uma extensão do espaço Euclidiano onde há a adição de pontos ideais ou pontos localizados no infinito. Nele, os pontos são representados por coordenadas homogêneas, através de um vetor  $(x, y, w)$  e a sua conversão a partir de coordenadas cartesianas se dá atribuindo  $w = 1$ . O contrário se faz através de  $(\frac{x}{w}, \frac{y}{w})$  quando  $w$  for diferente de zero. Os pontos que possuem  $w = 0$  são chamados de pontos ideais, conforme representado na [Figura 1](#) (FOLEY et al., 1996).

Figura 1 – Representação do plano 2D em coordenadas homogêneas.



**Fonte:** Traduzido e adaptado de [Hartley e Zisserman \(2004\)](#)

Dado duas linhas  $l$  e  $l'$ , o produto vetorial entre elas tem como resultado um vetor que representa um ponto em coordenadas homogêneas. Como descrito anteriormente, um ponto pertence à uma reta caso o produto escalar entre os mesmos for igual a zero. Devido às propriedades do produto misto entre vetores, podemos afirmar que  $l \cdot (l \times l') = l' \cdot (l \times l') = 0$ . Isso indica que a resultante do produto vetorial entre duas retas é o ponto de intersecção pois pertence às duas retas simultaneamente ([HARTLEY; ZISSERMAN, 2004](#); [WINTERLE, 2000](#)).

### 3.2 Transformações Geométricas em Imagens Digitais

Segundo [Hartley e Zisserman \(2004\)](#), as transformações geométricas em imagens de duas dimensões podem ser classificadas em classes. Essas classes possuem uma hierarquia, onde a primeira é a mais especializada em um tipo de distorção ou efeito e a última é a mais genérica. São quatro classes e as mesmas serão explicadas nas subseções a seguir.

#### 3.2.1 Transformações Isométricas

As transformações isométricas, também chamadas de isometrias ou transformações Euclidianas, são mudanças que preservam as distâncias Euclidianas da imagem. Uma isometria é representada, em sua forma completa, pela matriz da [Equação 1](#) ([HARTLEY; ZISSERMAN, 2004](#)).

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

Nesse tipo de transformação, só é introduzido rotação e translação, mantendo as proporções do objeto. O parâmetro  $\epsilon$  indica a orientação da transformação, podendo ser igual a  $\pm 1$ . Um valor positivo preserva a orientação da imagem e um valor negativo inverte a mesma. A matriz apresentada na [Equação 1](#) pode ser dividida, de uma forma mais concisa, através de uma matriz puramente de rotação  $\mathbf{R}$  com dimensão  $2 \times 2$ , um vetor translação  $\mathbf{t}$  e um vetor nulo  $\mathbf{0}$ . Assim, a forma resumida da matriz de transformação isométrica pode ser escrita como representada pela [Equação 2](#).

$$\mathbf{x}' = H_E \mathbf{x} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (2)$$

### 3.2.2 Transformações de Similaridade

Uma transformação de similaridade ou uma similaridade é uma transformação que combina rotação, translação e escala. Sua matriz de representação é a mostrada na [Equação 3](#) (SOYCAN; SOYCAN, 2019).

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3)$$

Pode ser considerada uma isometria sem o componente  $\epsilon$  e com um fator de escala  $s$  aplicado na matriz de rotação  $\mathbf{R}$ , como visto na simplificação da matriz na [Equação 4](#). Ao aplicar essa transformação em uma imagem, linhas que são retas e paralelas permanecem com essas características, apenas mudam de posição na imagem representada (HARTLEY; ZISSERMAN, 2004; SOYCAN; SOYCAN, 2019).

$$\mathbf{x}' = H_S \mathbf{x} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \quad (4)$$

### 3.2.3 Transformações Afins

Uma transformação afim, ou afinidade, introduz maiores deformações na imagem. Ela é composta por uma matriz de transformação linear  $\mathbf{A}$  de tamanho  $2 \times 2$  e um vetor de translação  $\mathbf{t}$ . Nesse tipo de transformação, duas retas paralelas que anteriormente se intersectavam em um ponto no infinito, são transformadas em retas também paralelas que se encontram em um diferente ponto no infinito, ou seja, uma afinidade preserva o paralelismo de retas (HARTLEY; ZISSERMAN, 2004; SOYCAN; SOYCAN, 2019).

Seu formato pode ser apresentado como:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (5)$$

e sua forma reduzida em blocos pode ser escrita como:

$$\mathbf{x}' = H_A \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}. \quad (6)$$



### 3.2.4 Transformações Projetivas

Transformação projetiva, projetividade ou homografia, é uma transformação linear de coordenadas homogêneas. Pode ser apresentada como uma generalização de uma afinidade e sua forma resumida é mostrada na [Equação 7](#), sendo  $\mathbf{v}$  um vetor com duas posições e  $v$  um número real. Geralmente, a matriz é parametrizada em função de  $v$ , deixando essa variável unitária. Entretanto, nem sempre é possível realizar tal operação, uma vez que  $v$  pode ser nulo ([HARTLEY; ZISSERMAN, 2004](#)).

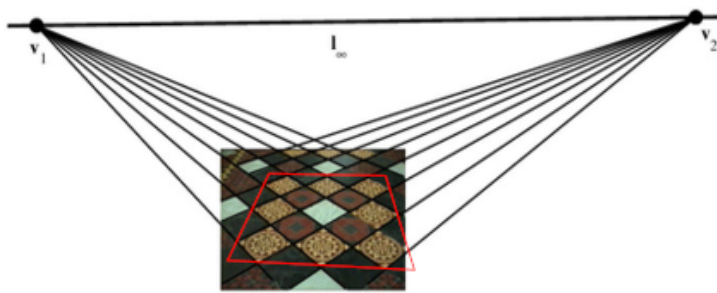
$$\mathbf{x}' = H_P \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{x} \quad (7)$$

Essa matriz tem como característica garantir que os pontos que pertencem a uma mesma linha antes da transformação, continuem colineares após serem transformados. Além disso, principalmente em fotos que retratam estruturas feitas pelo homem, uma projetividade tem a característica de transformar retas paralelas em retas concorrentes que convergem no ponto de fuga, ou *vanishing point*, como visto na [Figura 2](#). Isso se deve ao fato de que estruturas 3D estão sendo mapeadas em um plano de duas dimensões ([SOYCAN; SOYCAN, 2019](#); [SZELISKI, 2010](#)). Assim, segundo [Rezende e Stolfi \(1994\)](#), toda foto pode ser considerada uma transformação projetiva do plano da cena real.

Figura 2 – Comportamento de linhas paralelas sob transformações projetivas.



(a) Padrões em cerâmicas, onde círculos e quadrados são representados como tal.



(b) Sob transformações projetivas, linhas paralelas tornaram-se concorrentes, com pontos de fuga em  $v_1$  e  $v_2$ , e círculos tornaram-se elipses.

**Fonte:** Adaptado de [Hartley e Zisserman \(2004\)](#), [Liebowitz \(2001\)](#)

Por compor todas as outras classes anteriores, a transformação projetiva geralmente é generalizada como uma matriz  $H$  denominada somente como matriz de homografia, possuindo a estrutura vista na [Equação 8](#) ou, de forma simplificada  $\mathbf{x}' = H\mathbf{x}$ . Como citado anteriormente, a matriz pode ser parametrizada em função de  $h_{33}$ , restando, então, outros oito valores que podem ser encontrados com pelo menos quatro pontos conhecidos em ambos os sistemas referenciais (ponto  $x$  na origem e sua correspondente localização destino  $x'$ ) ([SOYCAN; SOYCAN, 2019](#)).

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}_{3 \times 3} \quad (8)$$

Através dos quatro pares de pontos é possível resolver a matriz homografia  $H$ , uma vez que os mesmos levam à oito equações lineares, como demonstrado por Hartley e Zisserman em 2004, considerando  $h_{33} = 1$  (HARTLEY; ZISSERMAN, 2004).

### 3.2.5 Decomposição da Matriz de Homografia

Como citado anteriormente, a matriz de homografia é formada por uma multiplicação de outras transformações. Ela pode ser decomposta em outras três: matriz de similaridade, de transformação afim e de transformação puramente projetiva, como mostrado na Equação 9 (LIEBOWITZ; ZISSERMAN, 1998).

$$\mathbf{H} = H_S H_A H_P = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \quad (9)$$

A função da matriz projetiva, neste caso, é transformar as linhas concorrentes em linhas paralelas. Isso é alcançado ao utilizar a linha de fuga do plano  $l_\infty = (l_1, l_2, l_3)$ , que é a linha no infinito da cena retratada na foto e pode ser encontrada através da conexão de dois pontos de fuga da cena. A reta  $l_\infty$  é a linha do horizonte percebida na cena real representada na imagem (LIEBOWITZ; ZISSERMAN, 1998).

Sob uma transformação afim, círculos são representados como elipses e não é possível tratá-los de forma distinta, ou seja, no plano afim só existe o conceito de elipse. A intersecção de duas destas produz quatro pontos em comum mas, geometricamente, dois círculos têm no máximo dois pontos de intersecção. Observando a sua equação em coordenadas homogêneas e com centro em  $(a, b, 1)$ , que tem a forma

$$(x - aw)^2 + (y - bw)^2 = r^2 w^2, \quad (10)$$

pode-se perceber que os pontos  $(x, y, w) = (1, \pm i, 0)$  são soluções viáveis para quaisquer valores de  $(a, b, r)$ . Portanto, fazem parte de todos os círculos e estão presentes na intersecção de qualquer par dos mesmos, completando os quatro pontos de intersecção de dois círculos (que são tratados como elipses) no plano afim. Esse par conjugado de pontos com coordenadas complexas está situado no infinito, uma vez que  $w = 0$ , e são chamados de *circular points*. Sob uma transformação afim, esses dois pontos  $\mathbf{I}$  e  $\mathbf{J}$  são transladados para  $(\alpha \mp i\beta, 1, 0)$ . Com isso, os parâmetros  $\alpha$  e  $\beta$  podem ser utilizados na matriz de transformação afim  $H_A$  para recuperar as propriedades da imagem em questão (HARTLEY; ZISSERMAN, 2004; REY, 2006).

Os *circular points* da imagem afim são encontrados pela intersecção de dois círculos plotados no plano complexo  $(\alpha, \beta)$  (LIEBOWITZ, 2001). Cada um dos círculos podem ser encontrados através de três tipos de informações:

- a) Ângulo conhecido entre linhas: sendo  $\theta$  o ângulo entre duas linhas no mundo real que foram transformadas em  $l_1 = (l_{1x}, l_{1y}, l_{1w})$  e  $l_2 = (l_{2x}, l_{2y}, l_{2w})$  através de uma transformação afim e considerando que  $a = -l_{1y}/l_{1x}$  e  $b = -l_{2y}/l_{2x}$  são as inclinações das retas, temos um círculo com centro

$$(c_\alpha, c_\beta) = \left( \frac{(a+b)}{2}, \frac{(a-b) \cot \theta}{2} \right) \quad (11)$$

e raio

$$r = \left| \frac{(a-b)}{2 \sin \theta} \right|. \quad (12)$$

- b) Ângulos iguais porém desconhecidos: considerando que existam dois pares de retas que deveriam possuir ângulos iguais mas, por estarem representadas sob transformação afim, são diferentes, com inclinações  $a_1$ ,  $b_1$  e  $a_2$ ,  $b_2$ , podemos encontrar um círculo no plano  $(\alpha, \beta)$  com centro em

$$(c_\alpha, c_\beta) = \left( \frac{a_1 b_2 - b_1 a_2}{a_1 - b_1 - a_2 + b_2}, 0 \right) \quad (13)$$

e raio

$$r^2 = \left( \frac{a_1 b_2 - b_1 a_2}{a_1 - b_1 - a_2 + b_2} \right)^2 + \frac{(a_1 - b_1)(a_1 b_1 - a_2 b_2)}{a_1 - b_1 - a_2 + b_2} - a_1 b_1. \quad (14)$$

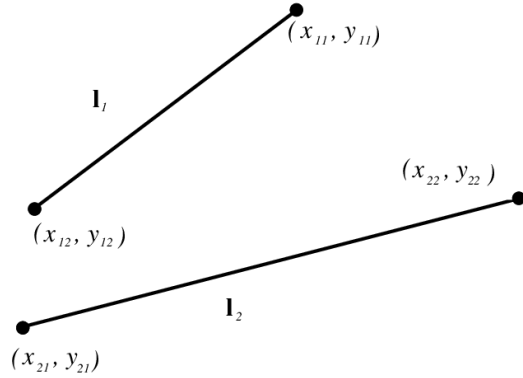
- c) Razão conhecida entre comprimento de segmentos de retas: considerando que  $s$  é a razão entre o comprimento de duas retas na cena real e estão representadas sob transformação afim como mostrado na [Figura 3](#) e tendo  $\Delta x_n = x_{n1} - x_{n2}$  e  $\Delta y_n = y_{n1} - y_{n2}$ , pode-se encontrar um círculo com centro em

$$(c_\alpha, c_\beta) = \left( \frac{(\Delta x_1 \Delta y_1 - s^2 \Delta x_2 \Delta y_2)}{\Delta y_1^2 - s^2 \Delta y_2^2}, 0 \right) \quad (15)$$

e raio

$$r^2 = \left| \frac{s(\Delta x_2 \Delta y_1 - \Delta x_1 \Delta y_2)}{\Delta y_1^2 - s^2 \Delta y_2^2} \right|. \quad (16)$$

Figura 3 – Representação de duas retas sob transformação afim.



**Fonte:** Retirado de [Liebowitz \(2001\)](#)

Utilizando as restrições supracitadas, é possível descobrir os pontos de intersecção dos dois círculos no plano  $(\alpha, \beta)$ . Após recuperar as chamadas propriedades não-métricas através das matrizes de transformação projetiva e afim, a imagem fica apenas com escala, rotação e traslação diferentes da cena representada, conhecidas como propriedades métricas ([LIEBOWITZ; ZISSERMAN, 1998](#)).

### 3.3 Detecção de Bordas

A detecção bordas é uma área importante principalmente para a segmentação de imagens, mas também é a base para muitos outros algoritmos de processamento de

imagens. Ela é fundamentada no fato de que bordas de objetos em cena ocorrem entre regiões de diferentes cores, intensidades ou texturas (SZELISKI, 2010).

Segundo Gonzalez e Woods (2010), a primeira e a segunda derivada são grandes aliadas na análise da continuidade dos pixels em uma imagem e, como estamos lidando com uma função digital, a derivada é apenas a diferença de dois fatores. A primeira derivada tem como propriedades:

$$f'(x) \begin{cases} = 0, & \text{em áreas com intensidade constante;} \\ \neq 0, & \text{em início de degraus ou rampas e quando em rampas.} \end{cases} \quad (17)$$

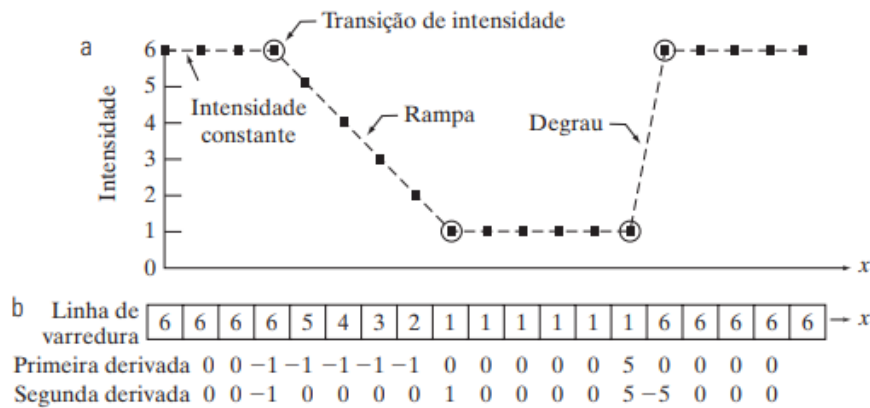
Já as propriedades da segunda derivada são:

$$f''(x) \begin{cases} = 0, & \text{em áreas com intensidade constante ou rampas com inclinação constante;} \\ \neq 0, & \text{em inícios e finais de degraus ou rampas.} \end{cases} \quad (18)$$

Essas propriedades podem ser confirmadas na Figura 4, onde a primeira e segunda derivadas são as diferenças:

$$\frac{df}{dx} = f(x+1) - f(x) \quad \text{e} \quad \frac{d^2f}{dx^2} = f(x+1) + f(x-1) - 2f(x). \quad (19)$$

Figura 4 – (a) Exemplo de uma função digital com uma dimensão representando a intensidade de uma imagem e suas características. (b) Primeira e segunda derivada da função representada em (a).



Fonte: Retirado de Gonzalez e Woods (2010)

Em uma imagem digital, muitas vezes as bordas se assemelham à rampas em sua intensidade. Utilizando a primeira derivada de tal imagem, obteríamos como resultado uma borda grossa, devido à sua propriedade de ser diferente de zero em rampas. Ao utilizar a segunda derivada, o resultado seria uma borda dupla com espessura de um pixel, uma vez que no início e fim da rampa em questão teríamos valores diferentes de zero rodeados de valores nulos (GONZALEZ; WOODS, 2010).

### 3.3.1 O Gradiente

Um operador bastante utilizado no processamento de imagens para a implementação da primeira derivada é o gradiente (GONZALEZ; WOODS, 2010; PEDRINI; SCHWARTZ,

2007). Sua direção indica os locais que sofreram maior variação na posição  $(x, y)$  e pode ser encontrado através do vetor coluna:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (20)$$

A magnitude  $M(x, y)$  do vetor  $\nabla f$  indica a taxa de variação na direção do gradiente, podendo ser encontrada através da equação [Equação 21](#). Vale ressaltar que  $g_x$ ,  $g_y$  e  $M(x, y)$  geram imagens com a mesma dimensão espacial que a original, uma vez que são iterações por cada um dos valores de  $(x, y)$  possíveis ([GONZALEZ; WOODS, 2010](#)).

$$M(x, y) = \sqrt{g_x^2 + g_y^2} \quad (21)$$

A [Equação 20](#) implementa derivadas parciais e é necessário uma simplificação para usá-las em imagens, uma vez que pode ser considerada uma função discreta ([GONZALEZ; WOODS, 2010](#)). Uma boa e conhecida aproximação pode ser encontrada na [Equação 22](#).

$$g_x = f(x + 1, y) - f(x, y) \quad g_y = f(x, y + 1) - f(x, y) \quad (22)$$

Uma forma simples de usar as equações demonstradas anteriormente em imagens é através de máscaras (matrizes  $n \times m$ ). Elas podem possuir diversos tamanhos e são deslocadas por todos os pixels da imagem. A máscara aplicada em um pixel de coordenada  $(x, y)$  estará sempre localizada sob o elemento central da matriz e os outros parâmetros correspondem aos pixels vizinhos ([GONZALEZ; WOODS, 2010](#)).

Assim sendo, as máscaras resultantes do gradiente mostrado na [Equação 20](#) serão matrizes de uma dimensão mostradas a seguir:

$$g_x = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 & 1 \end{bmatrix}. \quad (23)$$

Muitas outras máscaras foram implementadas para realizar a simplificação das derivadas parciais que revelam as bordas entre regiões de diferentes intensidades. As principais são mostradas na [Figura 5](#). As máscaras mais úteis para o cálculo da direção da borda são as simétricas ao redor do ponto central, com tamanho  $3 \times 3$ , pois levam em consideração a natureza dos dados ao redor do ponto que está sendo considerado, sendo aproximações mais precisas ([GONZALEZ; WOODS, 2010](#); [PEDRINI; SCHWARTZ, 2007](#)).

### 3.3.2 O Detector de Bordas Canny

Em 1986, Canny propôs um método mais complexo para encontrar bordas, focando em diminuir a taxa de erros, reduzir a influência do ruído presente na imagem e aperfeiçoar a localização da detecção. No ano de 2011, [Radha e Muthukrishnan \(2011\)](#) citaram que esse algoritmo era um dos mais utilizados na indústria para detecção de bordas.

A proposta de [Canny \(1986\)](#) inicia com uma suavização da imagem através de um filtro Gaussiano antes de realizar o cálculo do gradiente. Esse filtro tem como função

Figura 5 – Diversas máscaras utilizadas para a detecção de bordas para operações de convolução.

-1	0	0	-1
0	1	1	0

Roberts

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

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

Sobel

**Fonte:** Retirado de [Gonzalez e Woods \(2010\)](#)

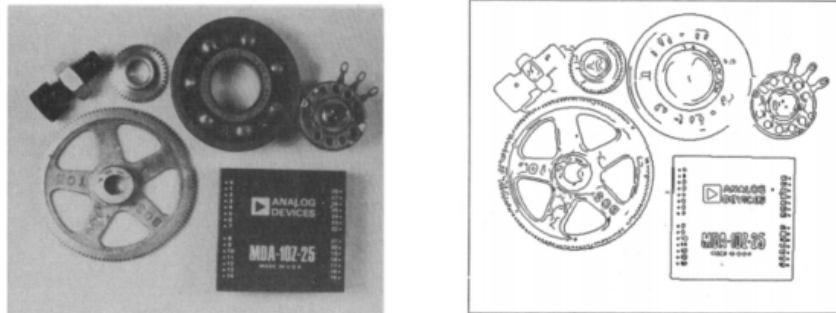
atenuar um possível ruído presente na borda, modelado idealmente como um ruído branco Gaussiano, o que experimentalmente gera uma boa aproximação daquele presente em muitas situações do mundo real. O tamanho do filtro é parametrizável, mas considera-se que quanto maior, mais suavizado será o resultado, podendo espalhar a influência dos pixels e conseqüentemente produzir resultados negativos nos próximos passos do algoritmo ([GONZALEZ; WOODS, 2010](#); [RADHA; MUTHUKRISHNAN, 2011](#)).

Após a suavização da imagem de entrada, é realizado o cálculo do gradiente através de uma das máscaras apresentadas na Subseção 3.3.1 e da magnitude e direção do mesmo, utilizando as fórmulas mostradas na mesma seção. O gradiente utiliza a primeira derivada e, como descrito anteriormente, o seu resultado sobre uma rampa produz bordas grossas. O próximo passo proposto por Canny é afinar essas características obtidas através da supressão dos não máximos. Essa técnica filtra os pontos que não são máximos localmente, deixando como resultado somente aqueles cuja magnitude se destaca no local analisado, reduzindo a espessura da borda ([PEDRINI; SCHWARTZ, 2007](#); [GONZALEZ; WOODS, 2010](#)).

Ao realizar a etapa de supressão dos não máximos, alguns fragmentos ou ruídos podem continuar presentes. Para reduzir esses erros, são utilizados dois parâmetros diferentes,  $T_1$  e  $T_2$  para aplicar uma limiarização com histerese, considerando  $T_2 > T_1$ . Através dessa técnica, os pontos que possuem magnitude maior que  $T_2$  são considerados pontos de borda,

qualquer outro pixel ligado à um ponto de borda só será mantido se sua magnitude tiver valor maior que  $T_1$  e pontos que são menores que o limiar inferior são automaticamente descartados (MACHADO, 2008; GONZALEZ; WOODS, 2010). Um exemplo do resultado do algoritmo apresentado é mostrado na Figura 6.

Figura 6 – Exemplo de aplicação do algoritmo Canny em uma imagem.



(a) Imagem original.

(b) Resultado do algoritmo proposto por Canny aplicado em (a) utilizando limiarização com histerese.

**Fonte:** Retirado de [Canny \(1986\)](#)

### 3.4 Detecção de Linhas

Encontrar bordas produz uma série de pontos que caracterizam o contorno de um objeto. Raramente esses pixels encontrados reproduzem fielmente as bordas esperadas, uma vez que o ruído e variações na intensidade podem influenciar bastante no resultado, gerando quebra na continuidade das bordas. Assim, geralmente é preferível realizar um ligamento e aperfeiçoamento do resultado encontrado, conectando em linhas para uma maior facilidade na utilização dos dados (GONZALEZ; WOODS, 2010).

Dois métodos se destacam para realizar esse processo, sendo eles através de aproximação por polilinhas de forma iterativa e através da transformada de Hough.

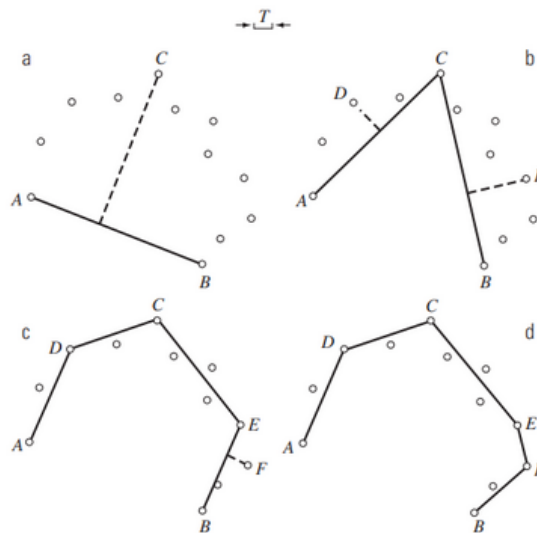
#### 3.4.1 Aproximação Iterativa por Polilinha

Uma técnica bastante simples para o ajuste de uma polilinha em um conjunto de pontos se dá através de sucessivas aproximações. Neste caso, são feitas várias tentativas para ligar os pixels encontrados que representam as bordas. Em cada iteração, é feito um cálculo para analisar a performance da hipótese criada. Caso o resultado for maior que determinado valor, é realizado uma nova tentativa (SZELISKI, 2010; GONZALEZ; WOODS, 2010).

Pode-se utilizar diversos algoritmos para essa tarefa, mas um deles pode ser exemplificado utilizando a Figura 7. Inicialmente, é feita uma tentativa ligando os pontos A e B. Após, é calculada a distância entre todos os outros pontos e a reta e escolhido aquele que resultou no maior valor. Caso esse valor for maior que determinado limiar, esse ponto começa a fazer parte da reta na próxima iteração. No exemplo, o ponto C foi escolhido para se tornar um vértice da reta. Assim, é calculada a distância dos pontos entre A e C e a reta AC e o mesmo ocorre para a reta CB. Caso algum ponto for maior que o limiar

estabelecido, o processo se inicia novamente com o pixel escolhido como novo vértice. Caso contrário, o algoritmo finaliza suas iterações com a melhor aproximação de acordo com os parâmetros definidos (GONZALEZ; WOODS, 2010).

Figura 7 – Exemplo de um algoritmo que realiza a aproximação de linhas através de pixels



**Fonte:** Retirado de Gonzalez e Woods (2010)

Segundo Gonzalez e Woods (2010), esse algoritmo é fácil de implementar e gera resultados aceitáveis para a grande maioria dos casos, mas por ser iterativo, pode ser computacionalmente oneroso, existindo outras técnicas atualmente que utilizam mais informações e podem ser mais precisas.

### 3.4.2 Transformada de Hough

Muitas vezes, a detecção de linhas que considera apenas componentes regionais, como a técnica apresentada anteriormente, pode levar a soluções desconexas, uma vez que uma linha pode ser composta por vários componentes que não estão conectados. Levando isso em conta, torna-se necessária uma abordagem que considera propriedades globais da imagem para realizar a conexão de pontos em linhas (SZELISKI, 2010; GONZALEZ; WOODS, 2010).

Hough (1962) propôs uma abordagem que utiliza informações globais para detectar formas geométricas com equações conhecidas utilizando um espaço diferente, criando então a conhecida Transformada de Hough.

Podemos utilizar coordenadas polares para representar uma reta qualquer que passa por um ponto  $(x, y)$ , levando-nos à Equação 24. Com essa informação, podemos utilizar o plano  $\rho\theta$ , também conhecido como espaço dos parâmetros, para sumarizar todo conjunto de retas que podem passar pelo ponto em questão, gerando formas senoidais, como exemplificado na Figura 8. O ponto em destaque na Figura 8(b) indica os parâmetros  $\rho$  e  $\theta$  da reta que passa por todos os pontos mostrados na Figura 8(a) (GONZALEZ; WOODS, 2010).

$$\rho = x \cos \theta + y \sin \theta \quad (24)$$

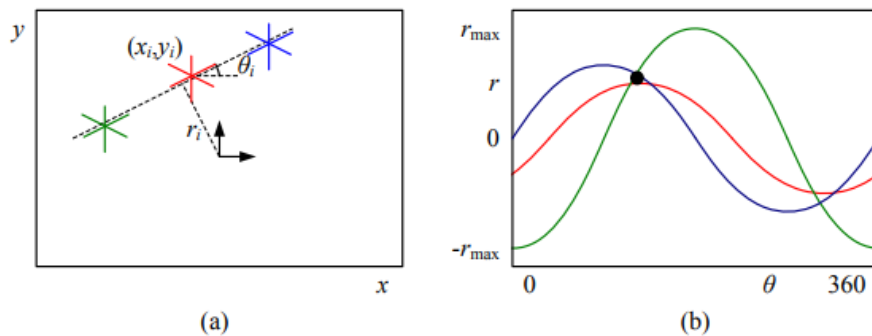


O grande diferencial da Transformada de Hough é utilizar um sistema de votação chamado células acumuladoras para encontrar os parâmetros  $\rho$  e  $\theta$  que possam corresponder a possíveis localizações das linhas. Assim, o plano  $\rho\theta$  é subdividido em células, sendo que a célula na coordenada  $(i, j)$  possui um acumulador  $A(i, j)$  associado aos parâmetros  $(\rho_i, \theta_j)$ . Assim sendo, para cada ponto a ser observado no plano  $xy$ , é construída a curva senoidal correspondente no espaço de parâmetros e é incrementado em uma unidade o acumulador de cada célula a qual a curva sobrepõe. Ao final desse processo, um valor de  $P$  no acumulador  $A(i, j)$  indica que  $P$  pontos analisados interseccionam a reta  $\rho_i = x \cos \theta_j + y \sin \theta_j$  (GONZALEZ; WOODS, 2010; SONKA; HLAVAC; BOYLE, 2014).

Muitas alterações podem ser feitas no processo de votação, como utilizar a intensidade dos pixels ou até mesmo o tamanho do segmento de reta que um ponto pode formar para pontuar positiva ou negativamente no seu correspondente acumulador. A quantidade de subdivisões do espaço de parâmetros altera a precisão a ser considerada na colinearidade dos pontos (SZELISKI, 2010).

Um grande benefício da Transformada de Hough é a sua robustez ao lidar com ruído ou partes desconexas das bordas. Isso é obtido ao utilizar o sistema de votação através das células acumuladoras, uma vez que caso falte alguma parte da borda, o acumulador não será tão afetado e um ruído ou uma borda muito ruidosa irá apenas contribuir para diversas células e seu centro de gravidade poderá ser considerado como uma representação da curva em questão (SONKA; HLAVAC; BOYLE, 2014).

Figura 8 – (a) Exemplificação de três pontos e uma reta que liga os mesmos, tomando como origem o centro da imagem. (b) Representação no plano  $\rho\theta$  dos pontos mostrados em (a). A localização da intersecção de todas as curvas senoidais representa uma aproximação da reta que liga os pontos.



Fonte: Retirado de Szeliski (2010)

### 3.5 Detecção de Pontos de Fuga

Muitos são os meios que podem ser utilizados para encontrar os pontos de fuga em uma imagem, como algoritmo de maximização de expectativa (NIETO; SALGADO, 2011), Transformada de Hough (CHEN et al., 2010), RANSAC (CHAUDHURY; DIVERDI; IOFFE, 2014) ou através do mapeamento de linhas em uma esfera Gaussiana (ROTHER, 2002). Entretanto, para manter este trabalho conciso, iremos explorar somente as duas últimas técnicas supracitadas, que são as que mais se destacam nos artigos encontrados na literatura.

### 3.5.1 Random Sample Consensus - RANSAC

RANSAC foi introduzido em 1981 por Fischler e Bolles como um algoritmo iterativo para estimar os parâmetros apropriados de acordo com um modelo determinado (FISCHLER; BOLLES, 1981). O mesmo atualmente pode ser usado para várias aplicações, como descobrir a matriz fundamental, encontrar correspondência entre pontos, calcular a matriz homografia, entre outros.

O método é baseado na premissa de que, dado um conjunto de dados de entrada, há aqueles que estão de acordo com um modelo estabelecido, chamados de *inliers*, e aqueles que não se enquadram no modelo, conhecidos como *outliers*. Segundo o trabalho de Janicka e Rapinski (2014), o RANSAC pode ser capaz de encontrar uma boa solução mesmo com mais de 50% do conjunto de entrada sendo *outlier*.

Através de um processo iterativo, o algoritmo possui basicamente duas etapas, sendo elas: estimação do modelo de hipótese e cálculo do erro. Como insumo inicial, devem ser informados uma lista de dados, o modelo a ser seguido, a quantidade mínima  $n$  de dados apropriados para encontrar um possível modelo e um valor de *threshold* com o qual é possível separar os *outliers* dos *inliers*, de acordo com o modelo informado (JANICKA; RAPINSKI, 2014).

A partir dos dados de entrada, é iniciada a primeira etapa com a seleção aleatória de um subconjunto de dados contendo  $n$  itens, também chamado de conjunto de hipótese, e cálculo do modelo que se adequa a esse subconjunto. Posteriormente, a etapa de cálculo do erro é realizada através da filtragem dos *outliers* utilizando o *threshold* dado como entrada e depois estima-se o erro dos *inliers* restantes para com o modelo hipotético. Após isso, novamente um subconjunto aleatório é selecionado e o processo é reiniciado, seguindo esse fluxo iterativo até que seja satisfeita alguma condição de parada, como um erro mínimo ou quantidade máxima de repetições (BOTTERILL; MILLS; GREEN, 2011).

O cálculo do erro empregado na segunda etapa varia de acordo com a aplicação à qual o algoritmo está sendo submetido, mas fórmulas utilizadas comumente estão relacionadas à distância Euclidiana. Um exemplo bastante utilizado na literatura é o problema de encontrar uma reta que representa um conjunto aleatório de pontos, apresentado na Figura 9. Nesse caso, a linha pontilhada mostrada é o *threshold* que separa os *inliers* dos *outliers*. Os dois pontos azuis foram escolhidos aleatoriamente como modelo hipotético, o qual obteve o menor erro dentre todos os outros testes realizados.

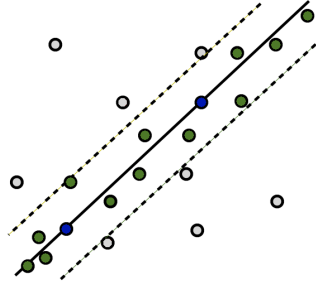
### 3.5.2 A Esfera Gaussiana

Em 1983, Barnard propôs usar um mapeamento Gaussiano para representar a orientação de linhas e planos de uma imagem (BARNARD, 1983). Essa transformação projeta o plano da imagem em uma esfera de raio unitário e, como consequência, segmentos de retas da cena são transformadas em círculos na esfera (ROTHER, 2002).

Na esfera em questão, como todas as retas são transformadas em círculos, os pontos de intersecção de dois destes são considerados pontos de fuga, como visto na Figura 10. Caso duas linhas paralelas no plano da imagem forem mapeadas para a esfera Gaussiana, seus pontos de encontro se localizarão nos polos da mesma, significando que estão presentes no infinito (YIN et al., 2011).

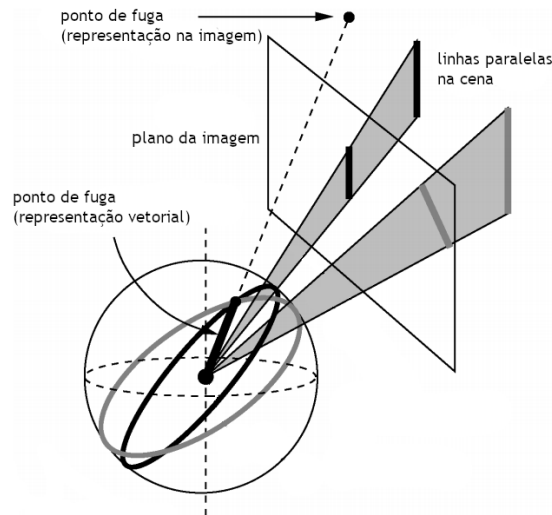
Uma vez que as linhas presentes em uma imagem podem gerar vários pontos de fuga, é necessário estabelecer um consenso para escolher o que melhor se encaixa na imagem.

Figura 9 – Exemplo de uma execução do algoritmo RANSAC para encontrar a reta que melhor descreve um conjunto de pontos. Aqueles que estão em azul foram selecionados aleatoriamente para compôr o modelo hipotético que obteve o menor erro dentre os demais e os verdes são os *inliers* desse modelo.



**Fonte:** Adaptado de [Hast, Nysjö e Marchetti \(2013\)](#)

Figura 10 – Mapeamento do plano de uma imagem em uma esfera Gaussiana.



**Fonte:** Adaptado de [Shufelt \(1999\)](#)

Para isso, a esfera geralmente é considerada como um espaço acumulador e dividida em células. Aquela que contiver mais votos, ou intersecções de retas, é considerada como localização do ponto de fuga ([CANTONI et al., 2001](#)).

Entretanto, [Shufelt \(1999\)](#) demonstrou que essa técnica pode levar à pontos de fuga errôneos, uma vez que o ângulo de visão, distância para o plano da cena ou até mesmo a presença de padrões na imagem podem gerar fracos efeitos de perspectiva. Isso levou outros autores a procurarem novos métodos para corrigir essa falha, como [Peng e Shi \(2007\)](#), que propuseram utilizar uma máscara na imagem para filtrar as áreas de interesse, mapeando a direção das suas características mais proeminentes em uma esfera Gaussiana.

## 4 Desenvolvimento do Método Proposto

No diagrama da [Figura 11](#), o fluxo geral da solução computacional proposta é sumarizado. Em azul (a) são mostradas as fases de leitura e pré-processamento da foto obtida pelo promotor de vendas; em amarelo (b) estão as etapas referentes à detecção das linhas; em vermelho (c), estão destacadas as etapas onde são encontrados os pontos de fuga; em verde (d) os ângulos verticais e horizontais da imagem são retificados; e em roxo (e) a matriz de homografia é aplicada à foto e é realizada uma etapa de pós-processamento da mesma.

Para o desenvolvimento do método proposto, foi utilizada a linguagem de programação C++, com o auxílio da biblioteca *OpenCV*, facilitando o uso de algoritmos básicos relacionados à visão computacional. As etapas mencionadas anteriormente e mostradas na [Figura 11](#) serão apresentadas em detalhes nas subseções seguintes.

### 4.1 Pré-Processamento da Imagem

A primeira etapa do algoritmo desenvolvido é a leitura da imagem obtida pelo promotor no ponto de venda. Nessa etapa, é importante salientar que a foto é captada exclusivamente através de câmeras presentes em *smartphones*, podendo-se então desconsiderar a presença de outras distorções na imagem, como distorções ópticas. Além disso, não foram aplicadas restrições quanto a natureza do equipamento de aquisição, podendo ser de qualquer modelo, mas manteve-se como padrão um tamanho de 4,7MP ( $2500 \times 1875$  ou  $1875 \times 2500$ ), com o intuito de garantir um nível aceitável de qualidade para as imagens.

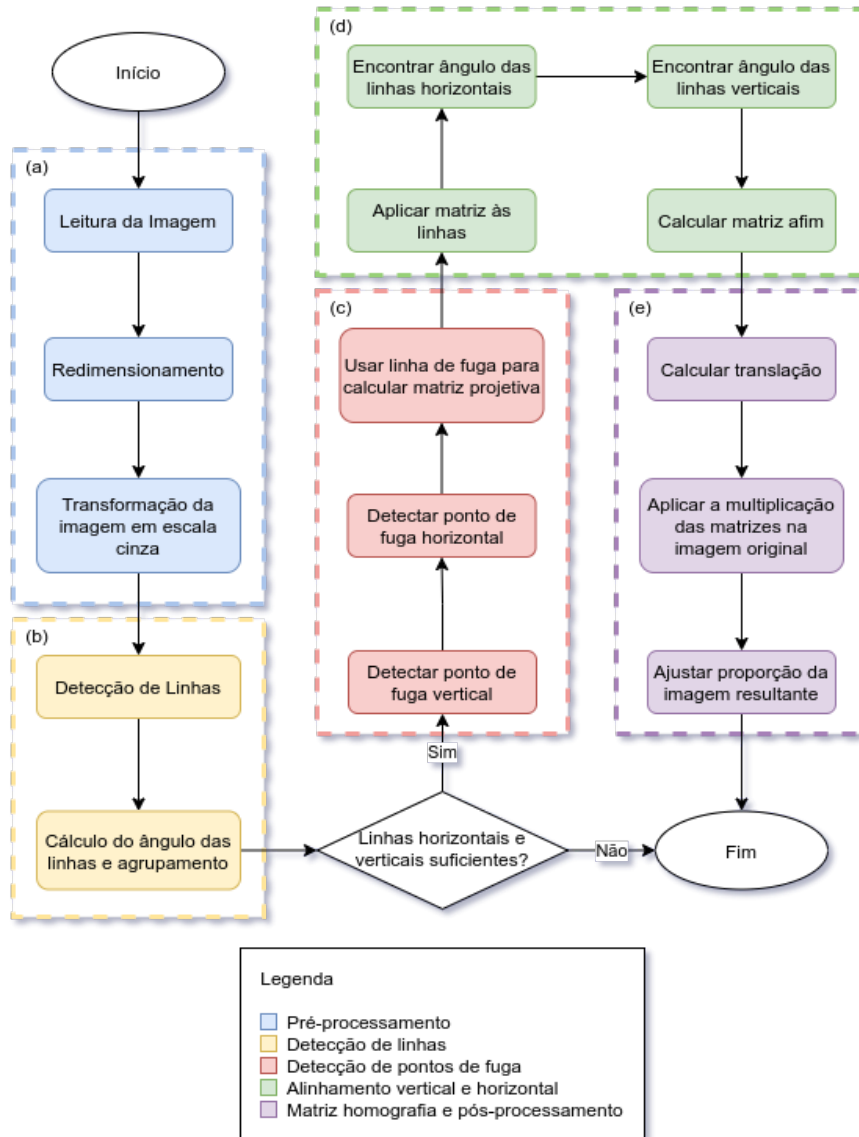
Uma mesma imagem, independente da sua escala, pode-se inferir que contém o mesmo fator de distorção de perspectiva em termos de angulação durante a aquisição. Assim sendo, optou-se por reduzir a resolução da foto que está sendo processada, deixando-a com tamanho 1250 de largura e 938 de altura, ou vice-versa, dependendo da sua proporção de entrada. Esse redimensionamento reduz a quantidade de dados em memória e conseqüentemente, também diminui o número de informações analisadas, impactando consideravelmente na performance do algoritmo nas etapas posteriores.

Na próxima etapa do pré-processamento, realiza-se uma transformação no espaço de cores da foto levando-a para tons de cinza, para efeitos de redução de dimensionalidade e simplificação da imagem sem perdas do seu contexto. O resultado então é representado somente por níveis de luminosidade, normalizada através da fórmula  $y = 0,299r + 0,587g + 0,144b$ , onde  $y$  é o valor resultante e o conjunto  $(r, g, b)$  são os níveis de vermelho, verde e azul do pixel na imagem, respectivamente.

A suavização é um processo que geralmente é aplicado em etapas do pré-processamento com o objetivo de remover pequenos detalhes na imagem, como ruídos, reduzindo erros de conexões de pequenas linhas ou curvas ([GONZALEZ; WOODS, 2010](#)). Para isso, é utilizado um filtro de suavização, representado por uma matriz que geralmente é quadrática. Ao aplicar um filtro linear de suavização, cada pixel é substituído pela média da intensidade de sua vizinhança, resultando em uma imagem com uma leve perda de nitidez, reduzindo intensidades irrelevantes. Há várias funções que podem ser utilizadas como filtro, mas as mais populares são: suavização por média aritmética espacial, suavização por mediana espacial e suavização Gaussiana.

Para finalizar essa primeira etapa foi aplicado uma suavização através de uma matriz quadrática de tamanho  $3 \times 3$ , com uma função de média aritmética espacial simples

Figura 11 – Fluxo geral da abordagem proposta para o realizar o ajuste de perspectiva em imagens de gôndolas de supermercado. O algoritmo foi sumarizado em cinco etapas: (a) aquisição e pré-processamento, (b) detecção de linhas, (c) detecção de pontos de fuga, (d) alinhamento vertical e horizontal e (e) cálculo da matriz homografia e pós-processamento.



Fonte: Próprio autor

como filtro. Essas especificações foram escolhidas dentre as demais por terem sido as que melhor performaram em testes empíricos realizados, considerando distintas configurações de filtragens. Esse processo foi realizado com o intuito de reduzir a quantidade de ruídos e atenuar as bordas de contorno, deixando somente as informações mais proeminentes na imagem.

## 4.2 Detecção de Linhas e Seleção do Algoritmo

### 4.2.1 Algoritmos de Detecção de Linhas

Foram testados três métodos disponíveis na biblioteca *OpenCV* para realizar a detecção de linhas: Transformada de Hough, Transformada de Hough Probabilística e o *Fast Line Detector*. A versão probabilística da Transformada de Hough é uma otimização feita pela biblioteca utilizada, fornecendo resultados em um tempo consideravelmente menor que a sua versão normal. Já o terceiro método é uma implementação do algoritmo apresentado por [Lee et al. \(2014\)](#).

A implementação do *OpenCV* para a Transformada de Hough retorna linhas no formato polar  $(\rho, \alpha)$ , onde  $\rho$  representa o ponto inicial da reta a partir da origem e  $\alpha$  representa o ângulo da linha em radianos, não informando o tamanho da mesma. Já a sua versão probabilística retorna como resultado uma lista de vetores com quatro coordenadas, para fins de posicionamento. As duas primeiras representam a localização do ponto inicial da reta e as próximas duas, o seu ponto final.

Ambos os métodos supracitados necessitam como entrada uma imagem binarizada. Para isso, comumente é utilizada a saída do método Canny, que já realiza a detecção das bordas e remoção de informações irrelevantes para a detecção de linhas. No algoritmo desenvolvido, optou-se também por utilizar essa técnica e para escolher os parâmetros  $T_1$  e  $T_2$  do Canny, foram utilizados valores referentes a 66% e 133% da mediana da intensidade dos pixels da imagem em escala cinza.

O método *Fast Line Detector* da biblioteca *OpenCV* também retorna uma lista de vetores com quatro coordenadas, referente aos pontos iniciais e finais das retas encontradas. Entretanto, diferentemente dos outros dois testados, não é necessário binarizar a imagem para esse algoritmo, uma vez que ele utiliza o Canny internamente, necessitando apenas passar os seus parâmetros. Novamente, foram utilizados valores proporcionais à mediana para essas propriedades.

Após utilizar o Canny, o método *Fast Line Detector* inicia a partir de um pixel de borda e iterativamente forma um segmento de reta através da conexão do mesmo com seus vizinhos, caso haja colinearidade. Uma vez que o algoritmo encontra uma curvatura acentuada, é retornado o segmento formado se o mesmo possuir uma quantidade de pixels maior que um valor definido por parâmetro. Além disso, é realizada uma etapa de pós-processamento das linhas encontradas, fundindo aquelas que estiverem sobrepostas ou muito próximas e tiverem uma diferença de orientação consideravelmente pequena ([LEE et al., 2014](#)).

### 4.2.2 Definição do Ambiente Experimental de Testes para Algoritmos de Detecção de Linhas

Foi realizado um ensaio experimental com os algoritmos acima mencionados para auxiliar na escolha do algoritmo de detecção de linhas que obteve melhor performance para o contexto apresentado. Uma vez que todos os métodos necessitavam de diversos parâmetros para serem configurados, foi desenvolvido um ambiente para testar as várias combinações de parâmetros dentro de um intervalo predefinido. As linhas encontradas eram desenhadas na imagem e então salvas em um diretório para uma escolha posterior dos melhores valores para o algoritmo. Para realizar esse teste, foi escolhida uma imagem aleatória dentre 10 selecionadas, mostrada na [Figura 12](#).

Como já comentado na seção anterior, a implementação do *OpenCV* para a

Transformada de Hough não retorna o tamanho da linha, somente seu ponto inicial e ângulo. Com isso, houve uma dificuldade na avaliação da performance desse algoritmo, uma vez que a reta poderia possuir o tamanho total da imagem, gerando resultados onde não era possível identificar a foto original. Já a sua versão probabilística desempenhou melhor, mas percebeu-se que era muito sensível à variações nos parâmetros escolhidos. O terceiro e último método era mais constante aos diversos valores utilizados, no entanto, produzia uma quantidade relativamente menor de retas.

Após escolher o valor dos parâmetros que melhor se encaixavam em cada método, foi realizada uma inspeção visual e o algoritmo que produziu os resultados mais precisos e estáveis foi o *Fast Line Detector*, implementado pelo *OpenCV*, como visto na [Figura 12](#) em comparação. Apesar de ser em menor quantidade, as linhas encontradas por esse método eram mais relevantes, com um bom encaixe aos padrões esperados e com um tamanho de acordo com o apresentado na imagem, não sendo tão fragmentadas como as linhas apresentadas pela Transformada de Hough Probabilística. Além disso, o algoritmo encontrou tanto retas verticais quanto horizontais, distribuídas em uma proporção adequada.

Após a identificação das linhas na imagem, foi calculado o ângulo  $\theta$  das mesmas em relação ao eixo horizontal negativo (da direita para a esquerda) e com sentido horário, de modo que  $0^\circ \leq \theta \leq 180^\circ$ . Com o intuito de classificar as linhas, foi definido uma margem de liberdade de  $15^\circ$  para o módulo da diferença do ângulo da linha com os eixos horizontais ou verticais. Para isso, realizou-se a seguinte distinção: se  $0^\circ \leq \theta \leq 15^\circ$  ou  $165^\circ \leq \theta \leq 180^\circ$ , então a reta era classificada como horizontal; se  $75^\circ \leq \theta \leq 105^\circ$ , classificou-se como reta vertical; caso contrário, a linha era ignorada.

### 4.3 Detecção de Pontos de Fuga

Antes de iniciar a próxima etapa do algoritmo, foi feita uma validação acerca da quantidade de linhas encontradas. Caso o estágio anterior retornar menos de duas retas verticais e duas horizontais, o programa é interrompido e nenhum ajuste na imagem é possível de ser realizado pela limitação de informações. Essa condição deve ser satisfeita devido ao fato de que necessita-se de pelo menos duas linhas para encontrar um ponto de fuga.

Com ao menos duas retas de cada orientação, foi utilizado o método RANSAC para encontrar os pontos de fuga vertical e horizontal. Com o objetivo de aumentar as chances de encontrar retas mais relevantes e melhorar o desempenho, o algoritmo inicia com a ordenação das retas por tamanho e seleção das 50% maiores. Com essa nova lista formada, o processo iterativo inicia selecionando aleatoriamente duas linhas e caso as mesmas possuam ângulos muito parecidos, é realizada uma nova tentativa. Então é calculado o ponto de encontro para ser considerado como modelo hipotético.

Para encontrar o erro do modelo de hipótese para com o resto das linhas, utilizou-se uma fórmula que era proporcional ao ângulo formado pela reta em questão e pela reta que liga o seu ponto médio e o ponto de fuga, como utilizado por [Nieto e Salgado \(2011\)](#) e mostrado na [Figura 13](#). O processo então reinicia selecionando novamente duas retas e calculando o modelo de hipótese. Aquele que tiver o menor erro é considerado o melhor ponto de fuga dentre os demais testados.

Como critério de parada do RANSAC, foi definido um número inicial de 500 iterações. Entretanto, ao longo do desenvolvimento do algoritmo percebeu-se que o melhor modelo comumente era encontrado antes da iteração 200. Testes exaustivos com 100 imagens



Figura 12 – Ambiente experimental para teste exaustivo de parâmetros de detectores de linhas. As retas em verde foram classificadas como horizontais e as azuis, verticais. As linhas ignoradas foram removidas da visualização para facilitar a inspeção.



(a) Imagem Original.



(b) Transformada de Hough.



(c) Transformada de Hough Probabilística.



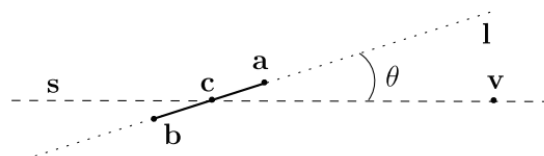
(d) *Fast Line Detector*.

**Fonte:** Próprio autor

confirmaram essa hipótese e então o número máximo escolhido foi de 200 repetições.

Após o encontro dos pontos de fuga vertical e horizontal, foi feita uma validação relativa à sua localização. Caso o ponto encontrado estivesse dentro dos limites da imagem, a correção de perspectiva resultante geraria muitas distorções visuais, já que a foto seria expandida para tornar as linhas paralelas. Nesse caso, nenhum ajuste é feito na imagem.

Figura 13 – Para o cálculo da distância da reta  $ab$  em relação ao ponto de fuga  $v$  utilizou-se uma função proporcional ao ângulo  $\theta$ .



**Fonte:** Retirado de Nieto e Salgado (2011)



Com essa etapa concluída, utilizou-se a matriz  $H_P$ , mostrada na Subseção 3.2.5 para tornar as retas concorrentes em paralelas. Um exemplo do efeito causado na foto é mostrado na segunda imagem da Figura 14.

#### 4.4 Alinhamento Horizontal e Vertical

Através da matriz de transformação puramente projetiva, foi recuperado o paralelismo das retas presentes na imagem. Uma vez que as próximas etapas da solução proposta continuam utilizando as linhas como insumo, foi necessário calcular a nova localização das mesmas, aplicando a matriz encontrada aos seus pontos iniciais e finais.

Liebowitz (2001) expôs que a matriz  $H_A$  pode ser decomposta em outras três matrizes, caso a informação disponível seja um par de direções ortogonais. A decomposição é mostrada na Equação 25.

$$H_A = A_2 A_1 R_A = \begin{bmatrix} \mu & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\cot \theta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (25)$$

A primeira matriz  $R_A$  rotaciona a imagem para alinhá-la com o eixo horizontal, sendo  $\phi$  a diferença entre o ângulo das linhas horizontais  $L_H$  e o eixo horizontal. A matriz  $A_1$  é utilizada para alinhar as retas verticais  $L_V$  com o eixo vertical, sem alterar a transformação feita anteriormente. Para isso, é utilizado o ângulo  $\theta$  formado entre  $L_V$  e  $L_H$ , considerando que as linhas verticais formam um ângulo  $\pi - \theta$  com o eixo vertical. Já a última matriz  $A_2$  ajusta a proporção da foto sem influenciar as outras transformações e  $\mu$  é um parâmetro que pode ser determinado através de medições ou através dos parâmetros internos da câmera que capturou a imagem.

Para obter o ângulo que deve ser ajustado em cada uma das etapas, optou-se por utilizar a média dos ângulos ponderados pelo tamanho das retas. Primeiramente foi encontrado o ângulo proeminente das linhas horizontais e utilizado a matriz  $R_A$  apresentada anteriormente, com a rotação referente à diferença com o eixo horizontal. O resultado dessa operação pode ser conferido na terceira imagem da Figura 14. O mesmo foi efetuado com o eixo vertical com a matriz  $A_1$ , com efeito mostrado na figura seguinte.

Os parâmetros internos de uma câmera são divididos em intrínsecos e extrínsecos. O primeiro corresponde às características do modelo da câmera, como distância focal, centro real da imagem, dimensão dos pixels e distorções da lente, já o segundo descreve a posição da câmera em relação à uma referência no ambiente real, como ângulo, distância e posição espacial (LAUREANO, 2013). Através do conhecimento desses parâmetros é possível modelar o processo de formação de imagens em uma câmera, facilitando assim a obtenção de métricas a partir da foto capturada.

Uma vez que a solução desenvolvida tem como proposta principal ser totalmente automática e sem nenhuma informação de entrada, como os parâmetros internos da câmera, optou-se por não realizar a etapa de ajuste da proporção da imagem.

#### 4.5 Construção da Matriz Homografia e Pós-Processamento

Muitas vezes o resultado obtido pelo ajuste na perspectiva ultrapassa o tamanho esperado da imagem, necessitando realizar uma translação para não perder informações.

Figura 14 – Efeito causado em uma imagem exemplo por cada uma das etapas do algoritmo desenvolvido. Em todos os exemplos foi aplicado uma matriz de translação para melhor visualização dos resultados.



(a) Imagem Original.



(b) Retificação das linhas concorrentes.



(c) Alinhamento com o eixo horizontal.



(d) Alinhamento com o eixo vertical.

**Fonte:** Próprio autor

Esse movimento é feito calculando a nova localização dos quatro cantos da imagem e caso ultrapassem o tamanho esperado, utiliza-se essa diferença em uma transformação de similaridade.

Após obter a matriz puramente projetiva, de alinhamento com o eixo horizontal e vertical e a matriz de translação, basta multiplicá-las para encontrarmos a matriz homografia, aplicando-a na imagem original para alcançar o resultado final.

Na Subseção 4.4, foi comentado sobre a impossibilidade de retificar a proporção da foto. Como alternativa, optou-se por manter a mesma proporção da imagem original, onde experimentos comprovaram que essa alteração resulta em uma foto mais agradável.

## 5 Resultados Experimentais

### 5.1 Ambiente Experimental

O *dataset* para testes e ajuste de parâmetros do algoritmo desenvolvido foi composto por 100 fotos selecionadas a partir de um conjunto de imagens capturadas por promotores de venda em supermercados em situações rotineiras do mundo real dos profissionais da

área. As mesmas foram escolhidas porque possuíam algum tipo de angulação em relação ao plano frontal da gôndola. Entretanto, vale ressaltar que nesse *dataset* não foram incluídas imagens que possuíam uma inclinação muito acentuada, de forma que informações de alguns produtos não pudessem ser reconhecidas.

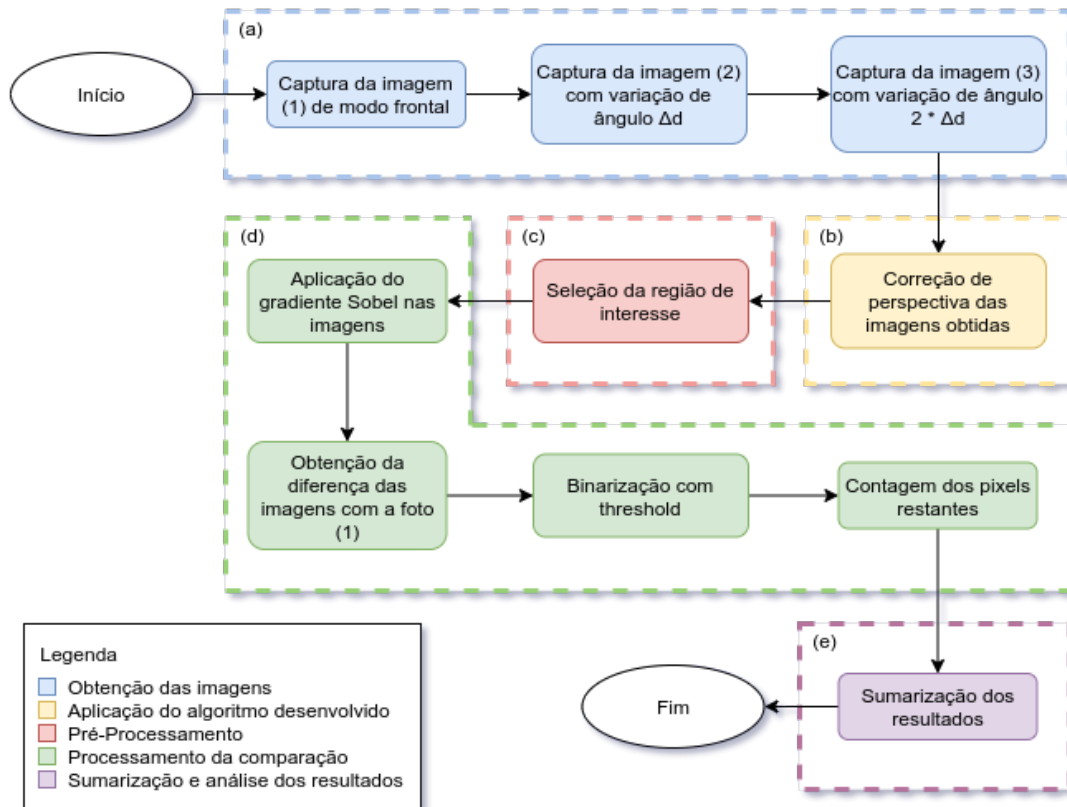
Para aferir a performance do algoritmo, foi cronometrado o tempo gasto para realizar o ajuste de cada uma das 100 imagens do conjunto descrito anteriormente. Esse teste foi realizado três vezes em diferentes dias da semana, para minimizar possíveis interferências decorrentes de outros processos que poderiam estar sendo executados na máquina utilizada.

Esse conjunto de imagens foi empregado para análise da eficácia dos parâmetros utilizados e da solução proposta quando submetidos à diversas situações, mas mantendo a restrição de entrada relacionada ao tamanho espacial de  $2500 \times 1875$  ou  $1875 \times 2500$ . Entretanto, para uma análise da eficiência geral em termos de análise quantitativa da precisão do algoritmo desenvolvido, foi necessária a construção de um novo *dataset* contendo fotos em conformidade à um protocolo mais restrito e obtidas em um ambiente controlado.

Através da anuência do gerente de um supermercado local, obteve-se fotos de 10 gôndolas e, para cada uma delas, foram seguidos os passos demonstrados na [Figura 15](#). O fluxo pode ser resumido em cinco etapas, sendo elas:

- (a) Obtenção das imagens: considerando apenas um módulo de gôndola, foram capturadas três fotos. A primeira delas foi obtida frontalmente ao plano da prateleira e sempre buscando um posicionamento perpendicular à gôndola, sendo considerada como resultado ideal; para capturar a segunda imagem, deu-se um passo para à esquerda, de aproximadamente 60 centímetros e focou-se no ponto central da imagem anterior; para a última foto, novamente movimentou-se aproximadamente a mesma distância à esquerda, direcionando a câmera para o mesmo ponto da primeira imagem capturada;
- (b) Aplicação do algoritmo desenvolvido: em cada uma das imagens obtidas no passo anterior, foi aplicado o algoritmo de correção de perspectiva apresentado nesse trabalho, resultando em um total de seis imagens;
- (c) Pré-processamento: foi selecionada uma região de interesse contendo uma quantidade de produtos presentes na gôndola. Para as fotos anguladas, foi selecionada a região a qual todas as frentes dos produtos selecionados estivessem presentes. Esse padrão foi mantido para as seis imagens;
- (d) Processamento da comparação: para cada uma das imagens, foi aplicado o gradiente Sobel. A diferença entre as fotos foi aplicada sempre em relação à primeira imagem obtida, com ângulo frontal da gôndola. Assim sendo, foi feita a subtração do recorte da segunda imagem obtida e após o recorte da sua respectiva imagem à qual aplicou-se o ajuste de perspectiva. O mesmo foi feito com a terceira foto. Com o resultado da diferença, realizou-se uma binarização onde todos pixels com valores acima do valor de intensidade igual a 100 eram convertidos em 255, caso contrário, se tornavam 0. Após esse procedimento, foi contado quantos pixels permaneceram brancos na imagem final;
- (e) Sumarização e análise dos resultados: como etapa final, foi construída uma tabela que resume os dados obtidos, sendo discutida na próxima subseção.

Figura 15 – Fluxo utilizado como protocolo para a validação do algoritmo proposto. O mesmo possui cinco etapas: (a) obtenção das fotos em um supermercado; (b) uso do sistema computacional desenvolvido; (c) pré-processamento das imagens; (d) processamento das imagens de comparação; (e) sumarização e análise das métricas obtidas.



Fonte: Próprio autor

Essa técnica de comparação foi escolhida por possibilitar analisar a diferença a nível de produto que o efeito de ajuste de perspectiva produz em uma imagem angulada. O algoritmo Sobel produz resultados de gradientes que sumarizam as características principais da imagem analisada, sendo possível utilizar somente as partes mais importantes da mesma para análise. Já a binarização da imagem resultante da subtração, com valor máximo de 100, têm como objetivo remover possíveis desalinhamentos fracos decorrentes da perspectiva dos produtos, uma vez que fotos obtidas a partir de ângulos distintos obtêm visões diferentes dos produtos em questão. Esse procedimento possibilitou evidenciar as diferenças mais sobressalentes entre as duas fotos.

O hardware do dispositivo utilizado para realizar todos os testes possui um processador Intel® Core™ i7-8550U com 4 núcleos físicos, 8MB de memória cache e frequência de 1.80GHz, além de 16GB de memória RAM e Disco Rígido de 1TB. Apesar de não utilizado, a máquina também possui um processador gráfico integrado Intel® UHD Graphics 620.

## 5.2 Validação Qualitativa e Quantitativa

Utilizando o *dataset* construído para testes e aperfeiçoamento de parâmetros, foi possível constatar que a média de tempo necessário para realizar o ajuste de perspectiva ficou entre 2,5 segundos, com desvio padrão de 1,12 segundo. Através dessa análise percebeu-se que o tempo gasto é proporcional ao tamanho da imagem resultante, uma vez que imagens com resolução espacial maior eram provenientes de execuções mais demoradas. Isso pode ser decorrente do processo de escrita em disco ser lento em máquinas com disco rígido convencional.

No conjunto de dados utilizado, foi constatado através de uma análise visual um alinhamento adequado em 94% das imagens. As demais fotos ficaram distorcidas a ponto de impossibilitar a sua utilização, ou por deformar o formato real da gôndola ou por ocultar informações importantes sobre os produtos presentes.

A [Figura 16](#) demonstra a capacidade do método proposto, onde é possível verificar o alinhamento de algumas fotos pertencentes ao *dataset* de validação, juntamente com as suas imagens originais. Realizando uma inspeção de forma visual, pode-se concluir que os resultados são satisfatórios, deixando as linhas das prateleiras paralelas e alinhadas ao eixo horizontal, além de manter um padrão mesmo quando submetido à diversos tipos de exposição de produtos.

No entanto, apesar da análise visual demonstrar acuidade no alinhamento automatizado dos produtos da gôndola, uma análise quantitativa preliminar é necessária. Para tanto, foi seguido os procedimentos mostrados na [Figura 15](#) e descritos na seção anterior, para validar o alinhamento com base na informação de gradiente.

Através desse método, foi possível verificar o erro relacionado a diferença de pixels entre as imagens de cada gôndola e a foto considerada como ideal, obtida de modo frontal e perpendicular ao plano da gôndola. Os dados obtidos estão apresentados na [Tabela 1](#), onde para cada gôndola e ângulo analisado, é mostrada a proporção de pixels brancos na imagem em relação ao tamanho total da mesma, antes e depois de realizar o ajuste de perspectiva. Sendo assim, quanto mais próximo a 1, mais diferenças a imagem possui se comparada à sua foto obtida perpendicularmente à gôndola.

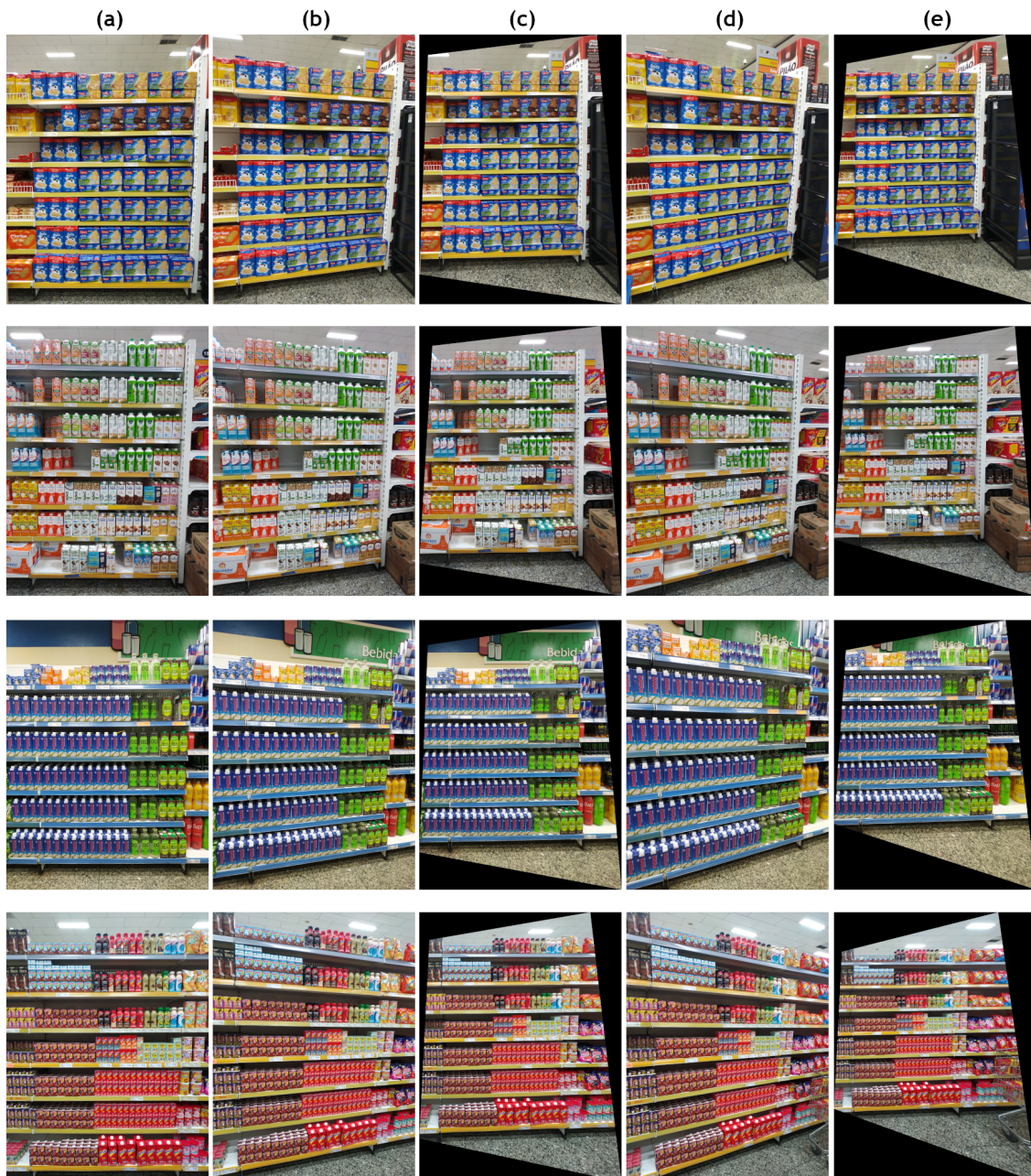
Pode-se perceber que em todos os casos, o erro diminuiu após o ajuste de perspectiva, indicando que através da utilização do sistema computacional proposto neste trabalho, foi possível obter uma foto que está mais próxima da imagem ideal em todos os casos amostrados.

Um ponto importante a ser destacado é que a imagem considerada ideal, apesar de aparentar estar frontal ao plano da gôndola, ainda possuía alguma inclinação. Isso se deve ao fato de que não foi utilizado nenhum mecanismo mecânico e não foi feito nenhum cálculo para posicionar a câmera no local desejado, ocorrendo de forma totalmente manual e empírica. Esse fato pode ter impactado na métrica construída para validação, uma vez que a comparação não está sendo em relação à uma imagem perfeitamente ideal.

Na [Figura 17](#) é apresentado um exemplo das imagens resultantes do padrão utilizado para recorte da área de interesse e a [Figura 18](#) demonstra o resultado do protocolo empregado na validação, confirmando que o erro diminuiu após a realização do ajuste de perspectiva.



Figura 16 – Avaliação visual qualitativa dos resultados obtidos pela abordagem computacional proposta. Na primeira coluna em (a) a imagem considerada ideal, capturada de forma frontal, é apresentada. Na segunda (b) e terceira (c) colunas são demonstradas a imagem com 01 deslocamento a esquerda e a correção de perspectiva, respectivamente. Similarmente, na quarta e quinta colunas (d) e (e) a imagem com 02 deslocamentos e a respectiva correção de perspectiva são demonstradas.



Fonte: Próprio autor

Tabela 1 – Análise do erro da diferença entre os gradientes das fotos anguladas e a foto obtida de modo frontal à gôndola, antes e depois do ajuste de perspectiva, relativo à quantidade total de pixels. Para cada gôndola, foram capturadas duas imagens a partir de posições diferentes, onde a segunda possuía um ângulo maior em relação ao plano frontal.

Gôndola	Angulação da Foto	Antes do Ajuste de Perspectiva	Depois do Ajuste de Perspectiva
1	Caso 1	0,24	0,21
1	Caso 2	0,24	0,21
2	Caso 1	0,20	0,18
2	Caso 2	0,22	0,18
3	Caso 1	0,13	0,09
3	Caso 2	0,18	0,12
4	Caso 1	0,14	0,10
4	Caso 2	0,16	0,09
5	Caso 1	0,15	0,10
5	Caso 2	0,16	0,11
6	Caso 1	0,15	0,11
6	Caso 2	0,17	0,12
7	Caso 1	0,24	0,17
7	Caso 2	0,25	0,19
8	Caso 1	0,19	0,10
8	Caso 2	0,20	0,13
9	Caso 1	0,24	0,16
9	Caso 2	0,26	0,18
10	Caso 1	0,16	0,12
10	Caso 2	0,16	0,14

Figura 17 – Exemplo de recorte da área de interesse realizado para a etapa de validação da gôndola 8 da Tabela 1. Foi mantida a coerência de apresentar sempre os mesmos produtos nos recortes realizados.



(a) Foto capturada frontalmente ao plano da gôndola.



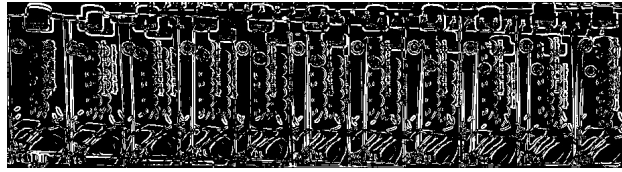
(b) Foto obtida a partir de um ângulo não frontal



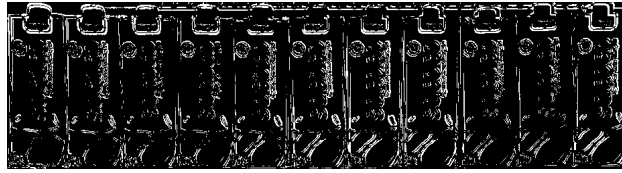
(c) Resultado do ajuste de perspectiva de (b).

Fonte: Próprio autor

Figura 18 – Diferença entre a binarização das imagens apresentadas na Figura 17. Pode-se perceber que quando utilizada a foto com a perspectiva ajustada, a diferença é consideravelmente menor.



(a) Diferença entre a Figura 17a e Figura 17b.



(b) Diferença entre a Figura 17a e Figura 17c.

**Fonte:** Próprio autor

## 6 Conclusões e Trabalhos Futuros

O presente trabalho apresentou as variáveis e características que contemplavam as etapas de desenvolvimento de um algoritmo que permite realizar um ajuste na perspectiva de imagens obtidas em supermercados. Através de um *dataset* de testes, foi possível comprovar um alinhamento adequado de 94% das fotos da solução computacional proposta, e com um conjunto de fotos de validação, obteve-se uma melhoria na semelhança da imagem com a imagem considerada ideal em todos os casos testados. Com um baixo custo computacional, o tempo médio despendido para a execução ficou em 2,5 segundos, onde a solução desenvolvida se mostrou viável para o contexto apresentado.

Para tal, foi utilizado o método baseado em pontos de fuga, encontrados através do algoritmo RANSAC. Para a etapa de validação qualitativa e quantitativa, utilizou-se fotos obtidas em um ambiente controlado, realizando uma análise visual e uma comparação através da diferença de gradiente das fotos, respectivamente.

Com vista na evolução do trabalho apresentado, o algoritmo poderia ser incrementado com uma etapa de pós-processamento da imagem, realizando um corte automático da região de interesse, para aumentar a área útil da mesma. Essa etapa facilitaria a identificação dos produtos e enquadramento da gôndola na foto, reduzindo assim as áreas pretas inseridas devido ao efeito de perspectiva.

Uma outra possibilidade seria incrementar a robustez do algoritmo. Isso poderia ser realizado adicionando uma etapa de validação automática da foto ao final do processo e, caso a mesma não possuir uma qualidade mínima, o algoritmo poderia ser reiniciado com algumas alterações nos parâmetros. Essa etapa de avaliação da imagem resultante também poderia ser útil para invalidar casos onde a matriz homografia inseriu muita perspectiva na foto, inviabilizando a utilização da mesma.



## Referências

- ALMEIDA, V. M. C. D. et al. Trade marketing no setor de lojas de conveniência. *Revista de Administração de Empresas*, v. 52, n. 6, p. 643–656, 2012. ISSN 0034-7590. Citado na página 3.
- ARBACHE, F. S. et al. *Gestão de logística, distribuição e trade marketing*. Editora FGV. Rio de Janeiro: Publicações FGV Management, 2011. ISBN 9788522511549. Citado na página 3.
- ÁVILA, B. T.; LINS, R. D. A fast orientation and skew detection algorithm for monochromatic document images. *Proceedings of the 2005 ACM Symposium on Document Engineering*, p. 118–126, 2005. Citado na página 3.
- BARNARD, S. T. Interpreting perspective images. *Artificial Intelligence*, v. 21, n. 4, p. 435–462, 1983. ISSN 00043702. Citado na página 18.
- BATISTA, L. d. S. *A Importância Da Fotografia Como Fonte De Informação No Âmbito Do TRT Da Paraíba*. 203 f. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal da Paraíba, João Pessoa, 2014. Citado na página 3.
- BOTTERILL, T.; MILLS, S.; GREEN, R. Fast RANSAC hypothesis generation for essential matrix estimation. *Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011*, p. 561–566, 2011. Citado na página 18.
- CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, 1986. ISSN 01628828. Citado 2 vezes nas páginas 13 e 15.
- CANTONI, V. et al. Vanishing point detection: Representation analysis and new approaches. *Proceedings - 11th International Conference on Image Analysis and Processing, ICIAP 2001*, n. Iciap, p. 90–94, 2001. Citado na página 19.
- CHAUDHURY, K.; DIVERDI, S.; IOFFE, S. Auto-Rectification Of User Photos. *International Conference on Image Processing(ICIP)*, p. 3479–3483, 2014. Citado 2 vezes nas páginas 4 e 17.
- CHEN, X. et al. A new vanishing point detection algorithm based on Hough transform. *3rd International Joint Conference on Computational Sciences and Optimization, CSO 2010: Theoretical Development and Engineering Practice*, v. 2, n. 1, p. 440–443, 2010. Citado na página 17.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, v. 24, n. 6, p. 381–395, 1981. ISSN 15577317. Citado na página 18.
- FOLEY, J. D. et al. *Computer Graphics: Principles and Practice*. [S.l.]: Addison-Wesley, 1996. 1264 p. ISBN 0201848406. Citado na página 7.

- GOLPARDAZ, M.; NEZAMABADI-POUR, H. Perspective rectification and skew correction in camera-based Farsi document images. *2011 7th Iranian Conference on Machine Vision and Image Processing, MVIP 2011 - Proceedings*, p. 1–5, 2011. Citado na página 6.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. São Paulo: Pearson Education do Brasil, 2010. 624 p. Citado 7 vezes nas páginas 12, 13, 14, 15, 16, 17 e 20.
- HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2004. 655 p. Citado 5 vezes nas páginas 6, 7, 8, 9 e 10.
- HAST, A.; NYSJÖ, J.; MARCHETTI, A. Optimal RANSAC - Towards a repeatable algorithm for finding the optimal set. *Journal of WSCG*, v. 21, n. 1, p. 21–30, 2013. ISSN 12136972. Citado na página 19.
- HOUGH, P. V. C. Method and means for recognizing complex patterns. *U.S. Patent, no. 3069654*, 1962. Disponível em: <<https://patents.google.com/patent/US3069654A/en>>. Citado na página 16.
- JANICKA, J.; RAPINSKI, J. Outliers Detection By Ransac Algorithm in the Transformation of 2D Coordinate Frames. *Boletim de Ciências Geodésicas*, v. 20, n. 3, p. 610–625, 2014. ISSN 1413-4853. Citado na página 18.
- KIRAN, G.; MURALI, S. Automatic Rectification of Perspective Distortion from a Single Image Using Plane Homography. *International Journal on Computational Science & Applications*, v. 3, n. 5, p. 47–58, 2013. ISSN 2200-0011. Citado na página 5.
- KOTLER, P.; KELLER, K. L. *Administração de Marketing*. São Paulo: Prentice Hall, 2013. v. 14 edição. 767 p. ISBN 978-85-8143-000-3. Citado na página 3.
- LACERDA, A. L. de. A fotografia nos arquivos: produção e sentido de documentos visuais. *História, Ciências, Saúde*, v. 19, n. 1, p. 283–302, 2012. Citado na página 3.
- LAUREANO, G. T. *Deteção Topológica de Padrões Xadrez para Calibração de Câmeras*. 140 p. Tese (Tese de Doutorado) — Universidade de São Paulo, 2013. Citado na página 25.
- LEE, H. et al. Automatic upright adjustment of photographs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 877–884, 2012. ISSN 10636919. Citado na página 3.
- LEE, J. H. et al. Outdoor place recognition in urban environments using straight lines. *Proceedings - IEEE International Conference on Robotics and Automation*, p. 5550–5557, 2014. ISSN 10504729. Citado na página 22.
- LI, X. et al. Method for rectifying image deviation based on perspective transformation. *IOP Conference Series: Materials Science and Engineering*, v. 231, n. 1, 2017. ISSN 1757899X. Citado na página 5.
- LI, Y. et al. Perspective Plane Program Induction from a Single Image. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 4433–4442, 2020. ISSN 10636919. Citado na página 6.

- LIEBOWITZ, D. *Camera Calibration and Reconstruction of Geometry from Images*. 209 p. Tese (Doutorado) — University of Oxford, 2001. Disponível em: <<https://www.robots.ox.ac.uk/~vgg/publications/2001/Liebowitz01/liebowitz01.pdf>>. Citado 5 vezes nas páginas 6, 9, 10, 11 e 25.
- LIEBOWITZ, D.; ZISSERMAN, A. Metric rectification for perspective images of planes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 482–488, 1998. ISSN 10636919. Citado 2 vezes nas páginas 10 e 11.
- LU, S.; CHEN, B. M.; KO, C. C. Perspective rectification of document images using fuzzy set and morphological operations. *Image and Vision Computing*, v. 23, n. 5, p. 541–553, 2005. ISSN 02628856. Citado na página 6.
- MACHADO, M. d. C. *Geração De Malhas De Falhas Em Dados Sísmicos Por Aprendizado Competitivo*. 203 p. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, 2008. Citado na página 15.
- MANTEL, C. et al. Method for Estimation and Correction of Perspective Distortion of Electroluminescence Images of Photovoltaic Panels. *IEEE Journal of Photovoltaics*, v. 10, n. 6, p. 1797–1802, 2020. ISSN 21563403. Citado na página 5.
- NIELSEN. *Nielsen Series – A hora certa de ativar o shopper*. [S.l.], 2015. 1–20 p. Disponível em: <<https://www.nielsen.com/wp-content/uploads/sites/3/2019/04/NielsenSeries-Shopper-2015.pdf>>. Citado na página 3.
- NIETO, M.; SALGADO, L. Simultaneous estimation of vanishing points and their converging lines using the EM algorithm. *Pattern Recognition Letters*, Elsevier B.V., v. 32, n. 14, p. 1691–1700, 2011. ISSN 01678655. Disponível em: <<http://dx.doi.org/10.1016/j.patrec.2011.07.018>>. Citado 3 vezes nas páginas 17, 23 e 24.
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. [S.l.]: Editora Thomson Learning, 2007. 528 p. Citado 3 vezes nas páginas 12, 13 e 14.
- PENG, J.; SHI, C. Recognizing road environment for vision system on autonomous vehicles. In: *ITST 2007 - 7th International Conference on Intelligent Transport Systems Telecommunications*. [S.l.: s.n.], 2007. p. 118–123. ISBN 1424411785. Citado na página 19.
- RADHA, M.; MUTHUKRISHNAN, R. Edge Detection Techniques for Image Segmentation. *International Journal of Computer Science & Information Technology (IJCSIT)*, v. 3, n. 6, p. 259–267, 2011. ISSN 0012-3692. Citado 2 vezes nas páginas 13 e 14.
- REY, C. *Projective geometry an introduction*. Oxford: Oxford University Press, 2006. 1–211 p. Citado na página 10.
- REZENDE, P. J. de; STOLFI, J. *Fundamentos de geometria computacional*. São Paulo: Universidade Federal de Pernambuco, Departamento de Informatica, 1994. 222 p. Disponível em: <<http://books.google.com.br/books?id=Vck4GwAACAAJ>>. Citado na página 9.
- ROTHER, C. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, v. 20, n. 9-10, p. 647–655, 2002. ISSN 02628856. Citado 2 vezes nas páginas 17 e 18.

- SANTANA-CEDRÉS, D. et al. Automatic correction of perspective and optical distortions. *Computer Vision and Image Understanding*, v. 161, p. 1–10, 2017. ISSN 1090235X. Citado 2 vezes nas páginas 3 e 4.
- SHUFELT, J. A. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, n. 3, p. 282–288, 1999. ISSN 01628828. Citado na página 19.
- SONKA, M.; HLAVAC, V.; BOYLE, R. *Image processing, Analysis, and Machine Vision*. Stamford: Cengage Learning, 2014. 930 p. ISBN 9781133593607. Citado na página 17.
- SOYCAN, A.; SOYCAN, M. Perspective correction of building facade images for architectural applications. *Engineering Science and Technology, an International Journal*, Karabuk University, v. 22, n. 3, p. 697–705, 2019. ISSN 22150986. Citado 2 vezes nas páginas 8 e 9.
- SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2. ed. Springer, 2010. 979 p. ISBN 978-1-84882-934-3. Disponível em: <<http://szeliski.org/Book/>>. Citado 5 vezes nas páginas 9, 12, 15, 16 e 17.
- TAKEZAWA, Y.; HASEGAWA, M.; TABBONE, S. Camera-captured document image perspective distortion correction using vanishing point detection based on Radon transform. *Proceedings - International Conference on Pattern Recognition*, v. 0, n. 1, p. 3968–3974, 2016. ISSN 10514651. Citado na página 3.
- TSIRONIS, V. et al. Automatic rectification of building façades. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, v. 42, n. 2W3, p. 645–650, 2017. ISSN 16821750. Citado na página 3.
- WINTERLE, P. *Vetores e Geometria Analítica*. São Paulo: Pearson Education do Brasil, 2000. 249 p. ISBN 8534611092. Citado na página 7.
- YANG, W.; FANG, B.; TANG, Y. Y. Fast and Accurate Vanishing Point Detection and Its Application in Inverse Perspective Mapping of Structured Road. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 48, n. 5, p. 755–766, 2018. ISSN 21682232. Citado na página 5.
- YIN, X. C. et al. Robust vanishing point detection for MobileCam-based documents. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, p. 136–140, 2011. ISSN 15205363. Citado na página 18.