

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA  
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**GUSTAVO PEDRINI MAESTRI**

**FUSÃO DE IMAGENS EM DIFERENTES PERSPECTIVAS PARA GERAÇÃO DE  
FACHADAS LIVRES DE OCLUSÃO**

**JOINVILLE  
2015**

**GUSTAVO PEDRINI MAESTRI**

**FUSÃO DE IMAGENS EM DIFERENTES PERSPECTIVAS PARA GERAÇÃO DE  
FACHADAS LIVRES DE OCLUSÃO**

Orientador: André Tavares da Silva

JOINVILLE  
2015

**GUSTAVO PEDRINI MAESTRI**

**FUSÃO DE IMAGENS EM DIFERENTES PERSPECTIVAS PARA GERAÇÃO DE  
FACHADAS LIVRES DE OCLUSÃO**

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciência da Computação como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação pela Universidade do Estado de Santa Catarina.

**Banca Examinadora**

---

Prof. Dr. André Tavares da Silva  
(Orientador)

---

Prof. Dr. Marcelo da Silva Hounsell

---

Prof. Dr. Rogério Eduardo da Silva

**Joinville, data 11/06/2015**

## RESUMO

Este trabalho objetiva resolver os problemas de privacidade existentes nos serviços de visualização de imagens panorâmicas de regiões pelo mundo, visando remover os objetos passíveis de identificação das imagens dos serviços de visualização. Inicialmente foi executado um levantamento bibliográfico para identificação das abordagens de outros trabalhos com o problema identificado. Com este estudo foram identificados problemas nos métodos de remoção de objetos, como dependência de interferência humana e tempo de execução elevado. Após a pesquisa bibliográfica, foi feito um estudo das ferramentas e métodos utilizados pelos outros autores. Por fim foi projetado um método, se baseando nos trabalhos estudados, para a remoção dos objetos em movimentos que consiste em 3 etapas: na primeira, é feita a identificação dos pontos relacionados entre as 3 imagens vizinhas com a utilização do SURF; na segunda, os pontos serão utilizados para o alinhamento das 3 imagens através de uma transformação inversa de perspectiva se utilizando do algoritmo RANSAC para a extração da matriz fundamental; por fim, é feita uma identificação de fundo com uma abordagem *per-pixel*, identificando o *pixel* de fundo a partir da distância Manhattan dos *pixels* analisados. O método objetiva gerar uma imagem livre do objeto, pois o mesmo provavelmente estará em uma única imagem dentre as três analisadas. O método criado utilizou um processo automatizado e reduziu o tempo de execução em relação aos trabalhos relacionados automatizados. Porém, o método não entregou resultados com a qualidade esperada devido a problemas na identificação de fundo bem como na obtenção de pontos pelo SURF. Por conta disto, foram levantadas ideias para o melhoramento da qualidade das imagens resultantes em trabalhos futuros.

**Palavras-chave:** Google Street View. Remoção de Objetos. Registro de Imagem.

## ABSTRACT

This work objective solve the existing problems of privacy in view services of panoramic images from regions around the world, aiming to remove the objects that can be identified of the view services images. Initially was performed a literature review to identify the approaches of other jobs on the identified problem. With this study were identified problems in the methods for removing objects, such as dependence on human interference and high runtime. After bibliographic research, a study about the tools and methods used by other authors was realized. Finally a method was designed, based on the jobs studied, for the removal of objects in movement, the method consists of three stages: first, identification of related points among the three neighboring images using the SURF; in the second, the points will be used to align the three images through a inverse perspective transformation using the RANSAC algorithm for extracting the fundamental matrix; Finally, a background identification is made with an per-pixel approach, identifying the background pixel from the images with the Manhattan distance of the pixels analyzed. The method objective to generate a image without the object, because it probably will be in a single image among the three analyzed. The method created used an automated process and reduced the runtime compared to the automated related jobs. However, the method has not given the expected results due to quality problems in background identifying and obtaining points with SURF. Because of this, ideas were raised to improve the quality of the resulting images in future work.

**Keywords:** Google Street View. Object Removal. Image Registry.

## LISTA DE FIGURAS

Figura 1	Imagen exemplo retirada do GSV.	9
Figura 2	Imagen após processamento, com pessoas removidas.	10
Figura 3	Representação tridimensional RGB.	14
Figura 4	Distância de Manhatam.	14
Figura 5	Identificação de fundo matriz bidimensional RG.	16
Figura 6	Ilustração do resultado do algoritmo RANSAC.	17
Figura 7	Ilustração de Transformação de Perspectiva.	18
Figura 8	Exemplo de Funcionamento do KNN.	20
Figura 9	Exemplo de <i>inpainting</i> .	20
Figura 10	Representação multiescala usada no SIFT.	22
Figura 11	Histograma de orientação dos gradientes.	23
Figura 12	Geração dos descritores de pontos chave.	24
Figura 13	Exemplo de funcionamento do HOG.	24
Figura 14	Filtro de aproximação utilizando convolução.	26
Figura 15	Cálculo da orientação.	26
Figura 16	Matching por contraste do fundo.	27

Figura 17 Alinhamento por Transformação de Perspectiva.	29
Figura 18 Resultado da fusão.	29
Figura 19 Processo Preenchimento dos <i>pixels</i>	30
Figura 20 Processo Preenchimento dos <i>pixels</i> .	31
Figura 21 Relação de informação redundante em duas imagens do GSV.	32
Figura 22 Resultado da Transformação de Perspectiva.	33
Figura 23 Sobreposição das Caixas Detectadas.	34
Figura 24 Vestígio da Pessoa Retirada na Imagem Resultante.	35
Figura 25 Processo completo do método proposto por Nodari, Vanetti e Gallo (2012).	37
Figura 26 Interface do Programa.	37
Figura 27 Fusão das Máscaras dos <i>frames</i> .	38
Figura 28 Processo de Junção dos <i>frames</i> .	39
Figura 29 Fluxograma do Método Proposto.	44
Figura 30 Detecção dos Pontos com SURF.	45
Figura 31 Falsos Positivos do KNN.	46
Figura 32 Falsos Positivos Filtrados.	47
Figura 33 Imagens Antes e Depois da Transformação de Perspectiva	48

Figura 34 Esquema de Comparação de Distâncias.	49
Figura 35 Prédio em Berlin.	50
Figura 36 Obra em Andamento.	51
Figura 37 Banco do Bradesco.	51
Figura 38 Supermercado Angeloni.	51
Figura 39 Eletrônica.	52
Figura 40 Transformação errada por falta de pontos.	53
Figura 41 Imagem borrada por conta do não alinhamento no método.	53

## **LISTA DE ABREVIACÕES**

GSV	Google Street View
RANSAC	<i>Random Sample Consensus</i>
KNN	<i>K Nearest Neighbors</i>
SIFT	<i>Scale Invariant Feature Transform</i>
DoG	Diferença das Gaussianas
HOG	Histograma de Gradientes Orientados
SVM	<i>Support Vector Machine</i>
SURF	<i>Speeded Up Robust Features</i>
MNOS	<i>Multi-Net for Object Segmentation</i>
GUI	<i>Graphical User Interfaces</i>
ASIFT	<i>Affine-SIFT</i>

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>9</b>
1.1 OBJETIVO GERAL .....	11
1.2 OBJETIVOS ESPECÍFICOS: .....	11
1.3 METODOLOGIA UTILIZADA .....	12
1.4 ESCOPO .....	12
1.5 ESTRUTURA .....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>13</b>
2.1 IMAGEM .....	13
2.2 MANHATAM .....	14
2.3 IDENTIFICAÇÃO DE FUNDO.....	15
2.4 <i>RANDOM SAMPLE CONSENSUS</i> .....	16
2.5 TRANSFORMAÇÃO DE PERSPECTIVA 2D .....	18
2.6 <i>K NEAREST NEIGHBORS</i> .....	19
2.7 <i>INPAINTING</i> .....	20
2.8 DETECCÃO DE DESCRIPTORES.....	21
2.8.1 <i>SCALE INVARIANT FEATURE TRANSFORM (SIFT)</i> .....	21
2.8.2 HISTOGRAMA DE GRADIENTES ORIENTADOS (HOG) .....	24
2.8.3 <i>SPEEDED UP ROBUST FEATURES (SURF)</i> .....	25
<b>3 TRABALHOS RELACIONADOS.....</b>	<b>28</b>
3.1 REMOÇÃO DE OBJETOS OCLUSORES DE FACHADAS PARA MONTAGEM DE UMA MODELAGEM 3D .....	28
3.2 REMOVENDO OBJETOS DO PRIMEIRO PLANO UTILIZANDO INFORMAÇÃO DE PROFUNDIDADE .....	30
3.3 REMOVENDO PEDESTRES DO GSV UTILIZANDO SIFT E HOG.....	31
3.4 SUBSTITUINDO PEDESTRES DO GSV UTILIZANDO HOG E <i>INPAINTING</i> .....	35
3.5 FERRAMENTA PARA A REMOÇÃO DE PEDESTRES UTILIZANDO <i>INPAINTING</i> .....	37
3.6 REMOÇÃO DE OBJETOS EM IMAGENS MULTI-VIEW POR CAMADAS UTILIZANDO SEGMENTAÇÃO DE REGIÕES .....	38
3.7 CONSIDERAÇÕES .....	40
<b>4 FUSÃO DE IMAGENS EM DIFERENTES PERSPECTIVAS PARA GERAÇÃO DE FACHADAS LIVRES DE OCLUSÃO .....</b>	<b>42</b>

4.1	OBTENÇÃO DOS PONTOS .....	45
4.2	TRANSFORMAÇÃO DE PERSPECTIVA .....	47
4.3	IDENTIFICAÇÃO DE FUNDO.....	48
<b>5</b>	<b>RESULTADOS.....</b>	<b>50</b>
5.1	DISCUSSÃO .....	52
<b>6</b>	<b>CONCLUSÕES.....</b>	<b>54</b>
6.1	DISCUSSÃO .....	55
6.2	TRABALHOS FUTUROS.....	55
	<b>REFERÊNCIAS.....</b>	<b>57</b>

## 1 INTRODUÇÃO

Desde que foi lançado em 25 de maio de 2007, o GSV (Google Street View) vem possibilitando que os usuários percorram trajetos virtuais. Estes simulam a experiência de caminhar pelos locais onde as fotos foram tiradas, possibilitando os usuários conhecerem lugares em que nunca estiveram (KOZAC, 2014). Porém esta tecnologia tem gerado problemas de privacidade, provenientes das fotos tiradas pelo sistema. Esses problemas são devidos as pessoas e veículos presentes nas imagens tiradas pelo “carro do GSV”. O carro do GSV é o dispositivo utilizado para a captura das imagens panorâmicas utilizadas pelo GSV, deixando as pessoas e os veículos expostos a identificação.

A solução proposta por muitos trabalhos, como Flores e Belongie (2010), Nodari, Vanetti e Gallo (2012), Haro (2012), é a retirada das pessoas presentes nas imagens do GSV, como é ilustrado na Figura 1 e depois na Figura 2.

**Figura 1:** Imagem exemplo retirada do GSV.



**Fonte:** Produção do próprio autor.

**Figura 2: Imagem após processamento, com pessoas removidas.**



**Fonte:** Produção do próprio autor.

Para contornar o problema de identificação e violação da privacidade, a Google implementou um algoritmo para o “borramento” da face das pessoas e das placas dos carros. Esse processo dificulta a identificação porém através de manipulação digital da imagem é possível identificar características da pessoa (LORENA, 2011). Também é possível a identificação através do cruzamento das informações deixadas intactas como local, horário, roupas e silhueta, no caso das pessoas, e cor, modelo, local e horário, no caso dos carros. Fora a identificação das pessoas, existe também o problema de publicação de eventos fortes, como pessoas nuas. Estes eventos permaneceram disponíveis globalmente até serem tiradas fotos mais atuais do local, e esse tempo varia de acordo com os locais (RAKOWER, 2011). Esse tempo de permanência das imagens vem sendo contestado recentemente, pois muitas pessoas acabaram movendo ações para garantir o seu direito ao esquecimento. No dia de 30 de maio de 2014, o direito do esquecimento foi posto em prática na Google, garantindo que o usuário seja “esquecido” pela Internet, e permitindo o usuário retirar informações referentes ao mesmo da Internet. Este direito poderá ser aplicado ao GSV, mas por

enquanto apenas será aplicado ao sistema de busca da Google. Mesmo no Brasil existem vários casos onde pessoas conseguiram o direito do esquecimento, assim garantindo a retirada de informações provenientes da internet que se remetiam a fatos acontecidos a mais de 5 anos (BUCAR, 2013).

Os vários problemas relacionados à privacidade nos serviços de visualização de imagens panorâmica, geralmente são mais explorados no GSV, um dos principais motivos é a popularidade do mesmo, esse é o principal motivo para a escolha do GSV para este trabalho. Outro motivo seria a popularidade do mesmo, e por ser a ferramenta mais utilizada, isso devido a falta de concorrentes diretos, apenas “pseudo concorrentes”, segundo Kozac (2014). Outros trabalhos que visam os mesmos resultados deste também se baseiam no GSV, como Flores e Belongie (2010) e Nodari, Vanetti e Gallo (2012).

Mesmo com o borramento da face das pessoas e placas de carros proposto pela Google, várias ações jurídicas continuam sendo movidas contra a empresa. Um modo de se manter a privacidade das pessoas é simplesmente remover por completo as pessoas e os carros das imagens do GSV, deixando apenas as informações pertinentes que se encontram nas imagens, como por exemplo as fachadas das casas e estabelecimentos comerciais. Com isso, o usuário ainda teria como visualizar os locais virtualmente sem as pessoas e carros que teriam aparecido nestas imagens, melhorando a visualização das fachadas e evitando problemas judiciais.

## 1.1 OBJETIVO GERAL

Desenvolver um método eficiente de processamento de imagem que possibilite a remoção de objetos móveis como pessoas e veículos nas imagens de sistemas de visualização de imagens panorâmicas. O comportamento e desempenho do método será analisado de acordo com as métricas de tempo de execução e qualidade do resultado, estas serão comparadas entre o método desenvolvido e os trabalhos relacionados.

## 1.2 OBJETIVOS ESPECÍFICOS:

- Criar um método para a remoção de objetos em movimento, em um sistema de imagens de perspectivas diferentes;
- Criar um processo para o alinhamento das imagens de diferentes perspectivas;

- Criar um processo que identifique os *pixels* referentes ao fundo das imagens, criando uma nova imagem sem oclusão do objetos em movimento;
- Verificar a funcionalidade do método testando-o em imagens do GSV e comparando-o com os trabalhos relacionados de acordo com as métricas estabelecidas.

### 1.3 METODOLOGIA UTILIZADA

Para realizar o desenvolvimento desta ferramenta, primeiramente foi feita uma pesquisa bibliográfica, para buscar trabalhos na área. Logo após foi realizada uma pesquisa exploratória, visando o desenvolvimento de um método para solução do problema. Usando a perspectiva do paradigma tecnocrático, foi dado início ao desenvolvimento do método e testes da ferramenta.

### 1.4 ESCOPO

Este trabalho visa a remoção de objetos oclusores em imagens do GSV, tendo como pressuposto, que as 3 imagens do GSV contenham o objeto oclusor em posições diferentes e o mesmo se mover em velocidade diferente do “carro do GSV”. A remoção do objeto será feita se utilizando da informação redundante das imagens em perspectivas diferentes obtidas do GSV, serão utilizados métodos de baixo custo computacional. Com isso é visada a obtenção de resultados de qualidade e tempo de execução melhores que os outros trabalhos analisados.

### 1.5 ESTRUTURA

O restante do trabalho desenvolvido está dividido da seguinte forma: O Capítulo 2 apresenta os métodos e ferramentas utilizados para explicar os trabalhos relacionados que serão apresentados no Capítulo 3 e o método desenvolvido neste trabalho. No Capítulo 3 é feito um resumo dos trabalhos relacionados, estes foram utilizados para justificar e desenvolver o método utilizado neste trabalho, ao final do capítulo é feito um levantamento das características de cada um. No capítulo 4 é feita a apresentação do método proposto, detalhando seus passos e o processo de funcionamento. No capítulo 5 são exibidos os resultado que foram obtidos ao se testar o método, neste capítulo também serão analisados os resultados obtidos, a fim de identificar problemas no método. No capítulo 6 é feita uma conclusão sobre os estudos feitos no trabalho e os resultados obtidos nos testes realizados.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para o entendimento dos trabalhos relacionados e do projeto desenvolvido, será necessário o conhecimento de alguns fundamentos básicos de processamento de imagem. A seguir será explicado o funcionamento dos métodos utilizados e dos conhecimentos teóricos necessários para o desenvolvimento do trabalho.

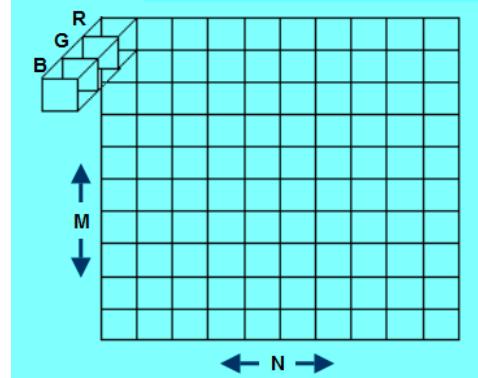
### 2.1 IMAGEM

Uma imagem pode ser descrita como uma função  $f(x,y)$  de duas dimensões, onde as variáveis  $x$  e  $y$  são as coordenadas de um ponto na imagem e o resultado dessa função é a intensidade da luminosidade da imagem no ponto em questão. Estes valores, tanto  $x$  e  $y$  quanto o resultado da função  $f$ , são infinitos, porém em imagens digitais estes valores são discretos, então uma imagem passa a ser uma matriz  $M \times N$  com valores discretos de intensidade de cor em cada posição, como ilustrado na Equação 1 (GONZALEZ; WOODS, 2002).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (1)$$

Em imagens coloridas as imagens são geralmente representadas por uma matriz  $M \times N \times 3$ , no sistema de cores RGB. Este sistema de cores aditivo funciona com a utilização de 3 cores bases: vermelho (red), verde (green) e azul (blue). Com a combinação dessas 3 cores em níveis variantes de 0 a 255, variando a intensidade das 3 cores se consegue gerar um intervalo de cores visíveis. A representação da imagem por matriz RGB é ilustrada na Figura 3. Existem outros tipos de sistemas de cores digitais, como HSV, HLS, CMYK, etc, porém, o mais utilizados deles é o RGB, este também foi o sistema de cores utilizado pelos trabalhos relacionados (SANTOS; VEIGA, 2010).

**Figura 3: Representação tridimensional RGB.**

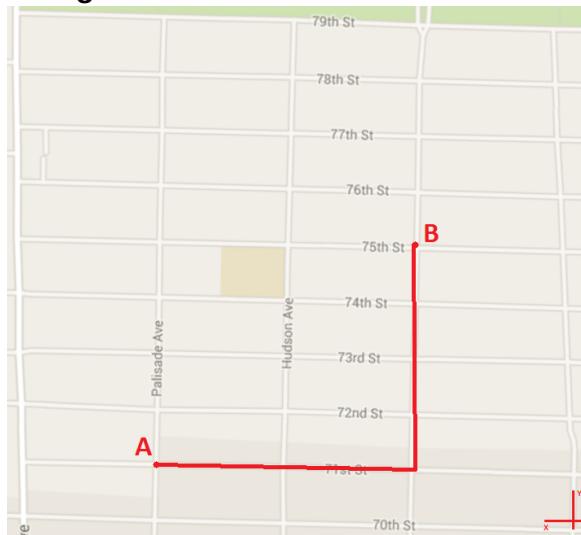


**Fonte:** Produção do próprio autor.

## 2.2 MANHATAM

A distância Manhatam será utilizada na identificação de fundo para a comparação entre os *pixels* de duas imagens. A distância Manhatam tem esse nome devido as ruas de Manhatam, nas quais devido a geometria das ruas, a distância mínima entre um cruzamento A até um cruzamento B consiste na diferença das distâncias entre os cruzamentos em x e y, não importando o caminho percorrido, como ilustra a Figura 4. Esse sistema permite a elaboração de uma fórmula simples,  $Dist(P, Q) = |x_P - x_Q| + |y_P - y_Q|$ , onde a distância entre os pontos P e Q é dada pela soma da diferença em módulo das coordenadas dos pontos (JANSSEN, 2007).

**Figura 4: Distância de Manhatam.**



**Fonte:** Produção do próprio autor.

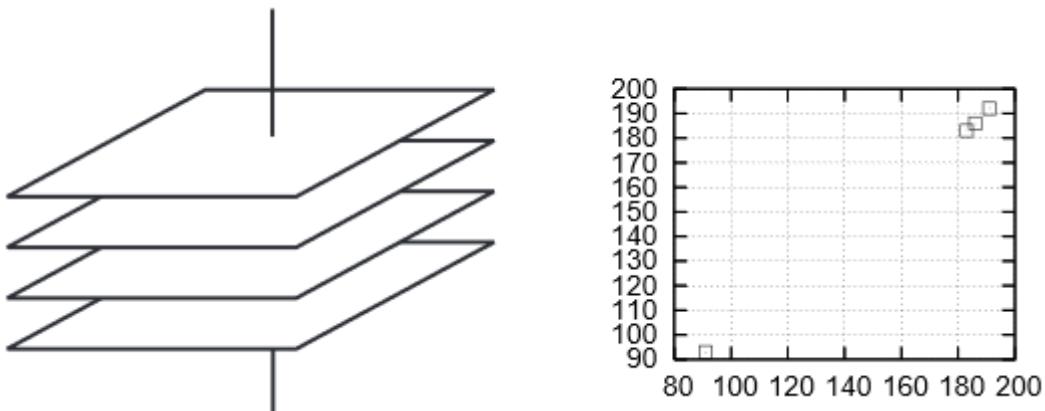
## 2.3 IDENTIFICAÇÃO DE FUNDO

O processo de identificação de fundo é muito utilizado por exemplo em sistemas de vigilância, nos quais uma câmera é fixada e então é retirada uma imagem de controle. Esta imagem nada mais é do que o ambiente vazio, apenas o fundo, após a retirada dessa imagem a sequência de imagens que será gravada é subtraída da imagem de controle. Com este processo é possível determinar os objetos presentes nas “cenas”, estes objetos consistem no resultado da subtração das imagens (BÖHM, 2004).

A dificuldade encontrada nesses métodos de subtração de imagem consistem na variação do fundo, pois podem ocorrer mudanças de iluminação e presença de sombras ou novos objetos estáticos. Existem vários algoritmos para a identificação de fundos, porém estes algoritmos de estado-da-arte acabam por precisar de um tempo de treinamento, tornando o processo lento por exigir muito tempo de processamento para a identificação do fundo da imagem. Estes algoritmos também precisam de uma quantidade de imagens elevada para o resultado ser aceitável (BÖHM, 2004).

Pelos motivos citados anteriormente, será aplicado um algoritmo mais simples. Este algoritmo é baseado no algoritmo de identificação de Böhm (2004), o algoritmo de Böhm (2004) basicamente percorre cada *pixel* das imagens paralelamente, então ele coloca estes *pixels* em um sistema de matriz tridimensional, no qual cada eixo da matriz correspondem as cores do sistema de cores RGB. Um exemplo para tal funcionamento é a Figura 5, onde para melhor visualização temos apenas as cores vermelha e verde com 4 imagens sendo analisadas. Neste processo é possível identificar *outliers*, *pixels* distantes do amontoado de *pixels* na matriz, estes *outliers* basicamente são frutos do objeto em passagem pelo fundo da imagem, assim podemos assumir que o fundo consiste dos *pixels* agrupados perto de um índice da matriz. Com isso é possível estimar um fundo fazendo a média entre os *pixels* agrupados, assim obtendo um fundo genérico para a imagem de fundo (BÖHM, 2004).

**Figura 5: Identificação de fundo matriz bidimensional RG.**



**Fonte:** (BöHM, 2004)

A técnica utilizada neste trabalho consiste na avaliação *pixel à pixel* das 3 imagens, tiradas do GSV, paralelamente. A cada *pixel* da imagem central será feita a comparação entre os *pixels* das imagens laterais e a central, isso é realizado para a determinação através da distância Manhatam de qual *pixel* será utilizado na imagem final gerada pelo método. Este processo será detalhado na descrição do método no capítulo 4, no qual serão descritas os procedimentos utilizados para a comparação entre as imagens.

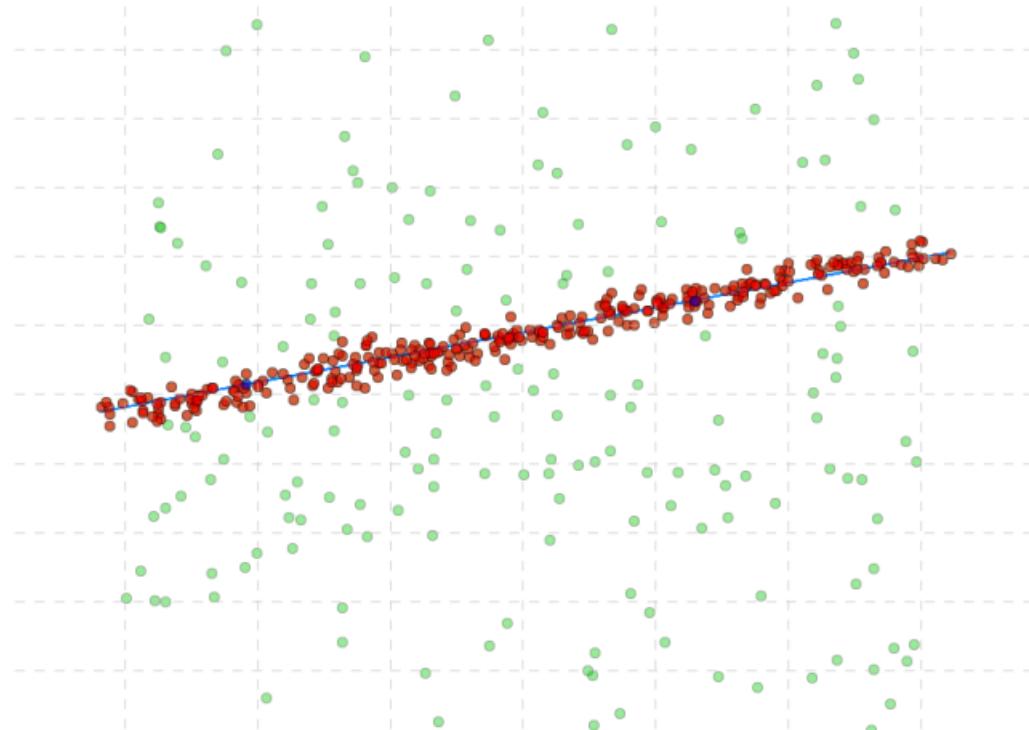
## 2.4 RANDOM SAMPLE CONSENSUS

Para a transformação de perspectiva que é realizada no trabalho para alinhamento das imagens, será necessária a aplicação do método RANSAC (*Random Sample Consensus*) na obtenção da matriz de transformação de perspectiva. O RANSAC é um método iterativo para se estimar parâmetros de modelos matemáticos a partir de uma coleção de dados (PALHARES, 2009).

O algoritmo funciona selecionando pontos de uma coleção com o intuito de criar um modelo matemático para descrever o maior conjunto possível de pontos presentes na coleção. Isso é possível pois é utilizada uma função objetivo, esta serve para se obter a adequação do modelo que está sendo moldado à coleção de pontos que se está utilizando. Um exemplo de resultado do algoritmo RANSAC é ilustrado na Figura 6, na qual a função objetivo consiste na distância entre os pontos e a reta traçada para resolução do problema. O resultado é uma reta que corresponde ao maior número de pontos possíveis na coleção apresentada (FISCHLER; BOLLES,

1981).

**Figura 6: Ilustração do resultado do algoritmo RANSAC.**



**Fonte:** Adaptado de (FISCHLER; BOLLES, 1981)

O algoritmo pode ser dividido em 3 passos:

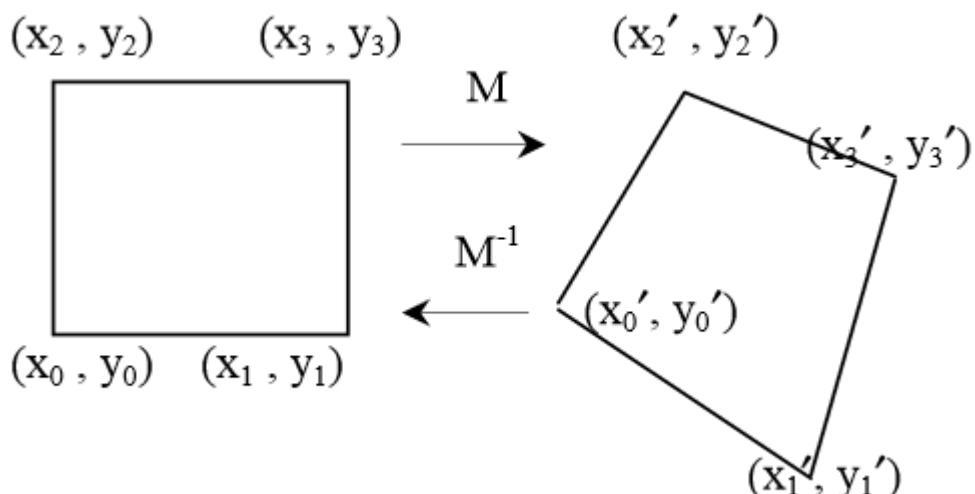
- Selecionar aleatoriamente um subconjunto pertencente à coleção de dados de entrada. No caso de uma reta em um espaço bidimensional dois pontos são o suficiente por exemplo;
- Os pontos selecionados são considerados parte da solução, após isso os pontos que satisfazem o modelo são adicionados em um conjunto solução, de acordo com a avaliação da função objetivo;
- Analisar se a quantidade de pontos encontrados no conjunto solução é o suficiente, baseando-se no limiar que é definido no algoritmo.

Estes passos são repetidos até que se obtenha a solução no passo 3, ou até que a quantidade de iterações definidas no algoritmo seja executada. No fim se obtém um conjunto de dados que podem ser descritos através de uma função matemática.

## 2.5 TRANSFORMAÇÃO DE PERSPECTIVA 2D

A transformação de perspectiva 2D é uma operação de processamento de imagem, na qual uma imagem varia na sua escala, rotação e translação. Esta é usada para mudar a perspectiva da imagem em questão. Isso é possível através da multiplicação da matriz de transformação  $M$  pela imagem a ser tratada, mudando a perspectiva da mesma, como ilustrado na Figura 7 (KIM; JANG; HWANG, 2002).

**Figura 7: Ilustração de Transformação de Perspectiva.**



**Fonte:** (KIM; JANG; HWANG, 2002)

A matriz  $M$ , ilustrada na Equação 2, é gerada a partir da execução do algoritmo de RANSAC. Se utilizando da fórmula do erro da transformação:  $\min \sum_i ((x'_i - (m0 * x_i + m3 * y_i + m6) / (m2 * x_i + m5 * y_i + m8))^2 + (y'_i - (m1 * x_i + m4 * y_i + m7) / (m2 * x_i + m4 * y_i + m8))^2)$  como função objetivo, o algoritmo de RANSAC cria um banco de valores aleatórios para os valores da matriz  $M$ , então a cada iteração do RANSAC ele refina este resultado, definindo os pontos que melhor se encaixam na matriz  $M$ . Para isso ser possível é necessária a identificação de pelo menos 4 pontos equivalentes em cada uma das imagens, isto é necessário para se ter uma maior fidelidade da matriz  $M$  por toda a imagem, pois se levado em consideração apenas um ponto equivalente será calculada apenas a transformação da imagem naquele ponto. Depois da execução do RANSAC, o mesmo retornará os pontos de  $M$  que possibilitam a transformação de qualquer ponto da imagem  $I$  em pontos da imagem  $I'$ , a imagem  $I$  é a matriz antes da transformação ser efetuada e a imagem  $I'$  é a imagem já transformada (THOMAS;

KAREEM; BOWYER, 2012).

$$M = \begin{pmatrix} m_0 & m_3 & m_6 \\ m_1 & m_4 & m_7 \\ m_2 & m_5 & m_8 \end{pmatrix}_{3 \times 3} \quad (2)$$

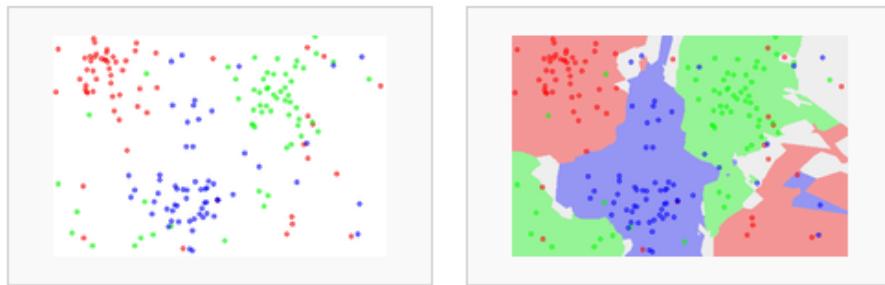
Para aplicar a transformação a cada ponto da imagem é utilizada a fórmula ilustrada na Equação 3. Nesta fórmula os valores de  $x$  e  $y$  são os valores da posição dos *pixels* da imagem original, a matriz apresentada é a matriz  $M$  de transformação, esta caracteriza a homografia entre as duas imagens, que consiste no mapeamento dos pontos da imagem  $I$  para a imagem  $I'$ , como evidenciado pelo valor 1 na Equação 3 as coordenadas utilizadas são homogêneas caracterizando uma homografia 2D. O resultado da Equação 3 é a imagem que origina da transformação de perspectiva da imagem original (THOMAS; KAREEM; BOWYER, 2012).

$$[x' \ y' \ z'] = [x \ y \ 1] * M \quad (3)$$

## 2.6 K NEAREST NEIGHBORS

Para realizar a comparação entre pontos de imagens relacionadas, alguns trabalhos, como Böhm (2004), Flores e Belongie (2010), se utilizam do KNN (*K Nearest Neighbors*). O KNN é um algoritmo para a classificação de objetos baseados em uma amostragem de treinamento. No caso a entrada do algoritmo conta com uma amostragem de objetos já classificados, com estes objetos o algoritmo seleciona os  $K$  objetos mais próximos em uma coleção de objetos, então o mesmo os analisa e classifica de acordo com a influência da vizinhança e da amostra de treinamento. Um exemplo do funcionamento do KNN pode ser observado na Figura 8 (CUNNINGHAM; DELANY, 2007).

**Figura 8: Exemplo de Funcionamento do KNN.**



**Fonte:** Adaptado de (CUNNINGHAM; DELANY, 2007)

Para identificar qual dos itens é mais parecido com o outro o algoritmo se utiliza de uma fórmula para mensurar a igualdade das características entre cada objeto, em objetos discretos geralmente se utiliza a distância euclidiana. Mas como citado o julgamento de a qual classe pertence certo objeto não fica a cargo apenas da sua distância em relação a amostragem, é verificada a vizinhança do mesmo, esta tem um peso na conta de classificação, na qual se toda a vizinhança do objeto for de uma certa classe é provável que o mesmo também seja desta classe.

## 2.7 INPAINTING

*Inpainting* é um método que tem como objetivo a correção de imagens que foram perturbadas por ruídos ou contém alguma parte da imagem faltando. O método de *inpainting* é largamente utilizada em processamento de imagem para preencher lacunas que são deixadas nas imagens, muitas vezes estas são criadas por remoção de objetos. Um exemplo de funcionamento do algoritmo de *inpainting* pode ser observado na Figura 9 (CAI; CHAN; SHEN, 2008).

**Figura 9: Exemplo de *inpainting*.**



**Fonte:** (CAI; CHAN; SHEN, 2008)

Segundo Criminisi e Toyama (2003), existem 3 tipos de *inpainting*:

- *Inpainting* Estrutural: Este algoritmo se utiliza das informações geométricas da região vizinha à região que será restaurada, com isso ele foca na consistência geométrica da imagem, geralmente utilizado em imagens sem textura para a restauração da geometria, pois o mesmo não restaura informações de textura.
- *Inpainting* por Textura: Consistem basicamente em preencher o espaço a ser restaurado com as informações de textura da vizinhança do mesmo. Esse processo costuma ter bons resultados em ruídos;
- *Inpainting* Estrutural e de Textura: Consiste na mistura dos dois métodos citados acima, onde primeiro se restaura as informações geométricas da imagem, após isso para cada geometria restaurada é feito um *inpainting* de textura, assim recuperando a textura referente a cada parte geométrica da imagem.

## 2.8 DETECCÃO DE DESCRIPTORES

Os métodos para a detecção dos descritores serão estudados pois serão utilizados na identificação de, no mínimo, 4 pontos que serão utilizados na transformação de perspectiva. Estes métodos identificam pontos de interesse que contém descritores que identificam os pontos, com isso é possível a comparação entre estes descritores e assim identificar pontos iguais nas imagens.

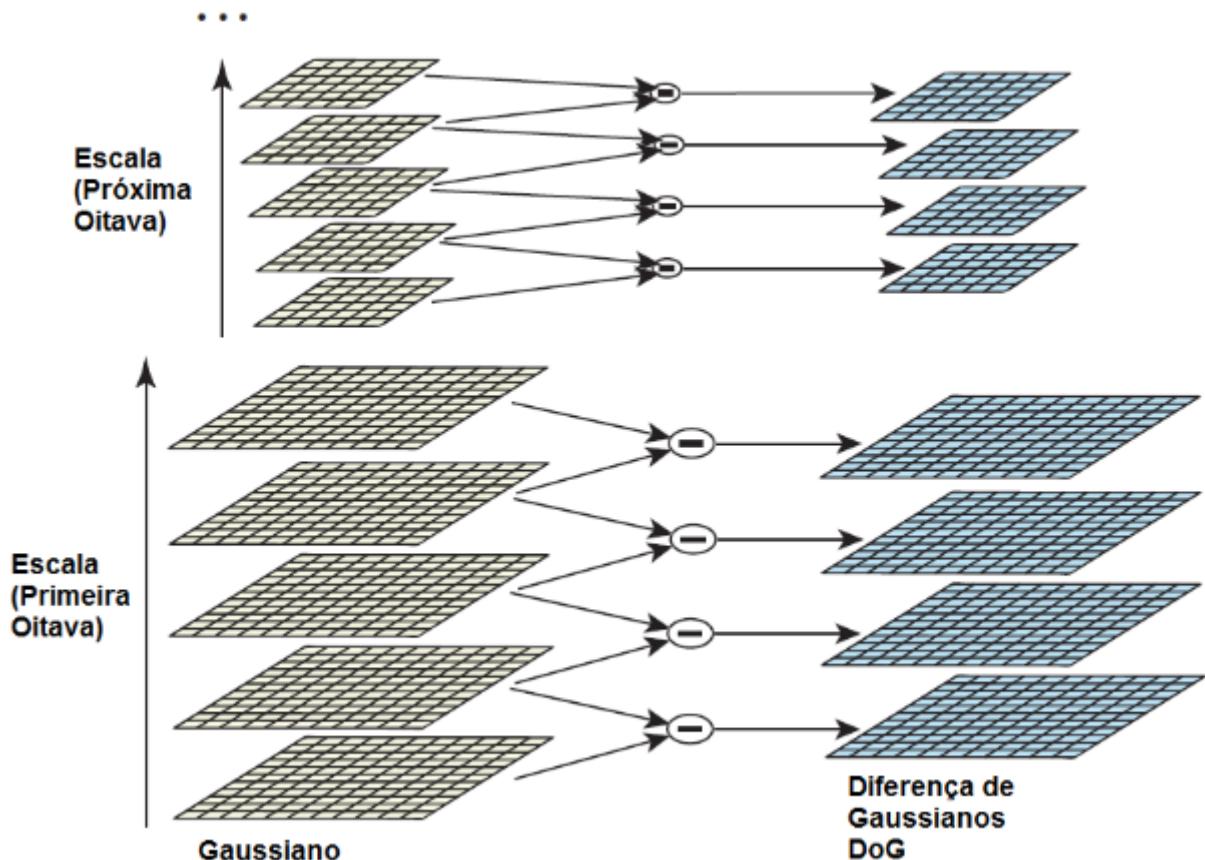
### 2.8.1 SCALE INVARIANT FEATURE TRANSFORM (SIFT)

O SIFT é uma técnica de processamento de imagens que permite a detecção e extração de descritores locais, razoavelmente invariáveis a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de perspectiva. Estes descritores locais identificam características da vizinhança, permitindo a comparação entre pontos pela semelhança de suas vizinhanças. Proporcionou um passo importante na representação de objetos através de suas partes, gerando um grande avanço em diversas áreas, como, reconhecimento de objetos, navegação automática de robôs, rastreamento e criação de imagens panorâmicas (LINDEBERG, 2012).

O processo de identificação feito pelo SIFT pode ser dividido em duas partes: a seleção de pontos de interesse e; criação de descritores para os mesmos. Para a parte da detecção de pontos se utiliza uma representação em forma de uma pilha de imagens que estas diferem na escala, como ilustra a Figura 10. Isto é obtido a partir da convolução de um filtro Gaussiano. Essa pilha de imagens em vários níveis de

detalhe é geralmente denominada oitavas de Gaussianas. Cada nível  $f_i$  da pirâmide contém uma oitava obtida através da sub-amostragem sobre oitava  $f_{i-1}$ , localizada no nível imediatamente inferior. Esse processo é repetido recursivamente até que se obtenha o nível de representação desejado, geralmente contendo blocos de  $8 \times 8$  pixels, associado as maiores escalas observáveis (OPENCV, 2014).

**Figura 10: Representação multiescala usada no SIFT.**



**Fonte:** (LOWE, 1999)

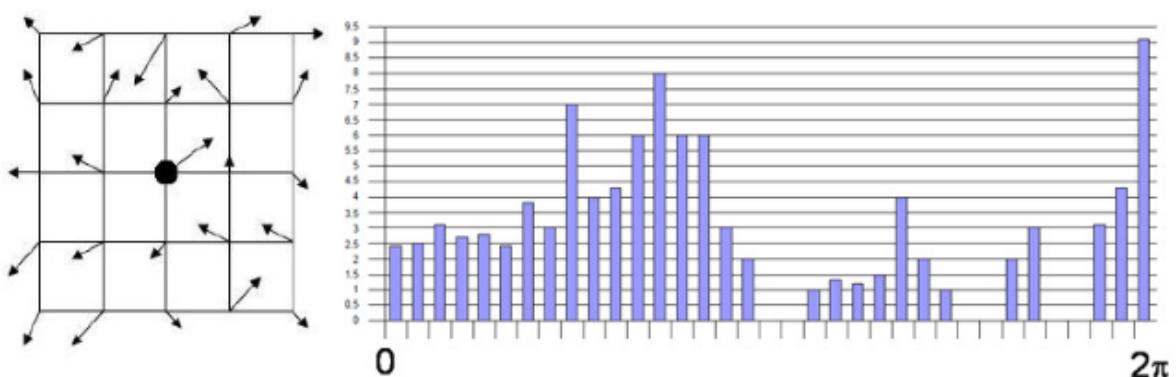
Após serem geradas as oitavas Gaussianas, a obtenção de descritores SIFT é feita nas seguintes etapas (GONZALES; LUIS; MEGGIOLARO, 2010):

- Detecção de extremos (máximos e mínimos): Nesta primeira etapa é feita procura para todas escalas e localizações de uma imagem. Isto é feito utilizando-se a função DoG (Diferença das Gaussianas), esta função gera novas oitavas de diferença da Gaussiana, como ilustrado na direita da Figura 10. Com essa diferença pode se identificar os pontos de interesse invariáveis à escala e rotação, estes pixels se caracterizam pelo valor DoG maior que os de todos os seus vizinhos;

- Localização de pontos chave: Para cada ponto em que foi detectado um extremo, um modelo detalhado é ajustado de modo a se determinar localização e escala. Pontos chaves, ou pontos de interesse, são então selecionados baseando-se em medidas de estabilidade, a estabilidade dos pontos é calculada a partir de uma série de Taylor e visa a retirada de pontos susceptíveis à instabilidade por ruído. Nesta etapa também são descartados os pontos pertencentes às arestas, pois estas têm um "forte" valor de DoG e podem ser escolhidas como pontos extremos, este processo é feito a partir da matriz Hessiana composta pelas derivadas dos pontos extremos;

- Definição de orientação: É definida a orientação de cada ponto chave através dos gradientes locais da imagem. A orientação dos gradientes locais é feita a partir de um histograma que cobre todas as possíveis orientações do gradiente, como ilustra a Figura 11, para cada pico no histograma será gerada uma orientação para o ponto em questão, isso fará com que o mesmo seja mais estável em futuras transformações. Toda operação a partir de então será feita com relação a dados da imagem transformados em relação à orientação, escala e localização de cada ponto chave. Desta maneira se obtém invariância a estas transformações.

**Figura 11: Histograma de orientação dos gradientes.**

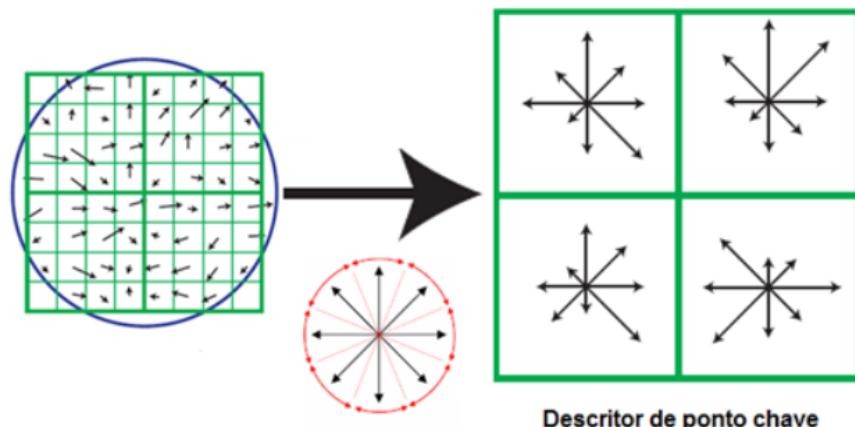


**Fonte:** (GONZALES; LUIS; MEGGIOLARO, 2010)

- Descritor dos pontos chaves: Nesta etapa é feita a construção dos descritores ao se medir gradientes locais em uma região vizinha a cada ponto de interesse. Estas medidas são então transformadas para uma representação que permite níveis significativos de distorção e mudança na iluminação. Para cada ponto de interesse, são definidas  $n \times n$  regiões, com  $k \times k$  pixels cada, ao redor da localização do ponto chave. Geralmente  $n = k = 4$ . Para cada região, é feito um histograma em 8 direções,

como ilustrado na Figura 12. Este histograma é feito com as magnitudes dos *pixels* pertencentes a cada região. O descriptor é então representado pelos histogramas das regiões.

**Figura 12: Geração dos descritores de pontos chave.**



**Fonte:** Adaptado de (LOWE, 1999)

### 2.8.2 Histograma de Gradientes Orientados (HOG)

A ideia principal do descritor HOG é dividir a imagem em pequenas regiões conectadas, assim o método descreve este objeto através da distribuição dos gradientes de intensidade dos *pixels* ou pelas direções das bordas, como ilustrado na Figura 13.

**Figura 13: Exemplo de funcionamento do HOG.**



**Fonte:** Adaptado de (DALAL; TRIGGS, 2005)

O processo para gerar o descritor pode ser dividido em quatro etapas: cálculo do gradiente em cada *pixel*, agrupamento dos *pixels* em células, agrupamento das células em blocos e obtenção do descritor.

- Cálculo do gradiente dos *pixels*: Primeiramente para o cálculo do gradiente no *pixel* em questão são utilizadas duas máscaras, uma no eixo vertical e uma no eixo horizontal, estas geralmente são máscaras 1D porém podem ser utilizado outras máscaras como o filtro de Sobel, com o resultados das mesmas é calculado o gradiente neste *pixel*;

- Agrupamento em células: Para cada certa região são agrupados os histogramas dos *pixels* presentes na mesma, a esse agrupamento se da o nome de células;

- Criação de blocos: Os blocos consistem em um agrupamento das células e seus histogramas;

- Obtenção do descritor: Na criação do descritor é feita uma normalização dos valores dos histogramas de todas as células e de todos os blocos, este processo é feito com base em seus próprios valores.

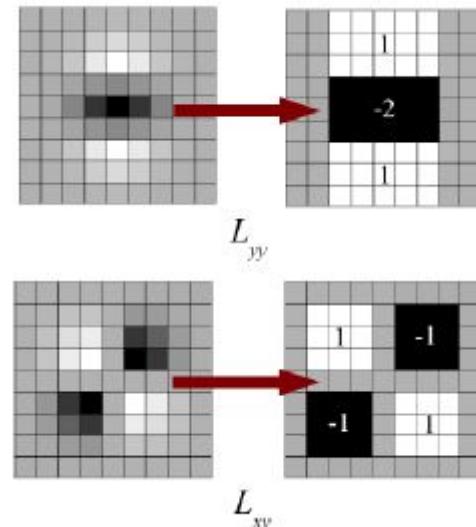
O último passo do sistema é a identificação dos objetos, esta é feita comparando os histogramas dos blocos gerados com o HOG com os histogramas de uma base de dados contendo o objeto em diferentes situações. Estas comparações geralmente são realizadas por um algoritmo classificador com aprendizado supervisionado. No artigo original foi usado o SVM (*Support Vector Machine*), utilizando-se de um núcleo linear, este foi utilizado para a detecção de pedestres em um vídeo (DALAL; TRIGGS, 2005).

### 2.8.3 SPEEDED UP ROBUST FEATURES (SURF)

O método SURF é uma alteração do método SIFT. A grande diferença é a utilização da convolução de filtros para a obtenção de pontos invariantes na variação de escala, assim o tempo para a escolha de pontos invariantes em escala se reduz drasticamente, um exemplo desse filtro pode ser observado na Figura 14. Outra melhoria no SURF é o cálculo da orientação, no SURF ele é feito aplicando os resultados verticais e horizontais das wavelets em uma vizinhança de tamanho 6 ao redor do ponto em questão, estes resultados então são plotados em um gráfico para se obter a orientação dominante, como ilustra a Figura 15. Após este processo é utilizado wavelets novamente para uma vizinhança de 20 dividida em 4 regiões de  $4 \times 4$  é então calculada a soma das orientações em cada uma dessas subdivisões,

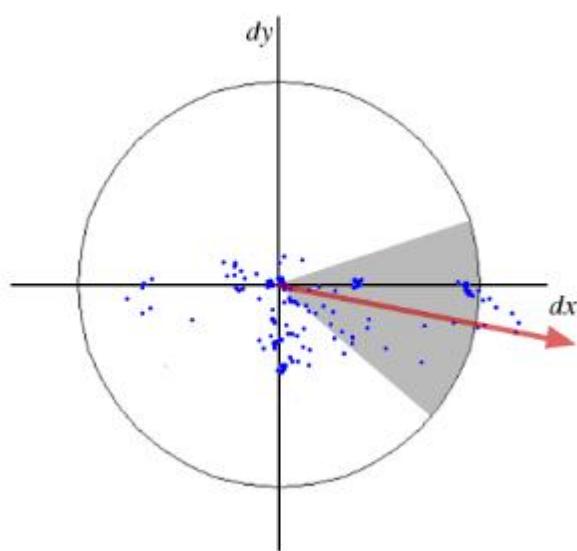
dando uma amostragem de orientações maior.

**Figura 14: Filtro de aproximação utilizando convolução.**



**Fonte:** Adaptado de (OPENCV, 2014)

**Figura 15: Cálculo da orientação.**



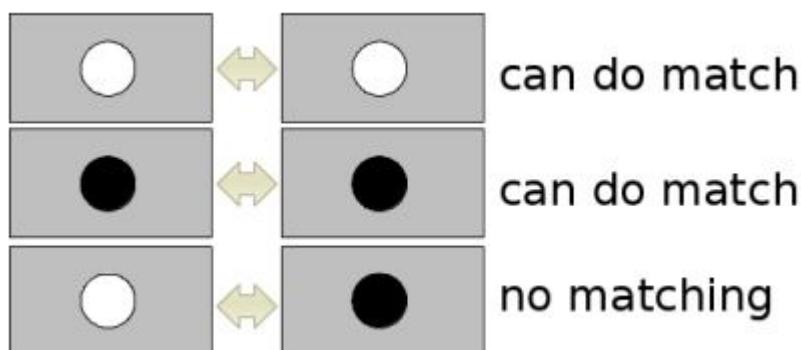
**Fonte:** (OPENCV, 2014)

A última melhoria feita no método é a análise das manchas no processo de matching, são pré-analisadas, com um sistema simples de contraste de fundo se analisa se existe o mesmo contraste com o fundo nos dois pontos chaves a

serem analisados, então se o *matching* não ocorre é descartado antes de qualquer processamento, como ilustra a Figura 16 (OPENCV, 2014).

Com isso o desempenho do SURF melhora significativamente, cerca de 3 à 7 vezes melhor que o SIFT. Porém o detector de pontos chaves utilizado pelo SURF tem força maior em cantos e junções, isso significa que o mesmo tem uma quantidade de pontos chaves mais reduzido do que o método SIFT, mas os autores garantem que os pontos encontrados são tão estáveis quantos os encontrados pelo SIFT.

**Figura 16: Matching por contraste do fundo.**



**Fonte:** (OPENCV, 2014)

As principais diferenças apresentadas entre SURF e SIFT são:

- SURF usa um modelo aproximativo do espaço de escala, baseado em imagens integrais;
- O detector de pontos usado pelo SURF não necessita que o tamanho original da imagem seja alterado;
- A detecção dos pontos de interesse no SURF é baseada na supressão de não-máximos do determinante da matriz Hessiana enquanto o SIFT utiliza uma aproximação do traço dessa matriz. Em consequência disto, SURF tende a detectar cantos e regiões com textura enquanto SIFT geralmente tende a detectar bolhas e arestas.

### 3 TRABALHOS RELACIONADOS

A seguir serão apresentados cronologicamente os trabalhos relacionados ao tema do trabalho, estes trabalhos foram estudados com a finalidade da criação de um método para a remoção de pessoas em imagens em diferentes perspectivas. O método criado foi uma junção de métodos utilizados nos trabalhos relacionados com algumas modificações.

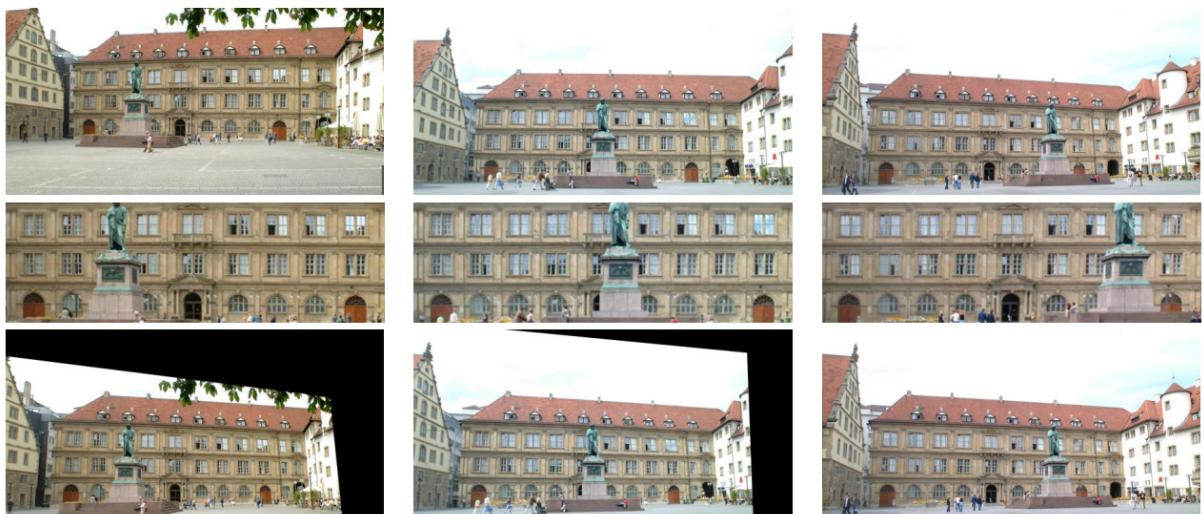
#### 3.1 REMOÇÃO DE OBJETOS OCLUSORES DE FACHADAS PARA MONTAGEM DE UMA MODELAGEM 3D

Böhm (2004) objetiva gerar um método para retirar objetos que possam atrapalhar na visão das fachadas dos prédios e casas, com as imagens geradas por este método, será criada uma cidade através de texturização de modelos 3D. As imagens são capturadas por uma câmera de diferentes perspectivas, a quantidade de fotos pode variar para cada fachada dependendo da quantidade de obstruções. Estas imagens serão utilizadas para obter informação redundante sobre as fachadas das casas e prédios, pois como a perspectiva mudou de uma foto para a outra a chance de se ter uma fachada livre de oclusão é grande. Após feita a extração da imagem sem oclusão esta será utilizada nos modelos 3D, assim dando realismo ao modelo.

O método proposto por Böhm (2004) consiste em 4 etapas:

- Identificação dos cantos: são identificados os *pixels* dos cantos das fachadas através de identificação em todas as imagens capturadas, tomando como os cantos das imagens mediana, o resultado da obtenção pode ser visualizado na segunda linha da Figura 17;
- Transformação de perspectiva: depois de serem obtidos os pontos referentes aos cantos das fachadas, eles são alinhados através de uma transformação de perspectiva, o resultado da transformação das imagens pode ser visualizado na terceira linha da Figura 17;
- Identificação de fundo: com a transformação é possível identificar as fachadas das casas, visto que será possível estimar o fundo da imagem através da semelhança entre os *pixels* das fachadas. Após a identificação é gerada uma imagem a partir da combinação dos *pixels* elencados baseados nas imagens vizinhas, a imagem gerada é então livre dos objetos que obstruíam as fachadas;
- Corte da fachada: por fim a fachada é recortada para poder ser utilizada nos modelos 3D.

**Figura 17: Alinhamento por Transformação de Perspectiva.**



**Fonte:** (BöHM, 2004)

Os resultado obtidos por Böhm (2004) são efetivos na retirada de objetos quando se utilizam mais de 3 imagens, porém como as fotos são tiradas de uma distância grande das fachadas somente objetos grandes são realmente retirados da imagem, como ilustrado na Figura 18. Para a retirada de objetos menores, como pessoas, seria necessária a captura de imagens mais próximas a fachada, porém isso aumentaria o número de fotos para cobrir a mesma. Outro problema é que o sistema não é totalmente automatizado, os pontos chave das fachadas precisam ser identificados manualmente.

**Figura 18: Resultado da fusão.**



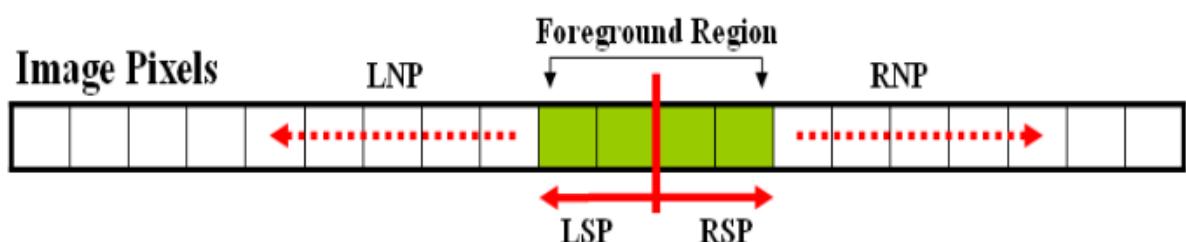
**Fonte:** (BöHM, 2004)

### 3.2 REMOVENDO OBJETOS DO PRIMEIRO PLANO UTILIZANDO INFORMAÇÃO DE PROFUNDIDADE

O objetivo do método desenvolvido por Lee e Kim (2009) é remover os objetos que estão obstruindo o fundo de uma imagem se utilizando das informações presentes em outras imagens de perspectivas diferentes. O sistema consiste em 2 partes: a primeira se resume a remoção da camada em que se encontra o objeto, assim o removendo e deixando um espaço preto no lugar; a segunda parte do sistema consiste em preencher o espaço com os *pixels* vizinhos encontrados na imagem com perspectiva diferente, assim resultando na imagem sem o objeto em questão.

No processo de remoção do objeto primeiramente é feita a divisão da imagem em camadas através de um mapa de profundidade se utilizando de segmentação por regiões. Depois é retirada a camada que contém o objeto obstruindo, ela é detectada através das informações de profundidade da imagem. Na segunda etapa do método na imagem com a perspectiva diferente é detectada a camada referente ao objeto, isso é feito comparando as distância das camadas entre as duas imagens. Por fim é feita a transferência dos *pixels* da imagem de perspectiva diferente para o espaço deixado na imagem original seguindo o esquema ilustrado na Figura 19. Neste esquema os *pixels* do espaço deixado pela remoção do objeto é dividido em duas seções, estas são preenchidas com os *pixels* referentes as bordas das mesmas seções na imagem com a perspectiva alterada. Na Figura 19 os *pixels* LNP e RNP são os *pixels* da borda do objeto da imagem com a perspectiva diferenciada, e os *pixels* LSP e RSP são os *pixels* pretos deixados pela remoção do objeto na imagem original.

**Figura 19: Processo Preenchimento dos *pixels***

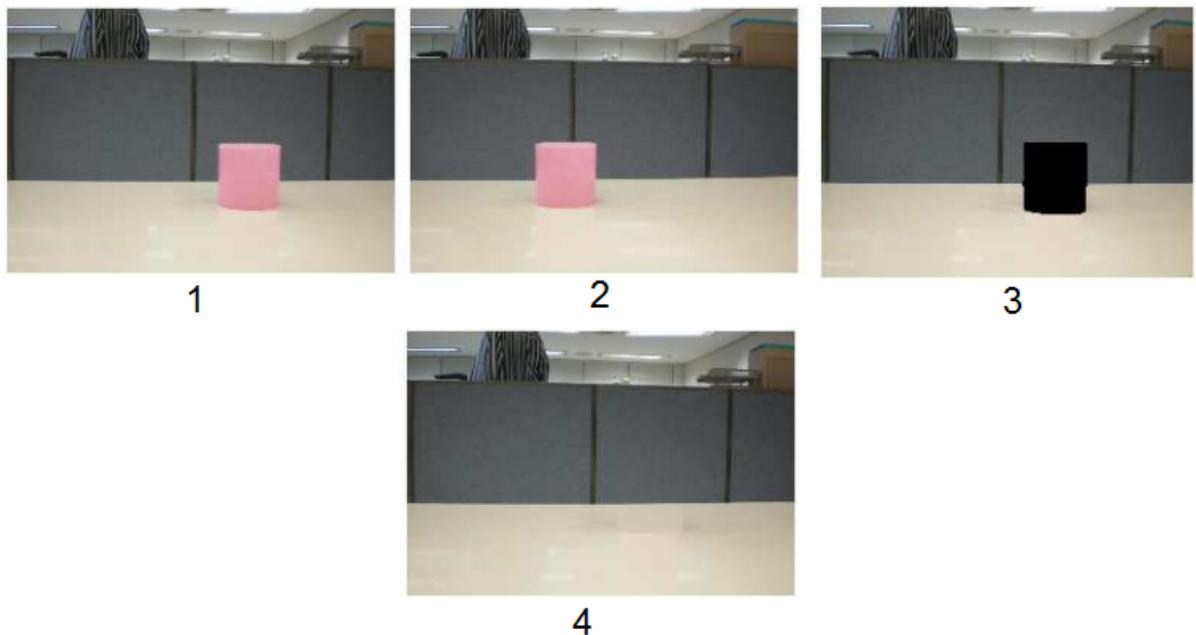


**Fonte:** (LEE; KIM, 2009)

Os resultados obtidos pelo trabalho de Lee e Kim (2009) removem os objetos completamente em todos os casos, porém acabam deixando uma certa sombra no local. O método utilizado tem um tempo de execução rápido se comparados aos

outros trabalhos analisados, os algoritmos utilizados são lineares de acordo com a quantidade de objetos presentes na imagem. Outra vantagem do método é por este ser totalmente automatizado, ele detecta o objeto mais a frente na imagem e o remove, porém pelo mesmo motivo ele não é muito flexível. Os resultados do método dependem muito da mudança de perspectiva ocorrida na imagem, esta tendo que ser uma mudança brusca, com isso o mesmo não pode ser utilizado nas imagens do GSV, que tem variação menores de perspectiva. Um exemplo da execução completa do algoritmo pode ser visualizada na Figura 20.

**Figura 20: Processo Preenchimento dos pixels.**



**Fonte:** Adaptado de (LEE; KIM, 2009)

### 3.3 REMOVENDO PEDESTRES DO GSV UTILIZANDO SIFT E HOG

Com o intuito de proteger a privacidade das pessoas no GSV, Flores e Belongie (2010) propõe um método automatizado para a remoção das pessoas presentes nas fotos do GSV. Flores e Belongie (2010) partem do princípio de que uma imagem tirada do GSV tem imagens “vizinhas”, estas imagens tem uma grande quantidade de informação redundante, ou seja os objetos presentes nestas imagens são os mesmos da imagem inicial. Com esses dados redundantes é possível retirar uma pessoa que está presente em uma das imagens se utilizando dos dados contidos nas imagens “vizinhas”, como podemos ver na Figura 21 as informações do fundo obstruídas pelo

pedestre nas duas imagens são diferentes.

**Figura 21: Relação de informação redundante em duas imagens do GSV.**



**Fonte:** (FLORES; BELONGIE, 2010)

Uma das diferenças entre essas imagens seria a mudança de perspectiva, com isso é necessária a utilização de uma transformação de perspectiva. Tal operação é possível devido a maioria das imagens do GSV serem tiradas de áreas urbanas logo existe uma superfície planar na imagem (as fachadas das casas e prédios), assim é possível fazer uma referência entre as imagens, alinhando as mesmas de acordo com as fachadas. Na Figura 22 foi feita a transformação de perspectiva para alinhar as duas imagens de acordo com suas fachadas, alinhando uma das fachadas aos cantos das imagens. Estes pontos são extraídos das duas imagens com a utilização do SIFT, após isso é feita a transformação com a ajuda do algoritmo de RANSAC.

**Figura 22: Resultado da Transformação de Perspectiva.**



**Fonte:** (FLORES; BELONGIE, 2010)

A próxima etapa a ser feita pelo método é a identificação das pessoas presentes na imagem, isso é realizado através de um algoritmo desenvolvido com base no HOG. O algoritmo modificado retorna uma caixa de detecção ao redor da pessoa, após a detecção das bordas poderia ser feito uma relação por *pixel* entre as caixas

detectadas, porém em alguns casos as caixas se sobrepõem, como ilustrado na Figura 23. Para não substituir *pixels* do pedestre da primeira imagem para *pixels* do fundo da segunda imagem é então feito um cálculo para se descobrir o *pixel* de menor valor de luminosidade entre os candidatos, então este será o *pixel* eleito como fundo, que será mantido na região de sobreposição, assim gerando a imagem final livre de oclusão.

**Figura 23: Sobreposição das Caixas Detectadas.**



**Fonte:** (FLORES; BELONGIE, 2010)

A aplicação do método em imagens do GSV em várias localidades urbanas resultou na remoção completa das pessoas, porém em alguns casos o método acaba por deixar alguns vestígios da presença de pessoas, como ilustra a Figura 24. O maior dos motivos desses acontecimentos, segundo Flores e Belongie (2010), é a possibilidade do pedestre ter andado juntamente com o carro do Google, assim a pessoa continua a obstruir a parte da fachada na imagem “vizinha”. Outro problema que o método encontrou é na função que avalia qual *pixel* será utilizado, muitas vezes ela acaba por escolher o *pixel* da pessoa ao invés do *pixel* do fundo, isso devido as cores das roupas que as pessoas estão utilizando, muitas vezes parecida com a cor

do fundo. Uma alteração que o autor propõe para trabalhos futuros é a utilização de mais imagens da “vizinhança” não apenas uma. Isso poderá solucionar grande parte dos problemas vistos nos resultados. Outro problema observado neste trabalho é o desempenho do método, visto que o mesmo não foi otimizado para se utilizar de poucos recursos computacionais, pois o mesmo se utiliza de dois métodos com custo computacional elevado, o SIFT e o método de reconhecimento de pessoas.

**Figura 24: Vestígio da Pessoa Retirada na Imagem Resultante.**



**Fonte:** (FLORES; BELONGIE, 2010)

### 3.4 SUBSTITUINDO PEDESTRES DO GSV UTILIZANDO HOG E INPAINTING

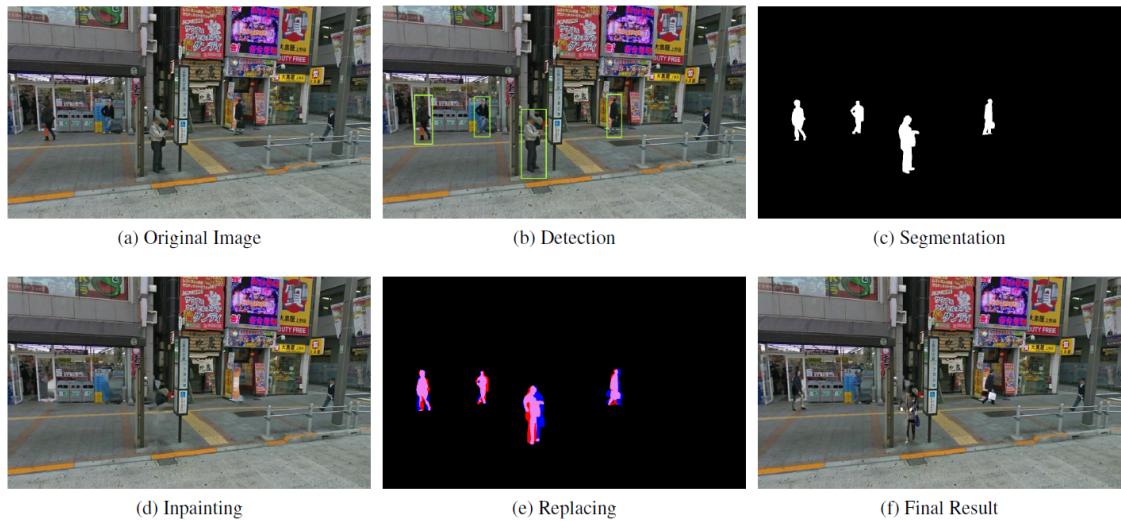
Nodari, Vanetti e Gallo (2012) também partem da justificativa da privacidade das pessoas para a retirada das mesmas da imagens presentes no GSV, porém diferentemente de Flores e Belongie (2010) eles as substitui por imagens de pessoas

de um banco de dados com autorização de publicação de imagens. Os autores primeiramente realizam a detecção das pessoas a serem substituídas, depois disso é retirada a silhueta identificada deixando um espaço preto na imagem. Então é preenchido o espaço com as cores vizinhas ao espaço deixado, isso para não aparecer partes pretas na imagem final, pois nem sempre a pessoa do banco de dados estará em uma posição igual a pessoa da imagem do GSV. Por fim é feita a escolha da imagem no banco de dados mais parecido com a pessoa detectada no GSV, então esta é colocada sobre a posição inicial da pessoa detectada pelo método.

É utilizado o método HOG para a detecção das pessoas a serem retiradas da imagem, quadro (b) na Figura 25, a retirada é efetuada utilizando o algoritmo MNOS (*Multi-Net for Object Segmentation*), este é um algoritmo baseado em redes neurais utilizado para a segmentação de imagens. Por utilizar-se do HOG este método compartilha de um problema também identificado no trabalho de Flores e Belongie (2010), pois se a pessoa estiver utilizando algum acessório como guarda chuvas ou bolsas, estes não serão detectados pelo HOG, logo não serão removidos. O método para preenchimento do “buraco” deixado na imagem é feito através de *inpainting*, quadro (d) na Figura 25, após preenchido o espaço deixado pela pessoa removida é realizada uma busca no banco de imagens para a escolha de uma imagem apropriada. Esta escolha é feita a partir da máscara de resultado do algoritmo MNOS, este contém a silhueta destacada da imagem do GSV, então através da fórmula de distância de Chamfer é decidida qual imagem do banco de dados será utilizada. Para a conformação dos aspectos visuais do modelo do banco de dados é utilizado um algoritmo para a correção dos valores de brilho e contraste da imagem do banco de dados para ficarem parecidas com a da imagem do GSV, tornando praticamente imperceptível a troca da pessoa na imagem.

Como resultado foram obtidas imagens onde as substituições são praticamente imperceptíveis, exceto em casos onde as pessoas carregavam itens, ou as sombras eram muito nítidas nas imagens. Porém, testes feitos pelos próprios autores mostram um dos grandes problemas deste método, para a substituições de uma pessoa em uma imagem do GSV demorou em média 21 segundos, logo este método é lento, visto que também se utiliza de métodos custosos computacionalmente, como o HOG para a detecção de pessoas para obter um resultado com uma qualidade elevada. Um exemplo do processo que o método efetua pode ser visualizado na Figura 25.

**Figura 25:** Processo completo do método proposto por Nodari, Vanetti e Gallo (2012).

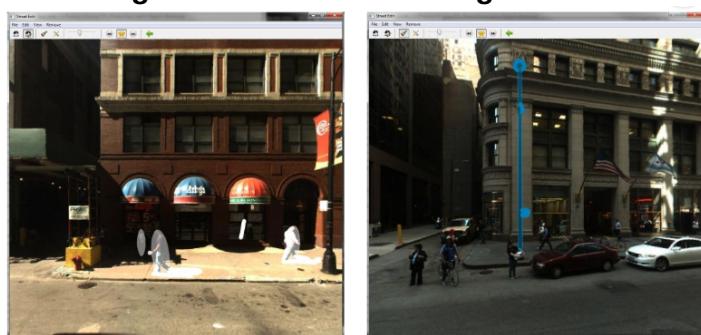


**Fonte:** (NODARI; VANETTI; GALLO, 2012)

### 3.5 FERRAMENTA PARA A REMOÇÃO DE PEDESTRES UTILIZANDO INPAINTING

O trabalho de Haro (2012) apresenta uma GUI (*Graphical User Interfaces*) para a seleção de pessoas a serem removidas em uma imagem, como ilustrado na Figura 26. Após a seleção ser feita ele remove as pessoas e reconstrói o fundo da imagem. Para remover as pessoas é gerada uma máscara baseada nas seleções feitas pelo usuário, então é feita uma subtração da máscara com a imagem original, assim removendo as pessoas. O processo de reconstrução é possível pois as imagens são capturadas em vários *frames*, então com a informação redundante é feita por um processo de *inpainting* a recuperação do fundo da imagem.

**Figura 26:** Interface do Programa.



**Fonte:** (HARO, 2012)

A máscara utilizada na subtração da imagem é gerada se utilizando um algoritmo de *watershed*, um algoritmo de segmentação de regiões, para se obter os limites da imagem da pessoa a partir da marcação do usuário, esse processo é feitos em todos os *frames* da imagem. Depois disso é feito um processo de identificação dos pontos de interesse de cada *frame* através de SURF, isso é realizado para possibilitar o processo de alinhamento utilizando-se o RANSAC. Por fim é feito uma fusão entre as imagens dos *frames* já com as pessoas removidas. Para suavizar o resultado da fusão é realizado um método de *inpainting* para eliminar possíveis áreas encobertas pelas imagens. As falhas encontradas se a fusão fosse realizada sem o *inpainting* podem ser visualizada na Figura 27.

**Figura 27: Fusão das Máscaras dos frames.**



**Fonte:** Adaptado de (HARO, 2012)

Como resultado Haro (2012) obteve imagens nas quais os objetos selecionados foram completamente removidos. Porém o método utiliza muito trabalho manual da parte do usuário, não desejável em nenhum método computacional. Outra desvantagem do método é a necessidade da captura ser controlada, no caso a mesma terá que ser feita em multi-*frame* para se obter vários *frames* da mesma imagem.

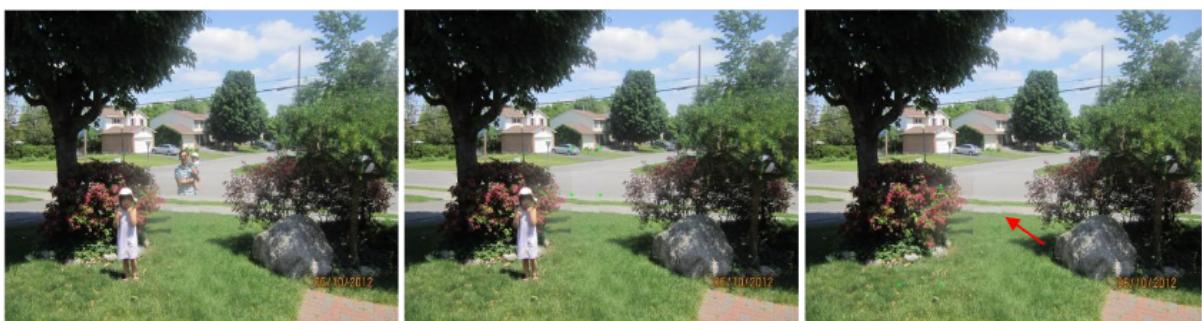
### 3.6 REMOÇÃO DE OBJETOS EM IMAGENS MULTI-VIEW POR CAMADAS UTILIZANDO SEGMENTAÇÃO DE REGIÕES

O método de Dubois e Ho (2012) tem como objetivo a remoção de objetos de uma imagem através da fusão de imagens do mesmo local com perspectivas diferentes. O processo pode ser dividido em 3 partes: achar os pontos de correspondência entre as imagens capturadas; dividir as imagens em diferentes camadas utilizando os pontos reconhecidos anteriormente; e por fim segmentação

das regiões a serem substituídas na imagem original.

Os pontos de referência nas imagens são obtidas através de ASIFT (*Affine-SIFT*). O ASIFT se trata de uma versão melhorada do SIFT, a melhoria consiste em analisar os pontos detectados com variação no ponto de vista, com isso o método consegue retornar pontos mais invariáveis a mudanças de perspectiva. Depois de obtidos os pontos, eles são divididos em camadas checando a similaridade entre os pontos segundo um limiar. Depois de identificadas as camadas de cada imagem, são identificados os pontos correlacionadas entre as imagens, então onde é identificada uma diferença entre o conteúdo das camadas, elas são substituídas pelas camadas da outra imagem, teoricamente sem o objeto. Para recuperar os possíveis erros de alinhamento de imagem é utilizado um método de *inpainting*. O processo de junção de cada camada pode ser visualizado na Figura 28.

**Figura 28: Processo de Junção dos frames.**



**Fonte:** (DUBOIS; HO, 2012)

Os resultados obtidos pelo método de Dubois e Ho (2012) foram imagens onde a remoção das pessoas foi completa, porém deixando um pouco “borrada” a área onde se encontrava as pessoas. Um dos problemas encontrados neste método é que as imagens capturadas utilizadas são selecionadas para sempre terem informações redundantes, a oclusão nunca aparece em mais de uma imagem, logo o método sempre funciona. Outro ponto negativo do método é o tempo de execução, este em média é de 25 segundos por imagem processada. O método é bem diferenciado em comparação com os apresentados anteriormente, visto que consegue uma qualidade elevada nos resultados, porém ainda não é um método automatizado, no qual o usuário tem que identificar a área com o objeto a ser removido.

### 3.7 CONSIDERAÇÕES

Como evidenciado nos trabalhos analisados nenhum método se mostrou totalmente eficaz, alguns falharam no quesito desempenho, outros não são totalmente automatizados, e outros apresentavam resultados bons apenas em casos específicos. É possível dividir os trabalhos entre seus principais problemas:

- Desempenho Comprometido: (FLORES; BELONGIE, 2010), (NODARI; VANETTI; GALLO, 2012), (DUBOIS; HO, 2012);
- Intervenção Manual: (BÖHM, 2004), (HARO, 2012);
- Casos Específicos: (LEE; KIM, 2009), (DUBOIS; HO, 2012).

Em um método de processamento de imagem é interessante manter um desempenho otimizado, possibilitando a utilização do mesmo em outros trabalhos sem adicionar tempo de execução extra. Outra vantagem é a utilização em um conjuntos de dados maior do que seria possível com outros métodos, pois um método não tão otimizado poderia ser inaplicável em tais situações.

É desejável que o método seja totalmente automatizado, assim podendo ser aplicado sem a necessidade de entrada de informação manual. Com a entrada de dados manual o sistema se torna mais vulnerável a erros humanos de operação e também causando a perda do seu desempenho.

Como o objetivo deste trabalho é um funcionamento em qualquer imagem presente no GSV, métodos como o de Lee e Kim (2009) que funcionam especificamente para certos casos não são aplicáveis neste contexto. Sendo assim estes, apesar de resultados bons e desempenho otimizado, não serão utilizados.

A Tabela 1 compara as principais características encontradas em cada trabalho, evidenciando os problemas citados acima. Também foi adicionado o método criado por este trabalho que será explicado no capítulo 4, este pode ser aplicado a maioria dos objetos que geram oclusões no GSV e tem um custo computacional mais baixo, visto que para imagens com mais de um objeto oclusor o método proposto seria mais rápido que os métodos automatizados estudados. E, como o mesmo é um método automatizado poderá ser utilizado em grandes coleções de imagens sem maiores problemas.

**Tabela 1: Tabela Comparativa Entre os Trabalhos**

	MÉTODOS	AUTOMÁTICO	OBJETIVO	TEMPO	OBSERVAÇÕES
(BÖHM, 2004)	RANSAC, Seleção de Fundo	NÃO	Remover objetos grandes que obstruam as fachadas	-	-
(LEE; KIM, 2009)	Mapa de Profundidade, Segmentação por Regiões, Preenchimento de <i>Pixels</i>	SIM	Remover o Objeto Mais a Frente da Imagem	-	-
(FLORES; BELONGIE, 2010)	SIFT, RANSAC, HOG	SIM	Remover pessoas do GSV	-	Pode deixar vestígios das pessoas
(NODARI; VANETTI; GALLO, 2012)	HOG, MNOS, Inpainting	SIM	Substituir os pedestres do GSV com imagens legalmente viáveis	21 segundos por pessoa	Dificuldade com pessoas com assessórios
(HARO, 2012)	Watershed, Inpainting, SURF, RANSAC	NÃO	Remover as pessoas de imagens perante a seleção do usuário	-	-
(DUBOIS; HO, 2012)	Segmentação de Regiões, ASIFT, Inpainting	SIM	Remover objetos que não estejam presentes em todas as imagens	25 segundos por objeto	Área da remoção fica "borrada"
(Maestri, 2015)	SURF, KNN, RANSAC, Seleção de Fundo	SIM	Remover objetos em movimento das imagens do GSV	40 segundos por imagem	Pode apresentar sombras das imagens laterais

**Fonte:** Produção do próprio autor.

## 4 FUSÃO DE IMAGENS EM DIFERENTES PERSPECTIVAS PARA GERAÇÃO DE FACHADAS LIVRES DE OCLUSÃO

Com os problemas observados na tabela de comparações dos trabalhos relacionados, fica evidenciado que é um trabalho no estado da arte em aberto. Este trabalho criou um método que se utiliza de partes dos trabalhos estudados, sendo na maioria baseadas no trabalho de Böhm (2004), devido a este método ser genérico e ter um tempo de execução baixo, apesar de não ser um método automatizado. Para automatizar o método foi utilizada a abordagem de detecção de descritores para identificar os pontos de interesse para alinhar as imagens, como visto em outros trabalhos (HARO, 2012; FLORES; BELONGIE, 2010).

O método consiste em três etapas:

- Identificar os pontos chaves se utilizando do SURF juntamente com o relacionamento dos pontos utilizando KNN filtrados pela proporção da mínima distância encontrada. Foi utilizado o SURF pois segundo os estudos de Bezerra e Aguiar (2013) o SURF é o método de obtenção de descritores com o melhor custo benefício quando se trata de desempenho e quantidade de pontos obtidos em imagens estáticas;
- Fazer a transformação de perspectiva para alinhar as 3 imagens, isso se utilizando do RANSAC para o ajuste da transformação. São utilizadas 3 imagens devido a ser a quantidade de imagens diferentes que podem ser obtidas de uma mesma cena no GSV, o RANSAC foi utilizado por ser um método rápido e que aceita vários *outliers* na entrada de dados sem comprometer o resultado final, pois mesmo com a filtragem feita na primeira etapa alguns *outliers* podem passar esta etapa;
- Utilizar uma versão modificada do método de escolha de fundo proposto por Böhm (2004) para retirar o resultado da imagem final sem oclusões. Os testes iniciais utilizando o método de Böhm (2004) não obtiveram os resultados esperados, para melhoramento do mesmo foi adicionado um esquema de comparação de vizinhanças.

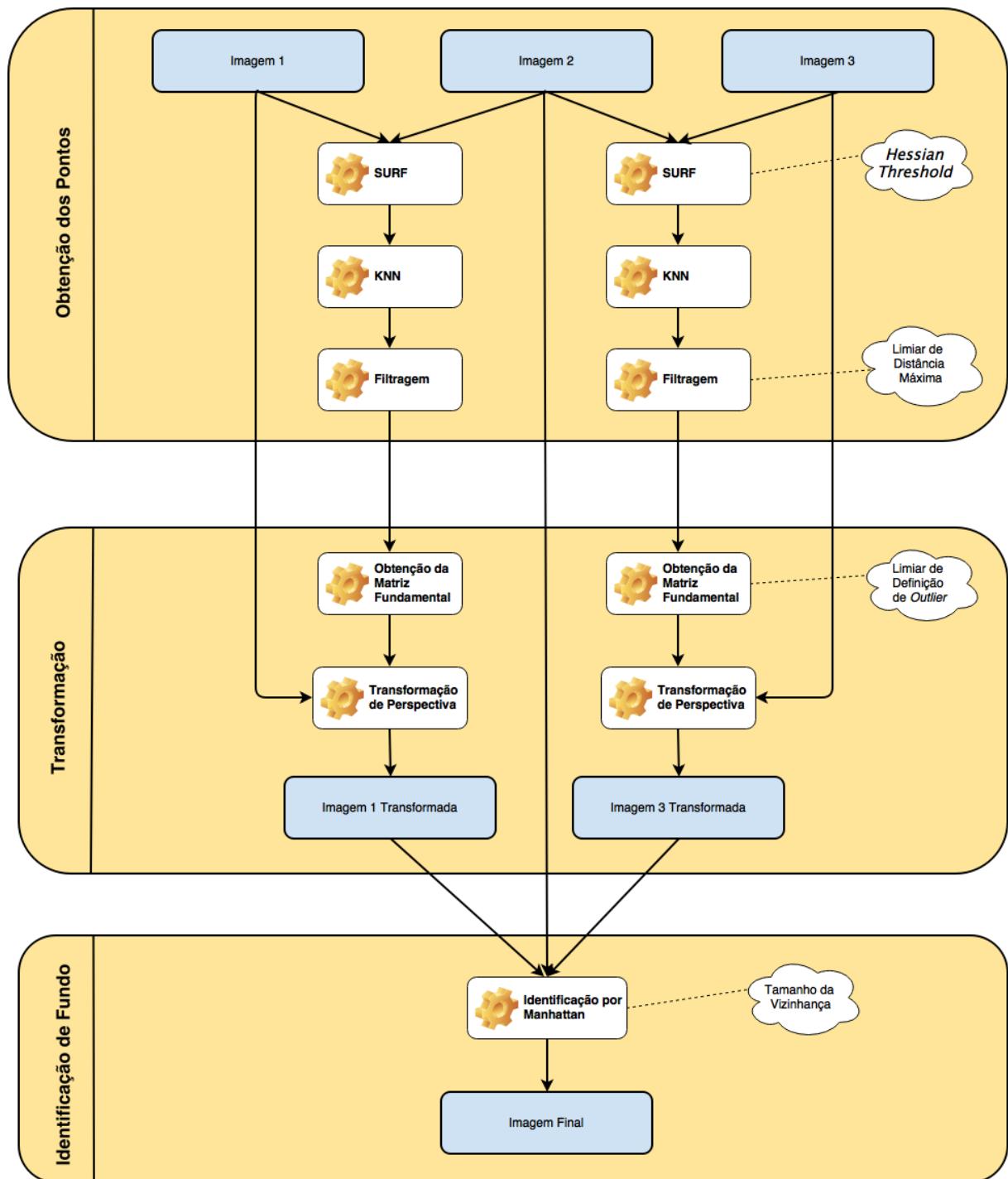
Como ilustra a Figura 29, estas 3 etapas podem ser divididas em vários métodos. Na etapa de detecção dos pontos é executado o algoritmo do SURF nas 3 imagens originais de entrada, resultando em vários pontos relacionados identificados em cada uma das imagens, então é aplicado o KNN entre duas imagens de cada vez, inicialmente a imagem 1 com a imagem 2 e depois com a imagem 2 e a imagem

3. Este processo permite obter os pontos relacionados entre duas imagens. Após identificados, os pontos são filtrados através de parâmetro de proporção que será detalhado posteriormente.

Na etapa de transformação, são utilizados os pontos relacionados entre as imagens para gerar a matriz de transformação. Isto é possível com o método de obtenção de homografia, que retorna uma matriz de transformação baseada nos pontos coincidentes das imagens. Depois de calculadas as matrizes fundamentais de cada par de imagens, essas matrizes são utilizadas para realizar a projeção das imagens dos cantos para a central, assim fazendo o alinhamento das 3 imagens.

Na etapa de identificação de fundo, as 3 imagens alinhadas são percorridas *pixel* à *pixel* paralelamente, avaliando os valores RGB em sua vizinhança, assim possibilitando a comparação entre as três imagens. Com a diferença da proporção entre as vizinhanças dos *pixels* das imagens são definidas quais são mais próximas através da distância Manhattan, de maneira que as vizinhanças com valores próximos tem seu *pixel* utilizado para a criação da imagem final. Devido ao objeto que estava gerando a oclusão em uma das imagens ter se movido, provavelmente nas outras duas imagens o objeto aparece em outro lugar, levando em consideração e que o objeto não se move na mesma velocidade do "carro do GSV". O método utilizado não faz a detecção dos objetos a serem removidos para melhorar o desempenho do mesmo, visto que um método de identificação de diferenças entre imagens é mais rápido que um método de detecção de pessoas ou objetos. As imagens que tem os *pixels* mais similares são as imagens sem o objeto, portanto ao fim do processamento a imagem final não terá o objeto oclusor.

**Figura 29: Fluxograma do Método Proposto.**



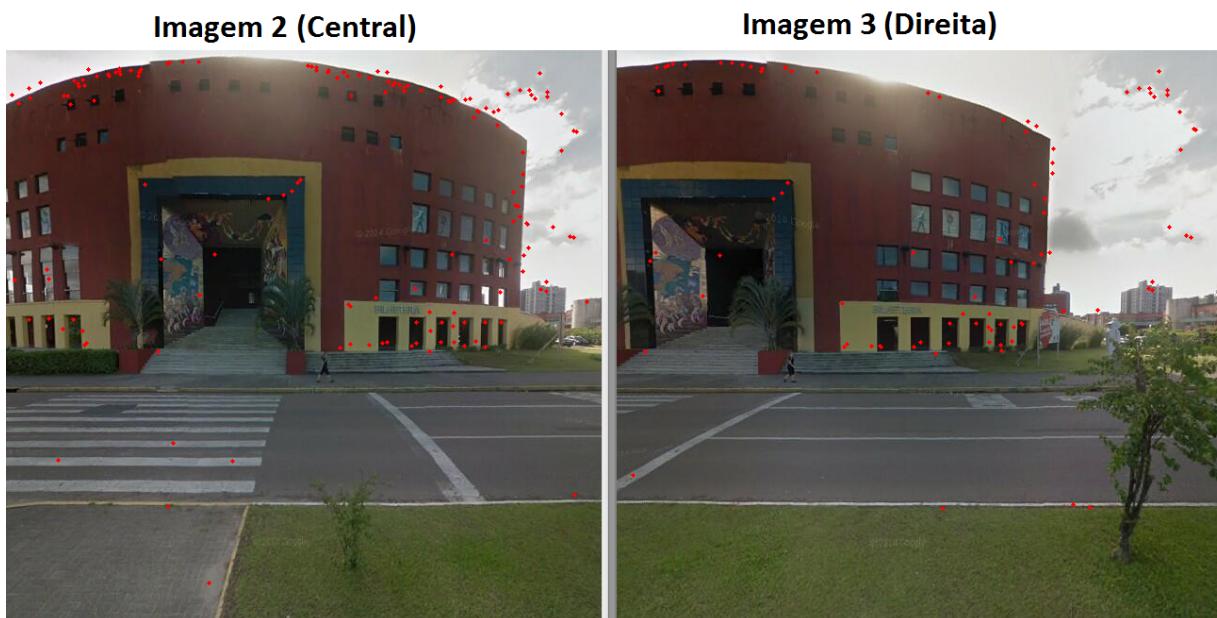
**Fonte:** Produção do próprio autor.

As próximas seções detalham o funcionamento do método, cada etapa do seu funcionamento, assim como resultados parciais e parâmetros utilizados para o ajuste do funcionamento do método.

#### 4.1 OBTENÇÃO DOS PONTOS

Inicialmente o método agrupa as imagens em 2 pares, onde as imagens laterais são relacionadas à imagem central. Para relacionar estas duas imagens, são detectados os descritores de cada uma das imagens se utilizando do método SURF, como ilustra a Figura 30. No algoritmo de SURF é necessário ajustar um parâmetro, o *hessian threshold* este limita a identificação dos pontos: apenas os pontos com maior força serão selecionados, de acordo com o limiar utilizado. Neste trabalho devido a testes empíricos e baseando-se na documentação do próprio OpenCV, foi determinado um parâmetro de em média 700 de limiar, este valor se mostrou mais eficiente nas imagens utilizadas no trabalho, visto que o mesmo proporcionou pontos com características fortes, mas em quantidade suficiente para um boa filtragem posterior.

**Figura 30: Detecção dos Pontos com SURF.**



**Fonte:** Produção do próprio autor.

Com os vetores descritores das duas imagens é possível a comparação entre os pontos identificados, geralmente, como visto em alguns trabalhos. É utilizado o método KNN para a identificação do ponto equivalente na outra imagem. Para a identificação dos pontos equivalentes ser possível, o KNN é inicializado com o vetor de descritores de uma imagem então, para cada ponto detectado da outra imagem,

é realizada uma execução do KNN com vizinhança igual a 1 ou 1NN. Devido a vizinhança de tamanho 1 no KNN, a vizinhança mais parecida na coleção de dados da outra imagem é o ponto da coleção mais parecido com o ponto analisado.

Após a execução do KNN para todos os pontos detectados, se obtêm um vetor de distâncias, relacionando os pontos das duas imagens. Para um melhor funcionamento da transformação de perspectiva, é necessária uma filtragem nos pontos retornados pelo KNN pois o método KNN pode retornar falsos positivos na detecção dos pontos equivalentes, na Figura 31 alguns dos falsos positivos estão circulados em vermelho.

**Figura 31: Falsos Positivos do KNN.**



**Fonte:** Produção do próprio autor.

Na filtragem dos pontos, se baseando nos trabalhos estudados foi utilizada uma fórmula, na qual é obtido o menor valor do vetor de distâncias obtidos do KNN, e com este valor é feita uma comparação com todos os outros valores do vetor, se o valor sendo analisado no momento for quatro vezes maior que o menor valor do vetor (Limiar de Distância Máxima), então este ponto é considerado um falso positivo, e é então retirado do vetor de equivalência. O valor 4 utilizado como parâmetro para este método foi definido baseando-se na documentação do OpenCV, na seção *Features2D + Homography*, e em testes empíricos realizados nas imagens selecionadas para a execução do método. Como é visível na Figura 32, quase todos os falsos positivos foram eliminados, porém juntamente com a filtragem dos falsos positivos foram eliminados muitos pontos válidos. Este fenômeno apesar de ser indesejável não será prejudicial para o método, visto que com 4 pontos corretos o método consegue relacionar as duas imagens através de uma transformação de

perspectiva, porém se alguns falsos positivos passarem para a próxima etapa, se a quantidade de pontos for maior o método de obtenção da matriz fundamental faz um filtro final baseado na própria transformação, com a ajuda de uma limiar passada por parâmetro.

**Figura 32: Falsos Positivos Filtrados.**



**Fonte:** Produção do próprio autor.

## 4.2 TRANSFORMAÇÃO DE PERSPECTIVA

Primeiramente para a execução da transformação de perspectiva é necessário o cálculo da matriz fundamental para cada um dos pares de imagem, para isso foi utilizado o método do OpenCV que se utiliza do algoritmo de RANSAC, para encontrar a matriz fundamental baseando-se no vetor de pontos filtrados. O algoritmo para a obtenção da transformação executa o algoritmo RANSAC duas vezes, a primeira vez levando em consideração todos os pontos do vetor passado como parâmetro, isso para a criação da matriz fundamental. Depois de calculada a matriz ele filtra os pontos de entrada baseando-se na inequação  $|PM2_i - (PM1 * M)| > limiar$ , onde  $PM2$  é um ponto da imagem 2,  $PM1$  é um ponto da imagem 1 e  $M$  é a matriz fundamental calculada anteriormente, assim eliminando os pontos que não se encaixaram na transformação. A  $limiar$  é passada por parâmetro na função. Essa limiar consiste na quantidade de *pixels* que se considera tolerante no processo, neste trabalho foram utilizados 5 *pixels* de tolerância, esse valor foi baseado na documentação do próprio OpenCV, na seção *Features2D + Homography*.

Após o cálculo da matriz fundamental é possível realizar a transformação entre as imagens, para cada par de imagens foi gerada uma matriz fundamental,

esta é utilizada para realizar a transformação das imagens laterais para serem alinhadas com a imagem central. Este processo foi feito através de um método do OpenCV que realiza a multiplicação da matriz fundamental pelos *pixels* da imagem passada por parâmetro, o método também preenche os espaços vazios deixados pela transformação com a cor preta, deixando a imagem do tamanho da imagem central. O resultado da transformação pode ser visualizada na Figura 33.

**Figura 33: Imagens Antes e Depois da Transformação de Perspectiva**

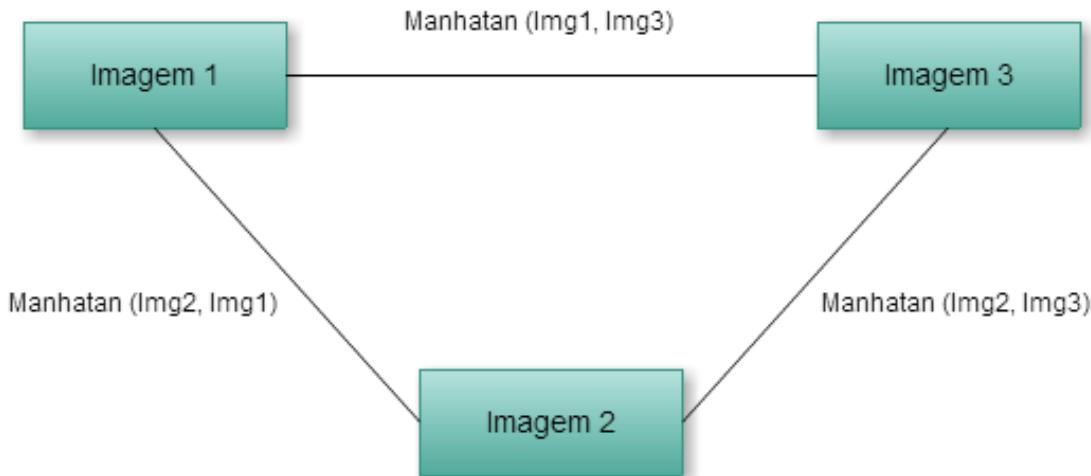


**Fonte:** Produção do próprio autor.

#### 4.3 IDENTIFICAÇÃO DE FUNDO

A identificação de fundo é feita através de uma abordagem *pixel à pixel* de checagem de vizinhança, onde para cada *pixel* percorrido na imagem central é comparada a semelhança entre os *pixels* da vizinhança de 24 *pixels* de cada uma das imagens. O valor de 24 *pixels* de vizinhanças foi obtido analisando os resultados e o desempenho do método. Para essa comparação ser realizada, primeiro são geradas as distâncias Manhatam seguindo o esquema da Figura 34, para cada um dos *pixels* da vizinhança analisada.

**Figura 34: Esquema de Comparação de Distâncias.**



**Fonte:** Produção do próprio autor.

Poderiam ser utilizadas outros tipos de equações de distância, como a Euclidiana, porém devido a motivos de desempenho e simplicidade se optou pela Manhattan. Para determinar qual o *pixel* que será utilizado na imagem final, é preciso comparar a vizinhança dos *pixels* das três imagens, essa comparação é realizada com o intuito de identificar os 2 *pixels* sem o objeto de oclusão para cada *pixel* da vizinhança. A comparação é dada pela Inequação 4, na qual, se a soma das distâncias entre as duas imagens laterais e a imagem central, for maior que duas vezes a distância entre a soma imagem 1 e a imagem 3, isso levando em consideração uma vizinhança de 24 *pixels*, significa que a imagem central contém o objeto, pois a mesma será diferente das imagens laterais na maioria da sua vizinhança. Verificado que a imagem central contém o *pixel* do objeto, o mesmo terá que ser trocado pelo *pixel* da primeira imagem.

$$\text{Manhattan}(\text{Img1}, \text{Img2}) + \text{Manhattan}(\text{Img3}, \text{Img2}) > 2 * \text{Manhattan}(\text{Img1}, \text{Img3}) \quad (4)$$

Aqui poderia ter sido escolhido o *pixel* da imagem 3 pois os dois tem um valor semelhante segundo os valores de distância das vizinhanças. Caso a condição não seja verdadeira a imagem central mantém o seu próprio *pixel*, esse processo gera a imagem final do método de remoção de objetos de oclusão, como ilustra a Figura 35.

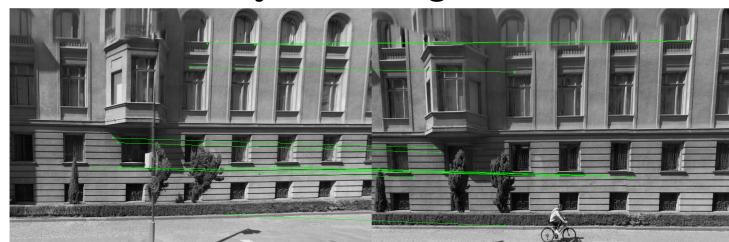
## 5 RESULTADOS

Para a obtenção dos resultados, foram utilizadas imagens de fachadas de casas e estabelecimentos comerciais, estas tiradas manualmente do GSV. Para cada caso de teste se teve como entrada 3 imagens, a da esquerda da imagem que contém o objeto a ser removido, a própria imagem contendo o objeto a ser removido e a imagem da direita a ser removido. Os casos de testes serão mostrados no formato antes e depois da imagem central ser processada, visto que é a imagem que contém o objeto a ser removido, apenas o Resultado Berlin será detalhado nos passos do método.

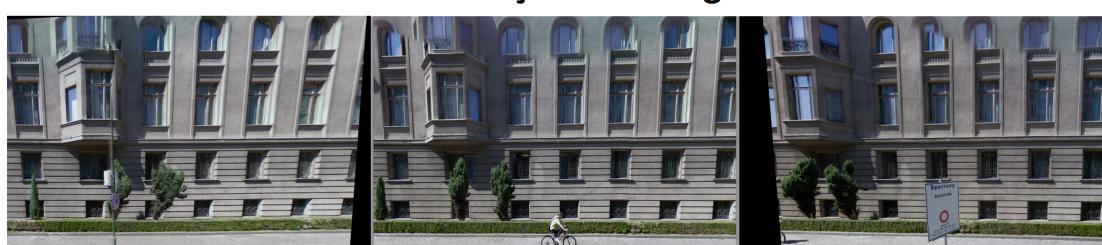
**Figura 35: Prédio em Berlin.  
Imagens Originais**



**Identificação e Filtragem de Pontos**



**Transformação das Imagens**



**Resultado Final**



**Fonte:** Produção do próprio autor.

**Figura 36: Obra em Andamento.**



**Fonte:** Produção do próprio autor.

**Figura 37: Banco do Bradesco.**



**Fonte:** Produção do próprio autor.

**Figura 38: Supermercado Angeloni.**



**Fonte:** Produção do próprio autor.

**Figura 39: Eletrônica.**



**Fonte:** Produção do próprio autor.

## 5.1 DISCUSSÃO

Como ilustrado nos resultados mostrados acima, o algoritmo não se comportou como o esperado na maioria das vezes devido ao sistema de identificação de fundo. Também houve problemas na identificação dos pontos de interesse com o SURF no início do processo em algumas imagens, como ilustra a Figura 40.

A maior dificuldade encontrada pelo algoritmo foi a identificação de fundo, como visto, as imagens resultantes ficam com borrões e sombras das imagens laterais. Onde a escolha errada dos *pixels* a ser utilizados causa inconsistência na continuidade da imagem final, gerando um efeito borrado na imagem, como ilustra a Figura 38. Muitas vezes o método também acaba por colocar sombras dos objetos encontrados nas imagens laterais, também ocorrendo devido ao mal julgamento do método, como ilustra a Figura 39.

Outra dificuldade foi na identificação de pontos, o SURF necessita de apenas um limiar e uma imagem como parâmetro de entrada para a detecção de pontos de interesse no mesmo, porém devido as imagens do GSV variarem muito de contexto, luminosidade, cores e formas, é difícil estipular um parâmetro geral para o SURF. Com a atribuição de um parâmetro geral é gerado um comportamento não desejável no método, onde em algumas imagens analisadas há poucos pontos reconhecidos, ou, até mesmo, não atingirem o mínimo de 4 pontos para a transformação, assim gerando distorção na imagem transformada, não alinhando as imagens laterais com a imagem central, como ilustra a Figura 40.

Como o alinhamento não ocorre de maneira correta, é causada uma falha pior ainda na identificação de fundo, na qual, em piores casos, toda a imagem fica

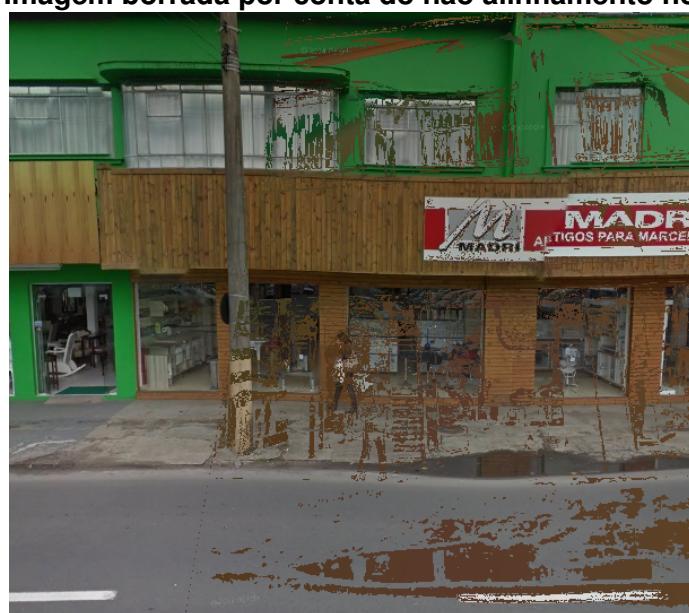
contaminada com *pixels* referentes as imagens laterais não alinhadas, como ilustra a Figura 41.

**Figura 40: Transformação errada por falta de pontos.**



**Fonte:** Produção do próprio autor.

**Figura 41: Imagem borrada por conta do não alinhamento no método.**



**Fonte:** Produção do próprio autor.

## 6 CONCLUSÕES

Cada vez mais o uso dos sistemas de visualização panorâmicas vem crescendo. E um exemplo desse processo é o próprio Brasil, no qual em pouco tempo passou de poucas cidades que suportavam o GSV, para praticamente todas as cidades do Brasil. Os problemas de privacidade enfrentados pelos sistemas de visualização panorâmica como o GSV, podem ser resolvidos retirando os elementos passíveis de identificação das imagens. Tal assunto como visto anteriormente já foi amplamente utilizado para justificar outros trabalhos citados.

Com os trabalhos relacionados, foi possível obter um panorama sobre o assunto de remoção de objetos em imagens. Os trabalhos em sua maioria se utilizavam de algoritmos que detectam as pessoas nas imagens para retira-las, porém este método apenas servia para a remoção de pessoas e os carros continuavam na imagem. Outra abordagem utilizada por tais trabalhos era a de remoção dos objetos que se movem, fazendo o alinhamento das imagens em multi-perspectiva e subtraindo umas das outras para a obtenção do fundo.

Os resultados obtidos por alguns trabalhos analisados foram imagens com a remoção completa dos objetos desejados, porém alguns dos trabalhos não propõem um método totalmente automatizado, logo seria trabalhosa a utilização do método em uma coleção relativamente grande de imagens do GSV. Geralmente estes trabalhos utilizavam da abordagem de alinhamento da imagem e subtração para obtenção de fundo, pois os pontos para o alinhamento geralmente são definidos manualmente.

O método desenvolvido nesse trabalho eliminou a interferência humana na identificação dos fundos das imagens do GSV. Uma vez capturadas as imagens elas são processadas pelo método automaticamente. Com isso este método se destaca em relação aos demais métodos não automatizados estudados.

Outro problema resolvido pelo método desenvolvido foi o tempo elevado de processamento, onde na maioria dos trabalhos estudados foram de em média 25 segundos por objeto removido da imagem, mesmo sem otimizações de código, visto que o método proposto foi desenvolvido em uma linguagem interpretada com processos iterativos, esse tempo foi reduzido, se levado em consideração uma imagem com mais de um objeto a ser removido, para em média 40 segundos por imagem. Isso foi possível devido a utilização de um método de subtração de imagens para a remoção dos objetos, geralmente utilizada nos métodos manuais, ao invés de uma abordagem de detecção e remoção de objetos utilizada pelos métodos automáticos dos trabalhos estudados.

A qualidade do resultados obtidos por outro lado, foi pior do que o esperado, visto que os outros trabalhos acabaram tendo resultados qualitativos melhores do que o método desenvolvido. Foi constatado que é possível criar um método automatizado que tenha um desempenho melhor que os trabalhos que utilizam a detecção de objetos para a remoção do mesmo, porém sem identificar a área ocupada pelo objeto fica difícil a remoção completa do mesmo, fazendo com que a qualidade da imagem final seja comprometida.

## 6.1 DISCUSSÃO

A principal causa da qualidade comprometida é o método que foi utilizado para a identificação de fundo, que mesmo levando em consideração uma vizinhança de 24 *pixels* para a comparação entre as imagens não consegue eliminar completamente o objeto, devido a não ser feita uma detecção da área do mesmo, assim gerando descontinuidade na imagem processada. Também dependendo do alinhamento das imagens alguns objetos com cores parecidas podem acabar se alinhando, trazendo os mesmos para a imagem final.

Outra causa da qualidade comprometida foi a escolha de um limiar fixo para a identificação de pontos de interesse pelo SURF, assim tirando flexibilidade do método para vários tipos de imagens, onde dependendo das características da imagem em questão poderiam ser reconhecidos vários pontos ou pontos insuficientes para uma transformação de perspectiva adequada.

## 6.2 TRABALHOS FUTUROS

Para trabalhos futuros na melhoria do método, poderiam ser feitas alterações no modo como ocorre a identificação de fundo, e alterações no modo como é definido o parâmetro do SURF. Para a melhoria da identificação de fundo, antes de feita a escolha do *pixel* que seria utilizado na imagem final seria criada uma máscara com a utilização de morfologia matemática. Para isso ser possível poderia ser feito o fechamento dos *pixel* escolhidos na identificação de fundo, com isso seriam eliminados os pontos de ruídos que tem uma área pequena, deixando apenas os pontos pertencentes aos objetos mais conexos, depois seria aplicada uma dilatação a essa máscara, isso para a criação de regiões maiores para serem colocadas na imagens final. Depois essa máscara seria utilizada para a criação da imagem final, onde a área contida na máscara viria de uma das imagens laterais e o resto da imagem final viria da imagem

central.

Para resolver o problema do SURF poderia ser empregado um método de análise de histograma, onde dependendo das características do histograma da imagem em questão seria utilizado um limiar específico, com essa adaptação as chances de se obter um número suficiente de pontos para todas as imagens são melhores do que no sistema atual.

## REFERÊNCIAS

- BEZERRA, K. V. S.; AGUIAR, E. de. Casamento de padrões em imagens e vídeos usando características de imagens. Universidade Federal do Espírito Santo, São Mateus, ES, 2013.
- BUCAR, D. Controle temporal de dados: o direito ao esquecimento. *O Civilista*, Universidade Estadual do Rio de Janeiro, v. 2, n. 3, p. 2, 2013.
- BÖHM, J. Multi-image fusion for occlusion-free façade texturing. *International Society for Photogrammetry and Remote Sensing Congress*, University Stuttgart, p. 867, 2004.
- CAI, J.-F.; CHAN, R. H.; SHEN, Z. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, v. 24, n. 2, p. 131 – 149, 2008. ISSN 1063-5203. Special Issue on Mathematical Imaging – Part {II}.
- CRIMINISI, P. P. A.; TOYAMA, K. Object removal by exemplar-based inpainting. Microsoft Research Ltd, Cambridge, UK, 2003.
- CUNNINGHAM, P.; DELANY, S. J. *k-Nearest Neighbour Classifiers*. [S.I.]: University College Dublin, 2007.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2005. (CVPR '05), p. 886–893. ISBN 0-7695-2372-2.
- DUBOIS, R. M. D. E.; HO, T. H. Object removal in multi-view images using disparity layering and piecewise homography. *Proceedings of the International Conference on Electrical and Computer Systems*, p. 191, 2012.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782.
- FLORES, A.; BELONGIE, S. Removing pedestrians from google street view images. In: *IEEE International Workshop on Mobile Vision*. San Francisco, CA: [s.n.], 2010.
- GONZALES, G.; LUIS, G.; MEGGIOLARO, M. A. Aplicação da técnica sift para determinação de campos de deformações de matérias usando visão computacional. *Maxwell*, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, p. 49, 2010.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0201180758.
- HARO, A. User-guided pedestrian and object removal. In: *User-Centered Computer Vision (UCCV), 2012 1st IEEE Workshop on*. [S.I.: s.n.], 2012. p. 50–55.
- JANSSEN, C. Taxicab geometry: Not the shortest ride across town (exploring conics with a non-euclidean metric). Iowa State University, Iowa, US, 2007.

- KIM, D.-K.; JANG, B.-T.; HWANG, C.-J. A planar perspective image matching using point correspondences and rectangle-to-quadrilateral mapping. In: *Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*. [S.I.: s.n.], 2002. (SSIAI '02), p. 87–. ISBN 0-7695-1537-1.
- KOZAC, A. Google street view and privacy concerns. *WPiA UW*, Poland, UW, 2014.
- LEE, J.; KIM, C. *Removing foreground objects by using depth information from multi-view images*. [S.I.]: Proc. SPIE, 2009. 669627-669627-9 p.
- LINDEBERG, T. Scale invariant feature transform. KTH Royal Institute of Technology, Stockholm, Sweden, 2012.
- LORENA, A. As consequências jurídicas e civis do google street view. *Conteúdo Jurídico*, Brasilia, DF, 2011.
- LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision - Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999. (ICCV '99), p. 1150–. ISBN 0-7695-0164-8.
- NODARI, A.; VANETTI, M.; GALLO, I. Digital privacy: Replacing pedestrians from google street view images. In: *Pattern Recognition (ICPR), 2012 21st International Conference on*. [S.I.: s.n.], 2012. p. 2889–2893. ISSN 1051-4651.
- OPENCV. *Introduction to SURF (Speeded-Up Robust Features)*. OPENCV, 2014. Disponível em: <<http://docs.opencv.org/doc/tutorials/tutorials.html>>.
- PALHARES, A. V. de A. Implementação do ransac com algoritmo de eight-point em gpu. Universidade Federal de Pernambuco, Pernambuco Brasil, 2009.
- RAKOWER, L. Blurred line: Zooming in on google street view and the global right to privacy. *Brooklyn Journal of International Law*, US, p. 317, 2011.
- SANTOS, D. M.; VEIGA, A. Tabela de quantização para imagens coloridas (rgb) usando algoritmos genéticos. *Revista de Ciência e Tecnologia do Vale do Mucuri*, Uberlândia, p. 63–76, 2010.
- THOMAS, J.; KAREEM, A.; BOWYER, K. Fast robust perspective transform estimation for automatic image registration in disaster response applications. In: *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*. [S.I.: s.n.], 2012. p. 2190–2193. ISSN 2153-6996.