

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO

EDUARDO CAVICHIOLI

**Uma Aplicação de CBIR para  
Identificação de Pinturas**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em  
Engenharia da Computação

Orientador: Prof. Dr. Marcelo Walter

Porto Alegre  
2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Engenharia de Computação: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“You cannot teach a man anything,  
you can only help him discover it in himself.”*

— GALILEO GALILEI

## **AGRADECIMENTOS**

Primeiramente, gostaria de agradecer aos meus pais, Wilson Cavichioli Junior e Siliane Ambrosi Cavichioli, que sempre me incentivaram a buscar os meus sonhos e apoiaram as minhas escolhas. Espero poder retribuir um dia tudo o que fizeram para me ajudar.

Gostaria de agradecer aos meus amigos, que sempre me ajudaram quando possível durante este período acadêmico. Isto para as amizades que fiz durante o curso, os amigos de longa data e também para os que não tenho mais contato, mas que tiveram importância neste período.

Também gostaria de agradecer meu orientador Prof. Marcelo Walter, por toda a atenção e dedicação durante este período.

Por fim, gostaria de agradecer ao Instituto de Informática e à Universidade Federal do Rio Grande do Sul pelo ensino de qualidade que recebi ao longo do curso, além das diversas oportunidades proporcionadas.

## RESUMO

Este projeto tem como objetivo desenvolver uma aplicação de recuperação de imagens por conteúdo (CBIR - *Content-Based Image Retrieval*) focada em identificar obras de arte através de imagens das mesmas. O foco da aplicação são pinturas, onde capturada uma imagem da obra, potencialmente em condições não ideais e com dispositivos móveis, utilizam-se técnicas de processamento de imagem para compará-la com uma base de dados de pinturas, e então apresentar a imagem que mais se aproxima da captura. Além disso, são comparadas diferentes técnicas utilizadas na filtragem dos resultados durante a recuperação de Pinturas. Os testes experimentais demonstram que é possível realizar a recuperação de pinturas através de fotos utilizando as técnicas descritas, chegando a uma conclusão de qual delas é mais apropriada de acordo com a situação.

**Palavras-chave:** CBIR. Processamento de Imagens. Pinturas. SURF.

## **An CBIR Application for Paintings Identification**

### **ABSTRACT**

The main goal of this project is the development of an application of Content-Based Image Retrieval (CBIR) capable of identifying works of art using images of the same. The focus of the application are paintings. From an image of a painting, taken potentially in non-ideal conditions and with mobile devices, we want to find the closest image in a database of paintings, using image processing techniques. Also, some algorithms used to filter results during painting retrieval process are compared. Our experimental results show that is possible to retrieve paintings from pictures using the proposed techniques, and concluding with which technique is more suitable according to the situation.

**Keywords:** CBIR, Computer Vision, Paintings, SURF.

## **LISTA DE ABREVIATURAS E SIGLAS**

CBIR	Content-Based Image Retrieval
DoG	Difference-of-Gaussians
IDE	Integrated Development Environment
RANSAC	Random Sampling Consensus
SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Feature

## LISTA DE FIGURAS

<p>Figura 1.1 Demonstração do pareamento dos pontos de interesse. A imagem da esquerda representa a imagem de entrada da aplicação. A imagem da direita representa uma imagem do banco de imagens onde será realizada a busca. Os círculos representam os pontos de interesse detectados. As linhas representam o pareamento dos pontos através da análise dos seus descritores.....</p>	14
<p>Figura 2.1 Algoritmo de detecção DoG .....</p>	19
<p>Figura 2.2 Vizinhos analisados na técnica DoG. Na camada central o pixel analisado está marcado com um "X", e seus 8 vizinhos de espaço são destacados por círculos na mesma camada. Nas camadas superior e inferior é possível observar os 18 vizinhos de escala, também marcados por círculos, sendo a primeira em escala maior e a última em escala menor. ....</p>	20
<p>Figura 2.3 À esquerda estão as imagens de treinamento para os dois objetos de busca. Como pode ser visto na imagem da direita, os objetos são reconhecidos mesmo com grande parte obstruída atrás de outros itens presentes na imagem. Um paralelogramo é desenhado em volta de cada objeto reconhecido para representar as fronteiras da imagem de treinamento. Os quadrados menores representam os pontos-chaves utilizados para o reconhecimento.....</p>	21
<p>Figura 2.4 À esquerda estão as derivadas parciais de segunda ordem da Gaussiana nas direções <math>y</math> e <math>xy</math>. À direita estão as aproximações das mesmas direções utilizando os box-filters. As áreas em cinza são iguais a zero.....</p>	22
<p>Figura 2.5 Na esquerda é demonstrada a técnica que reduz iterativamente o tamanho da imagem. Na direita está representado o uso de imagens integrais para aumentar o tamanho do filtro, com um custo constante. ....</p>	23
<p>Figura 2.6 Pontos de interesse detectados em uma imagem de um campo de girassóis. Este tipo de imagem mostra a natureza das características obtidas utilizando detectores baseados na matriz Hessiana. O tamanho do círculo representa a escala do filtro utilizado.....</p>	24
<p>Figura 2.7 Definição da Orientação - À esquerda está representado a região circular de raio <math>6s</math>. À direita estão representados os filtros da transformada de Haar com tamanho <math>4s</math>.....</p>	24
<p>Figura 2.8 Definição da Orientação - Uma janela deslizante de tamanho <math>\pi/3</math> detecta a orientação dominante das respostas da transformada de Haar carregadas com um Gaussiano em cada ponto de amostragem dentro de uma vizinhança circular que envolve o ponto de interesse.....</p>	25
<p>Figura 2.9 Região Quadrada - Ilustração da região quadrada de tamanho <math>20s</math>, com os pontos de amostragem espaçados em <math>5 \times 5</math> da transformada de Haar. ....</p>	26
<p>Figura 2.10 Cada quadrado marcado na imagem representa uma região identificada como ponto-chave a ser analisada. A dimensão de cada quadrado é resultante da escala na qual a região foi detectada.....</p>	27
<p>Figura 2.11 Na esquerda, região quadrada construída na orientação pré-determinada, sub-dividida em <math>4 \times 4</math> sub-regiões quadradas, centrada no ponto de interesse. Na direita, as sub-divisões <math>2 \times 2</math> de cada sub-região. Cada quadrado corresponde a um campo do vetor de descrição. Essa é a soma das transformadas de Haar e seus valores absolutos, de acordo com a orientação do vetor pré-calculado.....</p>	28

Figura 3.1	Visualização dos pontos de interesse detectados em uma pintura digitalizada. Eles estão representados em cores diferentes apenas para sua distinção....	31
Figura 3.2	Fluxograma simplificado do algoritmo de população do banco de dados.....	32
Figura 3.3	Fluxograma do processo.....	34
Figura 3.4	Pareador de Força Bruta - Pareador utilizando apenas o Passo 2, obtendo 2281 correspondências.....	35
Figura 3.5	Fluxograma simplificado do algoritmo do teste de proporção. ....	36
Figura 3.6	Teste de Proporção - Pareador até o Passo 3, reduzindo para 41 correspondências com proporção de 65%. ....	37
Figura 3.7	Fluxograma simplificado do algoritmo do teste de simetria.....	38
Figura 3.8	Teste de Simetria - Pareador até o Passo 4, reduzindo para 21 correspondências. ....	39
Figura 3.9	Geometria Epipolar.....	40
Figura 3.10	Fluxograma simplificado do algoritmo do teste de geometria epipolar.....	41
Figura 3.11	Teste de Geometria Epipolar - Pareador ao final do Passo 5, reduzindo para 14 correspondências.....	42
Figura 4.1	Imagem 1 - Mona Lisa - Fotografia com pequeno ângulo, reflexo, informações desnecessárias e com resolução 427x640. ....	43
Figura 4.2	Imagem 2 - Mona Lisa - Fotografia com ângulo, afastada, muita informação desnecessária e resolução 1600x900. ....	44
Figura 4.3	Imagem 3 - Mona Lisa - Fotografia direta, relativamente próxima, com obstruções, informações desnecessárias e resolução 2048x1024. ....	44
Figura 4.4	Imagem 4 - Quarto em Arles - Fotografia direta, com resolução 1024x874. ....	44
Figura 4.5	Imagem 5 - Vaso com quinze girassóis - Fotografia direta, relativamente próxima, com obstruções, informações desnecessárias e resolução 810x1024.....	45
Figura 4.6	Imagem 6 - Vaso com quinze girassóis - Fotografia direta, próxima, com obstruções, informações desnecessárias e resolução 1024x768. ....	45
Figura 4.7	Imagem 7 - O Nascimento de Vênus - Fotografia com ângulo, distância média, informações desnecessárias e resolução 1024x765. ....	46
Figura 4.8	Imagem 8 - O Nascimento de Vênus - Fotografia com ângulo, próxima, contém reflexos e resolução 1024x768.....	46
Figura 4.9	Imagem 9 - O Nascimento de Vênus - Fotografia direta, próxima, informações desnecessárias e resolução 1024x683.....	46
Figura 4.10	Teste de Proporção - Imagem 7 com proporção 35%.....	48
Figura 4.11	Teste de Proporção - Imagem 9 com proporção 35%.....	48
Figura 4.12	Teste de Proporção - Imagem 1 com proporção 65%.....	49
Figura 4.13	Teste de Proporção - Imagem 7 com proporção 65%.....	49
Figura 4.14	Teste de Proporção - Imagem 9 com proporção 65%.....	49
Figura 4.15	Teste de Proporção - Imagem 4 com proporção 95%.....	50
Figura 4.16	Teste de Simetria - Imagem 7 com proporção 35%.....	51
Figura 4.17	Teste de Simetria - Imagem 9 com proporção 35%.....	51
Figura 4.18	Teste de Simetria - Imagem 7 com proporção 65%.....	52
Figura 4.19	Teste de Simetria - Imagem 9 com proporção 65%.....	52
Figura 4.20	Teste de Simetria - Imagem 4 com proporção 95%.....	52
Figura 4.21	Teste de Geometria Epipolar - Imagem 9 com proporção 35%, distância 1, e confiança 0,69.....	54
Figura 4.22	Teste de Geometria Epipolar - Imagem 7 com proporção 65%, distância 1, e confiança 0,69.....	55
Figura 4.23	Teste de Geometria Epipolar - Imagem 9 com proporção 65%, distância 1, e confiança 0,69.....	56

Figura 4.24 Teste de Geometria Epipolar - Imagem 7 com proporção 95%, distância 1, e confiança 0,69.....	57
Figura 4.25 Teste de Geometria Epipolar - Imagem 9 com proporção 95%, distância 1, e confiança 0,69.....	57

## LISTA DE TABELAS

Tabela 2.1	Comparação entre SIFT e SURF. ....	26
Tabela 4.1	Teste de Proporção - Resultados.....	47
Tabela 4.2	Teste de Simetria - Resultados.....	50
Tabela 4.3	Teste de Geometria Epipolar - Resultados com proporção 35% .....	54
Tabela 4.4	Teste de Geometria Epipolar - Resultados com proporção 65% .....	55
Tabela 4.5	Teste de Geometria Epipolar - Resultados com proporção 95% .....	56

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
<b>1.1 Objetivo</b> .....	<b>13</b>
<b>1.2 Aplicações</b> .....	<b>15</b>
<b>1.3 Justificativa</b> .....	<b>15</b>
<b>1.4 Organização do Trabalho</b> .....	<b>16</b>
<b>2 TRABALHOS RELACIONADOS</b> .....	<b>17</b>
<b>2.1 Histórico</b> .....	<b>17</b>
2.1.1 Visão Geral de Trabalhos de CBIR até 2000 .....	17
<b>2.2 Tipos de Características da Imagem</b> .....	<b>18</b>
2.2.1 Características Globais.....	18
2.2.2 Características Locais .....	18
<b>2.3 Técnicas de Identificação de Imagens</b> .....	<b>18</b>
2.3.1 Diferença de Gaussianas .....	19
2.3.2 <i>Scale Invariant Feature Transform</i> (SIFT) .....	20
2.3.3 <i>Speeded Up Robust Features</i> (SURF).....	21
2.3.3.1 Detector.....	21
2.3.3.2 Descritor.....	23
2.3.4 Comparativo entre as Técnicas .....	26
<b>3 SISTEMA DESENVOLVIDO</b> .....	<b>29</b>
<b>3.1 Ferramentas e Tecnologias</b> .....	<b>29</b>
3.1.1 Linguagem .....	29
3.1.2 OpenCV .....	30
3.1.3 Ambiente de Desenvolvimento .....	30
<b>3.2 Populando o Banco de Dados</b> .....	<b>31</b>
<b>3.3 Pareador de Imagens</b> .....	<b>32</b>
3.3.1 Pareador de Força Bruta.....	34
3.3.2 Teste de Proporção .....	36
3.3.3 Teste de Simetria.....	38
3.3.4 Matriz Fundamental .....	39
3.3.5 Teste de Geometria Epipolar.....	40
<b>4 RESULTADOS</b> .....	<b>43</b>
<b>4.1 Imagens de Entrada</b> .....	<b>43</b>
<b>4.2 Imagens do Banco de Dados</b> .....	<b>47</b>
<b>4.3 Teste de Proporção</b> .....	<b>47</b>
<b>4.4 Teste de Simetria</b> .....	<b>50</b>
<b>4.5 Teste de Geometria Epipolar</b> .....	<b>53</b>
4.5.1 Proporção 35%.....	53
4.5.2 Proporção 65% .....	54
4.5.3 Proporção 95%.....	55
4.5.4 Análise Entre as Proporções .....	56
<b>4.6 Discussão</b> .....	<b>57</b>
<b>5 CONCLUSÕES E TRABALHOS FUTUROS</b> .....	<b>59</b>
<b>REFERÊNCIAS</b> .....	<b>61</b>

## 1 INTRODUÇÃO

Com o rápido desenvolvimento da tecnologia nas últimas décadas, temos fácil acesso à dispositivos que possuem diversas funcionalidades. Uma das mais importantes dentre elas é a capacidade de capturar fotografias. Diferente de antigamente, agora esta funcionalidade não pertence apenas às câmeras analógicas, sendo possível encontrá-la em telefones celulares, tablets, computadores pessoais, relógios, videogames, e muitos outros, além é claro das câmeras digitais.

Outra grande diferença é a facilidade de se trabalhar com as fotografias capturadas. No passado era necessário utilizar filme, dessa forma não era possível visualizar a imagem capturada até que este fosse revelado. Atualmente, a imagem fica salva digitalmente em memória, sendo possível visualizá-la assim que capturada. Com isso, a disponibilidade de imagens hoje é muito grande, existindo até mesmo redes sociais voltadas para o compartilhamento delas.

Devido ao crescente número de imagens digitais ao longo dos anos, surgiu a exigência de encontrar maneiras de facilitar o acesso à elas. Graças a essa necessidade de recuperar informações pelo conteúdo, surgiu a área de pesquisa de *Content-Based Image Retrieval* (CBIR), na década de 90, sendo (KATO, 1992) o primeiro trabalho a utilizar este termo. Nesse caso a busca por uma imagem utiliza outra imagem como informação de entrada.

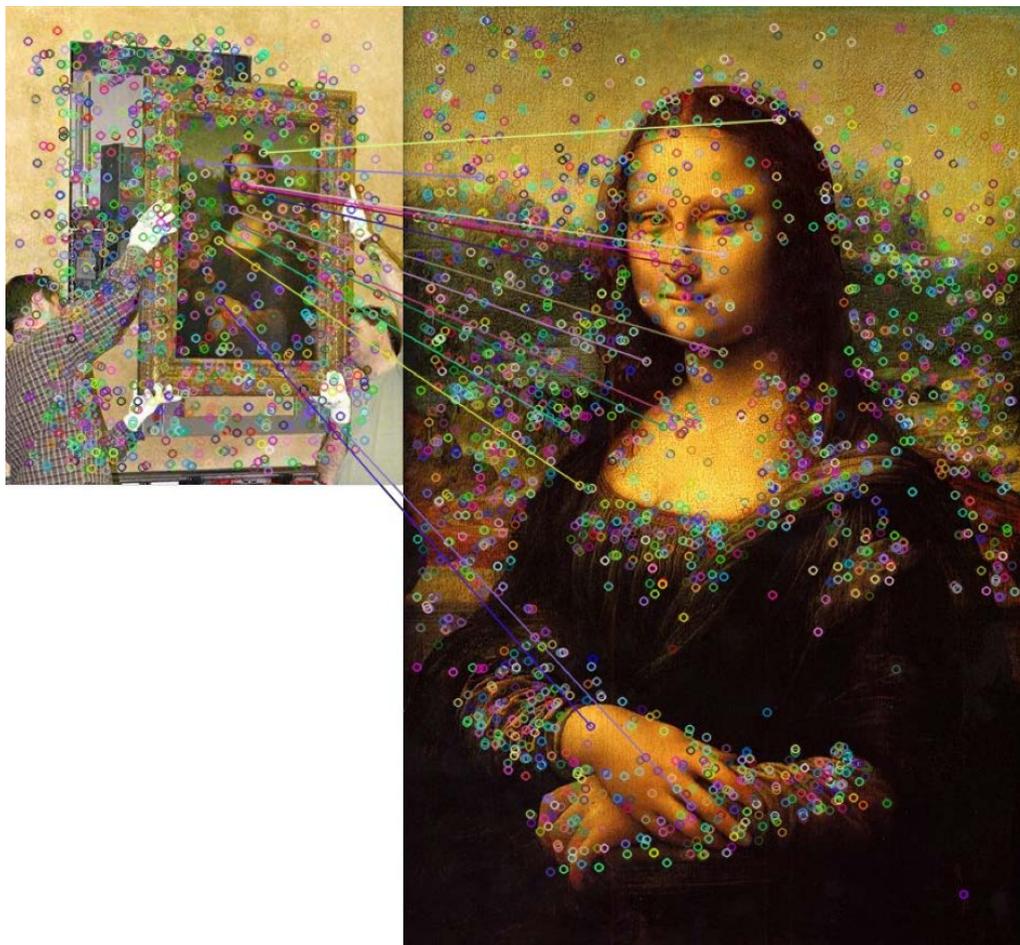
### 1.1 Objetivo

O objetivo deste projeto consiste em desenvolver uma aplicação CBIR, ou seja, recuperação de imagens por conteúdo no domínio das Artes Visuais. Este sistema deve ter a capacidade de realizar a identificação de obras de arte utilizando fotografias provenientes de uma câmera de um telefone celular ou outro dispositivo móvel, aproveitando a praticidade envolvida e a conveniência de que muitas pessoas possuem dispositivos com esta capacidade. Estas fotografias serão processadas por um algoritmo de processamento de imagens e então comparadas com um banco de dados de obras de arte para tentar identificar qual obra está presente na fotografia em questão. O desafio principal consiste nas condições não ideais de captura das imagens a serem utilizadas para busca no banco de dados, permitindo que mesmo com imagens em ângulo, rotacionadas, sem iluminação adequada, ainda assim seja possível encontrar uma correspondência.

O funcionamento do sistema consiste inicialmente na detecção de características locais, que são pontos de interesse na imagem. Após realizada a detecção dos pontos, estes são descritos, analisando a intensidade da distribuição de pixels na vizinhança de cada ponto de interesse.

Obtendo os pontos e seus descritores, somos capazes de comparar duas imagens e identificar elementos da primeira na segunda, ou vice-versa. Sendo assim, as fotografias das pinturas enviadas para o processamento deverão passar por estes dois estágios antes de serem comparadas com as imagens pré-processadas que estão no banco de dados. Através da comparação entre o descritor dos pontos da imagem enviada com os descritores das imagens do banco de dados, é possível identificar a pintura. Para reduzir a ocorrência de falsos pares de descritores, são realizados processos de refinamento nesta etapa. Um exemplo do pareamento dos pontos de interesse pode ser visto na Figura 1.1.

Figura 1.1: Demonstração do pareamento dos pontos de interesse. A imagem da esquerda representa a imagem de entrada da aplicação. A imagem da direita representa uma imagem do banco de imagens onde será realizada a busca. Os círculos representam os pontos de interesse detectados. As linhas representam o pareamento dos pontos através da análise dos seus descritores.



Fonte: Imagem resultante do sistema desenvolvido.

## 1.2 Aplicações

Um sistema de CBIR nesse domínio de pinturas pode ser empregado em diferentes aplicações e áreas, podendo ser utilizado tanto para recursos culturais, como também para fins de segurança. Um exemplo disso é a capacidade de utilizá-lo para guiar passeios em museus, ou galerias de arte, onde o visitante pode utilizar seu próprio dispositivo móvel para obter mais informações a respeito das obras em exposição. Um sistema desse tipo pode ser personalizável para cada instituição em questão, contendo no banco de dados apenas as imagens das obras que estão em exposição.

Outro exemplo, desta vez na área da segurança, é a possibilidade de ser utilizado por agentes da Polícia Federal em trabalho de campo, para verificar se pinturas analisadas por eles foram reportadas como desaparecidas, num ponto de fronteira, por exemplo. Nesse caso seria necessário um banco de dados de obras de arte roubadas ou desaparecidas, iniciativas ainda incipientes no Brasil.

## 1.3 Justificativa

O mundo da arte é repleto de diferentes pinturas, sendo praticamente impossível conhecer todas as obras existentes. Conseguir identificar pinturas desconhecidas pode ser trabalhoso até mesmo para pessoas que possuem um grande conhecimento na arte, então para um apreciador casual isso pode ser uma tarefa muito difícil.

A nossa interpretação das pinturas é realizada através de comunicação visual. Uma imagem é capaz de transmitir uma quantidade enorme de informações para uma pessoa. Conseguimos identificar facilmente qualquer elemento conhecido em uma imagem utilizando nossa memória. Entretanto, identificar elementos desconhecidos pode ser um grande problema, já que não conseguimos estabelecer nenhuma relação entre o que é visto e as informações que conseguimos lembrar.

Para conseguir identificar estes elementos desconhecidos, como uma pintura, por exemplo, é necessário analisar diversos fatores da mesma e realizar pesquisas para conseguir filtrar as opções até conseguir as informações corretas. Infelizmente, este processo pode consumir muito tempo. Estamos vivendo em um mundo onde a informação esta por todo lugar, onde tudo acontece muito rápido, e o tempo vale muito. Com isto em mente, conseguir identificar estas pinturas desconhecidas de forma eficiente possui um grande valor.

Existem outras soluções utilizadas para buscar informações de pinturas, como os QR Codes, entretanto, nem sempre eles podem ser empregados. Em aplicações como a de recuperação de obras roubadas, a única fonte de informação disponível ao agente é a própria pintura, sendo impossível utilizar QR Codes para esta aplicação.

#### **1.4 Organização do Trabalho**

No Capítulo 2, será apresentada uma revisão dos principais conceitos utilizados no trabalho, analisando trabalhos da área de identificação de características em imagens. As etapas para o desenvolvimento do sistema serão apresentadas no Capítulo 3, onde descreveremos os algoritmos e softwares utilizados durante o processo. No Capítulo 4, os resultados serão apresentados, junto com uma breve análise dos mesmos. Finalmente, no Capítulo 5, serão apresentadas as conclusões obtidas neste estudo e uma análise de possíveis trabalhos futuros.

## **2 TRABALHOS RELACIONADOS**

Neste capítulo, uma breve descrição dos conceitos utilizados neste trabalho é apresentada, assim como trabalhos relacionados na área de identificação de imagens utilizando características.

### **2.1 Histórico**

Em 1992, o termo CBIR foi introduzido em (KATO, 1992), onde foram descritos experimentos para a recuperação automática de imagens em um banco de dados através de cores e características de formatos. Desde lá, CBIR surgiu como novo campo de pesquisa e alguns trabalhos serão brevemente discutidos ao decorrer deste documento.

#### **2.1.1 Visão Geral de Trabalhos de CBIR até 2000**

Em (SMEULDERS et al., 2000) é feita uma análise de 200 referências na recuperação de imagens baseada em conteúdo. Primeiramente, são abordadas as condições de trabalho para a recuperação de imagens, como: padrões, tipos de imagens, características semânticas e sensoriais. Após isso, são discutidos os passos computacionais necessários para recuperação de imagens.

O primeiro passo a ser discutido é o processamento das imagens para a recuperação, classificadas normalmente por: cor, textura, e geometria local. Após, é discutida a utilização de características na recuperação, classificadas em características acumulativas e globais, pontos de saliência, características de objeto e formato, sinais, e combinações estruturais.

A semelhança entre imagens e objetos em imagens é analisada para cada um dos tipos de características, com os seus devidos tipos de resultados, provenientes de cada interação. Os aspectos de engenharia do sistema são brevemente discutidos, nos seguintes tópicos: banco de dados, arquitetura do sistema, e avaliação. Para concluir, é apresentada a visão dos autores à respeito da motivação da área, da herança e a influência do processamento de imagens, entre outros.

## 2.2 Tipos de Características da Imagem

Nesta seção são descritos os tipos de características de imagens presentes em outros trabalhos. Uma característica pode ser definida como uma informação relevante para resolver um problema computacional para certa aplicação. Dentre os diferentes tipos, os dois que parecem ser mais relevantes para o objetivo deste trabalho são: globais e locais.

### 2.2.1 Características Globais

As características globais (*Global Features*) são computadas na imagem inteira. As técnicas de busca de imagem que utilizam este tipo de característica dependem fortemente das cores da imagem para que a busca seja mais precisa.

Sendo assim, elas funcionam muito bem se a imagem possui cores distintas, e se o objetivo é a busca da imagem como um todo, não sendo possível distinguir objetos na imagem (TUYTELAARS; MIKOLAJCZYK, 2008).

### 2.2.2 Características Locais

Diferentemente das características globais, descritas na Seção 2.2.1, as características locais (*Local Features*) são computadas em regiões da imagem. Para realizar a separação das regiões um método de segmentação de imagem é utilizado (BANNOUR; HLAOUA; AYEB, 2009).

Estas características são representadas como padrões de imagem que possuem diferenças com a sua vizinhança próxima, que podem ser não apenas pontos na imagem, mas também bordas, cantos, ou pequenas áreas. Por fim, elas são utilizadas como centro para medidas de uma região no processo de criação dos descritores (TUYTELAARS; MIKOLAJCZYK, 2008).

## 2.3 Técnicas de Identificação de Imagens

Nesta seção estão descritas as diferentes técnicas analisadas durante a etapa de planejamento da aplicação. Elas encontram-se divididas nos diferentes tipos de características descritas na Seção 2.2. Levando em conta o objetivo do projeto, foram seleciona-

das apenas técnicas baseadas em características locais. Como um dos principais objetivos do projeto é ser capaz de identificar pinturas mesmo em imagens em condições não ideais e obtidas com diferentes câmeras, alguns pontos justificam esta escolha.

Primeiramente, é necessário considerar a diversidade de sensores encontrados nas câmeras, que podem interpretar cores de formas diferentes. Já que as características globais dependem muito das cores, isto pode reduzir muito a precisão da busca. Também existe a possibilidade de o software da câmera do aparelho utilizado realizar pós-processamento na imagem capturada. Além disso, a iluminação pode ser um importante fator para a fidelidade da captura das cores. E por último, não se pode garantir que a imagem capturada mostre apenas a pintura, sem outros elementos ao seu redor.

Para a avaliação das técnicas serão consideradas as suas capacidades de identificar imagens mesmo com diferenças de escala, rotação, posição, entre outros. Com isso, serão levantados pontos positivos e negativos para cada uma delas.

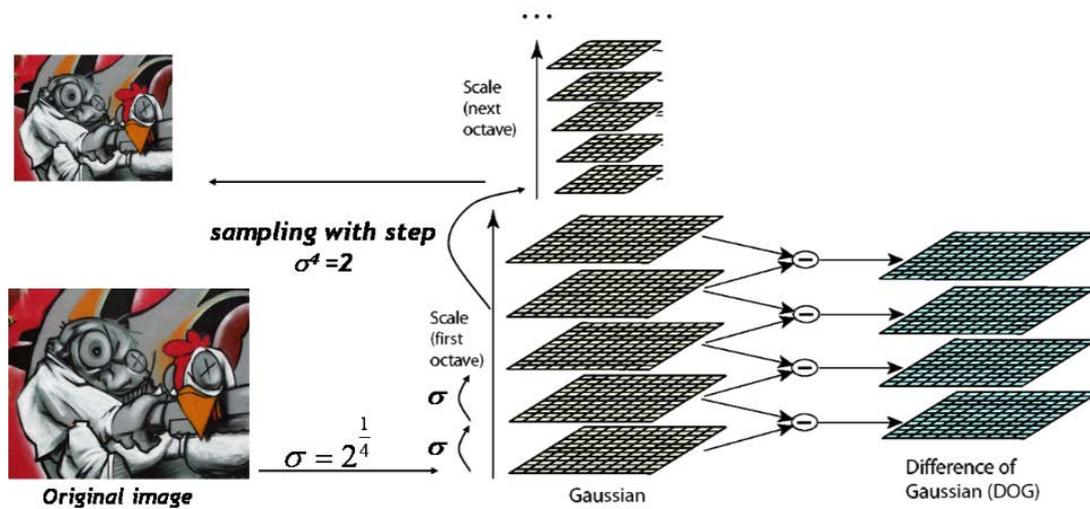
Esta análise será realizada através de informações obtidas em trabalhos de pesquisa anteriores. Grande parte destes trabalhos são comparações entre as diferentes técnicas, e até mesmo trabalhos que tratam de aplicações semelhantes à deste.

### **2.3.1 Diferença de Gaussianas**

O detector *Difference-of-Gaussians* (DoG) identifica características através da diferença de Gaussianas. Segundo (TUYTELAARS; MIKOLAJCZYK, 2008), isso é feito através da subtração de diferentes versões da imagem, que são suavizadas utilizando filtros Gaussianos. Para garantir que seja invariante à escala, esta subtração é realizada em diferentes escalas, como pode ser visto na Figura 2.1.

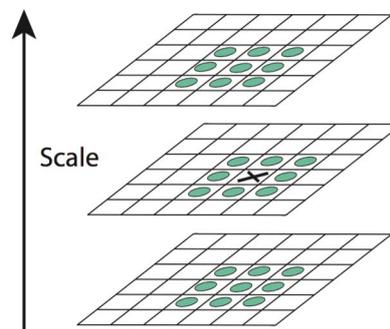
Os pontos de interesse detectados são gerados através do máximo e mínimo de cada região analisada. Esses valores são detectados fazendo a comparação de um pixel com seus vizinhos. Esta comparação inclui não apenas os 8 vizinhos de espaço, mas também os vizinhos de uma escala acima e uma abaixo. Estes passos estão representados na Figura 2.2.

Figura 2.1: Algoritmo de detecção DoG .



Fonte: (TUYTELAARS; MIKOLAJCZYK, 2008)

Figura 2.2: Vizinhos analisados na técnica DoG. Na camada central o pixel analisado está marcado com um "X", e seus 8 vizinhos de espaço são destacados por círculos na mesma camada. Nas camadas superior e inferior é possível observar os 18 vizinhos de escala, também marcados por círculos, sendo a primeira em escala maior e a última em escala menor.



Fonte: (TUYTELAARS; MIKOLAJCZYK, 2008)

### 2.3.2 Scale Invariant Feature Transform (SIFT)

Esta técnica transforma os dados da imagem em coordenadas invariantes de escala para as características locais. Um aspecto importante deste método é que ele gera um grande número de características que cobre a imagem em diferentes escalas e locais. Em (LOWE, 2004) estão descritas as principais etapas computacionais necessárias para gerar o conjunto de características em uma imagem, brevemente explicadas abaixo. Na Figura 2.3 é possível observar a capacidade deste método.

O primeiro passo desta técnica consiste em buscar pontos de interesse em todas as escalas e regiões da imagem. Para que isso seja feito, é aplicado o detector DoG, discutido

na Seção 2.3.1. Ele identifica potenciais pontos de interesse que são invariantes para a escala e orientação. Em seguida, para cada ponto considerado candidato na etapa anterior, é gerado um modelo para determinar a localização e a escala. Isso ajuda a rejeitar pontos que possuem baixo contraste ou que estão localizados nas bordas da imagem.

Figura 2.3: À esquerda estão as imagens de treinamento para os dois objetos de busca. Como pode ser visto na imagem da direita, os objetos são reconhecidos mesmo com grande parte obstruída atrás de outros itens presentes na imagem. Um paralelogramo é desenhado em volta de cada objeto reconhecido para representar as fronteiras da imagem de treinamento. Os quadrados menores representam os pontos chave utilizados para o reconhecimento.



Fonte: (LOWE, 2004)

Após ter encontrado a localização dos pontos, uma ou mais orientações são definidas para cada um deles, baseando-se na direção dos gradientes na região. Com isso, todas as futuras operações que serão executadas nas características localizadas serão invariantes para transformações de orientação, escala e localização.

Por fim, é calculado um descritor para a região em torno de cada ponto chave. Isto é feito através de amostras de intensidade na região do ponto, na escala apropriada. A razão disso é para ser capaz de eliminar as variações restantes, como as mudanças na iluminação ou ponto de vista tridimensional.

### 2.3.3 *Speeded Up Robust Features (SURF)*

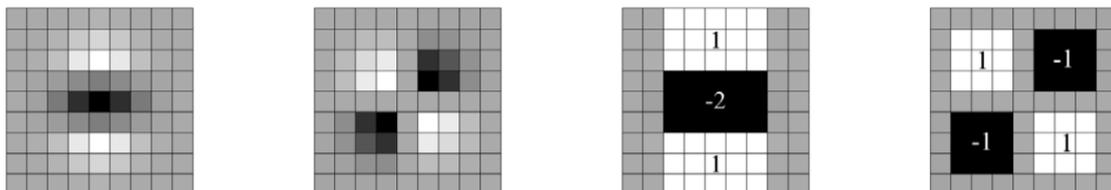
O método SURF, proposto em (BAY et al., 2008), é um detector de características invariante à escala baseado na matriz Hessiana. Ele possui um detector e um descritor das características, o primeiro detecta os pontos de interesse na imagem, o último descreve a vizinhança mais próxima destes pontos. Com estas informações é possível comparar duas imagens para tentar buscar seus pares correspondentes.

### 2.3.3.1 Detector

A propriedade mais importante de um detector de pontos de interesse é a sua repetibilidade, isto é, sua capacidade de encontrar os mesmos pontos de interesse em diferentes condições de visualização. Como dito anteriormente, o detector SURF se baseia na matriz Hessiana para encontrar os pontos de interesse. Entretanto, ao invés de utilizar formas diferentes de medir a localização e a escala, o determinante da matriz Hessiana é usado para ambos. Neste caso a matriz Hessiana é definida pelos pontos de convolução da derivada de segunda ordem do Gaussiano para um certo ponto e escala, sendo este ponto nas coordenadas  $(x, y)$ .

Utilizar filtros Gaussianos é uma solução ótima para a análise do espaço-escala. Entretanto, na prática o Gaussiano acaba causando efeito de suavização quando as imagens são sub-amostradas. Por este motivo neste detector são utilizados box-filters. Estes filtros aproximam a derivada de segunda ordem do Gaussiano, isso é feito de forma muito rápida utilizando imagens integrais, e independe do tamanho. Uma comparação entre o uso do filtro Gaussiano e os box-filters é feita na Figura 2.4, e como pode ser visto, seu resultado é semelhante.

Figura 2.4: À esquerda estão as derivadas parciais de segunda ordem da Gaussiana nas direções  $y$  e  $xy$ . À direita estão as aproximações das mesmas direções utilizando os box-filters. As áreas em cinza são iguais a zero.

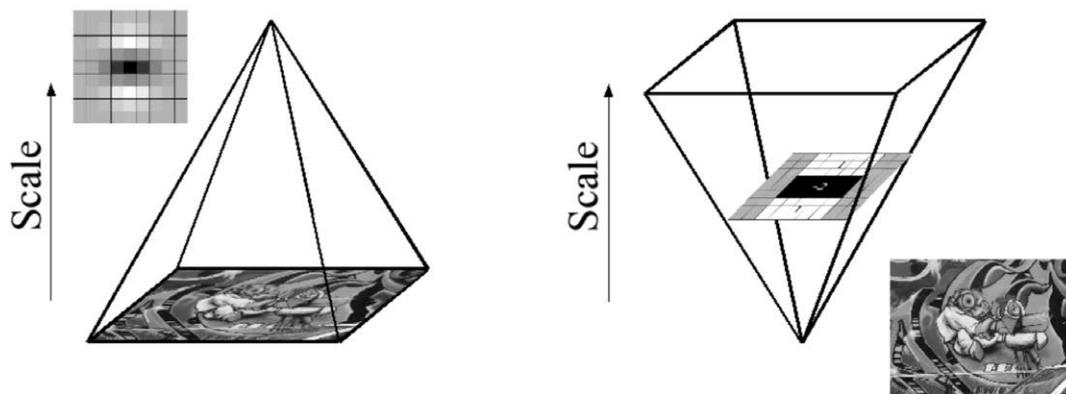


Fonte: (BAY et al., 2008)

Espaços-escala normalmente são implementados como pirâmides de imagem. Estas imagens são suavizadas repetidamente com um filtro Gaussiano e são sub-amostradas para conseguir atingir níveis mais elevados da pirâmide. Como neste detector são utilizados box-filters e imagens integrais, não é necessário aplicar o mesmo filtro em uma camada já filtrada anteriormente. Ao invés disso, é possível aplicar o filtro em qualquer tamanho com a mesma velocidade na imagem original. Sendo assim, a análise do espaço-escala é realizada com o aumento do filtro, e não com a redução do tamanho da imagem. A Figura 2.5 ilustra esta diferença.

O resultado da aplicação do box-filter de dimensão  $9 \times 9$ , como na representação vista na Figura 2.4, é considerado a escala inicial. As camadas subsequentes são obti-

Figura 2.5: Na esquerda é demonstrada a técnica que reduz iterativamente o tamanho da imagem. Na direita está representado o uso de imagens integrais para aumentar o tamanho do filtro, com um custo constante.



Fonte: (BAY et al., 2008)

das através da filtragem da imagem com máscaras gradualmente maiores. As dimensões utilizadas são de  $9 \times 9$ ,  $15 \times 15$ ,  $21 \times 21$ ,  $27 \times 27$ , e assim por diante. Em escalas maiores, o passo entre as dimensões consecutivas deve escalar da mesma forma. Desta forma, a cada oitava, o tamanho do filtro é dobrado, aumentando a diferença de 6 para 12, utilizando como exemplo as dimensões citadas acima.

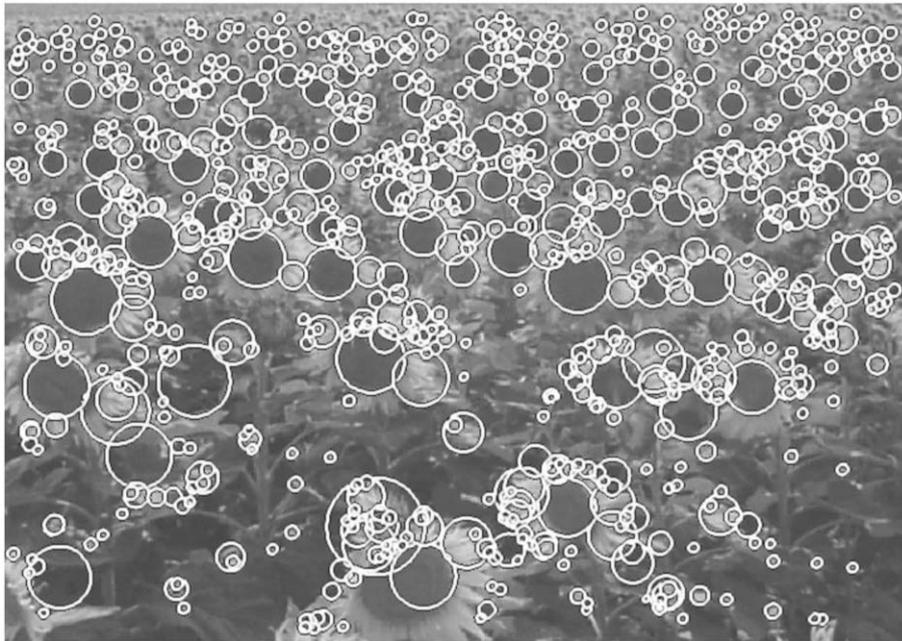
Para localizar os pontos de interesse na imagem em diferentes escalas, é realizada a supressão dos pontos que não são máximos em uma vizinhança de  $3 \times 3 \times 3$ . O máximo do determinante da matriz hessiana é interpolado na escala e no espaço da imagem. A interpolação do espaço-escala é importante neste detector, já que a diferença de escalas entre as primeiras camadas de cada oitava é grande. Na Figura 2.6 pode ser visto um exemplo dos pontos de interesse detectados utilizando este detector.

### 2.3.3.2 Descritor

O descritor SURF analisa a vizinhança de cada ponto de interesse, representando-a através de um vetor de características. Para isso, o algoritmo precisa avaliar deformações geométricas e também ser robusto o suficiente para trabalhar com o ruído na imagem.

Inicialmente, o descritor define uma orientação que possa ser reproduzida, baseando-se na informação de uma região circular que envolve o ponto de interesse, e extrai seu descritor com ela. A orientação é definida para garantir que o processo seja invariante à rotação. Com este propósito, é calculada a transformada de Haar nas direções  $x$  e  $y$  dentro de uma vizinhança circular com raio  $6s$  centrada no ponto de interesse, sendo  $s$  a escala

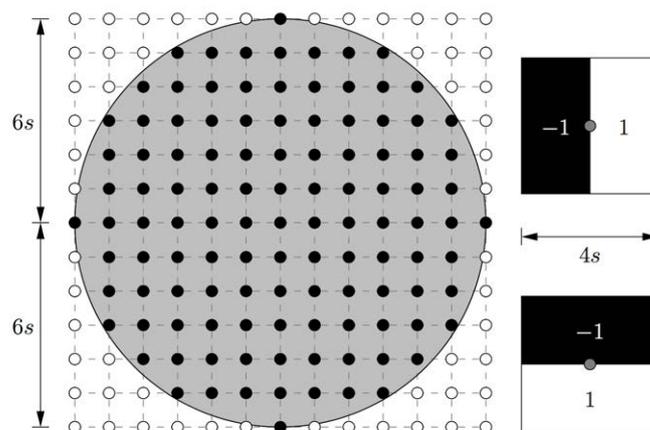
Figura 2.6: Pontos de interesse detectados em uma imagem de um campo de girassóis. Este tipo de imagem mostra a natureza das características obtidas utilizando detectores baseados na matriz Hessiana. O tamanho do círculo representa a escala do filtro utilizado.



Fonte: (BAY et al., 2008)

na qual o ponto de interesse foi detectado. O passo de amostragem depende da escala, e foi escolhido o valor de  $s$ . Por fim, o tamanho dos filtros da transformada de Haar também depende da escala, e foi definido com o valor  $4s$ . Estes estão representados na Figura 2.7. No trabalho original não é especificado como estes valores foram determinados, porém, existe a possibilidade de ter sido de forma experimental.

Figura 2.7: Definição da Orientação - À esquerda está representado a região circular de raio  $6s$ . À direita estão representados os filtros da transformada de Haar com tamanho  $4s$ .

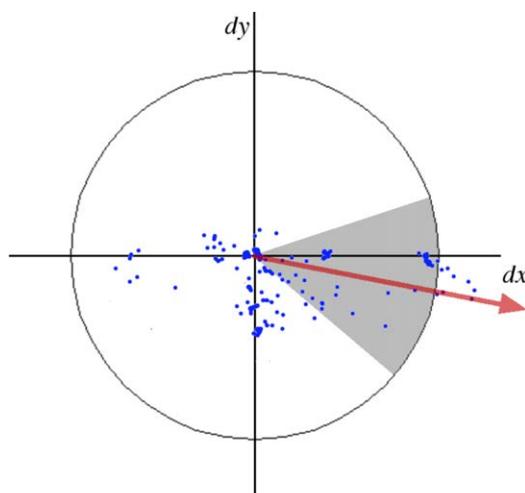


Fonte: (TERRIBERRY; FRENCH; HELMSEN, 2008)

Após obter o resultado das transformadas de Haar, é aplicado um Gaussiano centrado no ponto de interesse, os resultados são representados como vetores em um espaço

onde os resultados horizontais são distribuídos ao longo da abscissa e os verticais ao longo da ordenada. Para se determinar a orientação dominante, faz-se uma estimativa através da soma de todos os resultados dentro de uma janela deslizante de tamanho  $\pi/3$ . Os resultados horizontais e verticais são então somados dentro desta janela, isso resulta em um novo vetor. O vetor com maior comprimento é quem define a orientação do ponto de interesse. O tamanho desta janela deslizante é um parâmetro, este foi decidido experimentalmente para garantir os melhores resultados. Uma representação deste processo pode ser visto na Figura 2.8.

Figura 2.8: Definição da Orientação - Uma janela deslizante de tamanho  $\pi/3$  detecta a orientação dominante das respostas da transformada de Haar carregadas com um Gaussiano em cada ponto de amostragem dentro de uma vizinhança circular que envolve o ponto de interesse.

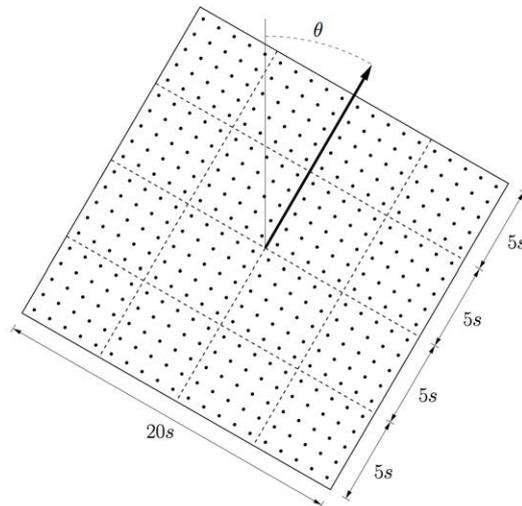


Fonte: (BAY et al., 2008)

Agora que o ponto de interesse possui uma orientação reproduzível, o próximo passo é construir uma região quadrada de tamanho  $20s$  alinhada a ela, centrada neste ponto. Na Figura 2.10 é possível ver estas regiões quadradas e sua orientação. Esta região é sub-dividida regularmente em sub-regiões quadradas menores de  $4 \times 4$ . Para cada sub-região são calculadas as transformadas de Haar em pontos de amostragem, regularmente espaçados em  $5 \times 5$ , ilustrados na Figura 2.9. Isto ocorre nas direções horizontal e vertical, em relação à orientação do ponto de interesse selecionado.

Para aumentar a robustez em relação a deformações geométricas e erros de localização, as respostas das transformadas são carregadas com um Gaussiano, centrado no ponto de interesse. A partir deste momento, as respostas das transformadas nos dois sentidos, horizontal e vertical, são somadas para cada sub-região e formam um primeiro conjunto de entradas para o vetor de características. Para obter informações à respeito da polaridade e mudanças de intensidade, também são somados os valores absolutos das

Figura 2.9: Região Quadrada - Ilustração da região quadrada de tamanho  $20s$ , com os pontos de amostragem espaçados em  $5 \times 5$  da transformada de Haar.



Fonte: (TERRIBERRY; FRENCH; HELMSEN, 2008)

respostas. Este processo resulta em um vetor de descrição para todas as sub-regiões. Este processo está representado na Figura 2.11.

Figura 2.10: Cada quadrado marcado na imagem representa uma região identificada como ponto chave à ser analisada. A dimensão de cada quadrado é resultante da escala na qual a região foi detectada.



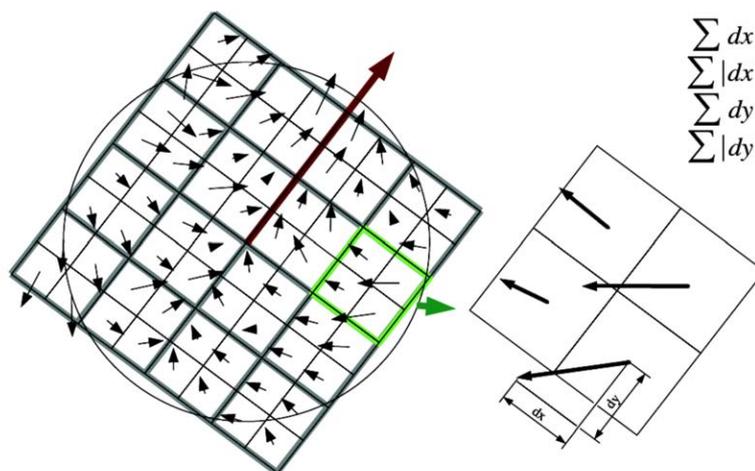
Fonte: (BAY et al., 2008)

### 2.3.4 Comparativo entre as Técnicas

Em (JUAN; GWON, 2009) é feita a comparação entre as técnicas SIFT e SURF utilizando diferentes tipos de imagens, representando natureza, cidade, objetos e arte de

rua. Para gerar os casos de teste, as imagens foram modificadas por grupos, cada grupo sofreu um diferente tipo de modificação, como transformação afim, mudança de escala, rotação, desfoque, e mudança de iluminação. Utilizando estes dados, é possível realizar uma melhor análise para decidir qual delas pode ser empregada na aplicação deste projeto.

Figura 2.11: Na esquerda, região quadrada construída na orientação pré-determinada, sub-dividida em 4x4 sub-regiões quadradas, centrada no ponto de interesse. Na direita, as sub-divisões 2x2 de cada sub-região. Cada quadrado corresponde a um campo do vetor de descrição. Essa é a soma das transformadas de Haar e seus valores absolutos, de acordo com a orientação do vetor pré-calculado.



Fonte: (BAY et al., 2008)

A técnica SIFT foi considerada lenta e não desempenha bem em casos de mudanças na iluminação, mas como já dito anteriormente, é invariante à rotação, mudanças de escala e transformações afim. Já a técnica SURF é veloz e obteve boa performance, semelhante à SIFT, mas não é estável nos casos de rotação e mudança de iluminação. O resultado completo da comparação entre as duas está na Tabela 2.1 .

Tabela 2.1: Comparação entre SIFT e SURF.

Method	Time	Scale	Rotation	Blur	Illumination	Affine
SIFT	common	best	best	best	common	good
SURF	best	good	common	good	best	good

Fonte: (JUAN; GWON, 2009)

Após ter analisado as diferentes técnicas e comparado SIFT e SURF, desconsiderado a DoG (seção 2.3.1) por ser muito simples, a escolha para a implementação deste projeto foi SURF. A escolha foi por ser mais veloz e mesmo assim obter resultados semelhantes à SIFT. Outro motivo é sua capacidade de trabalhar melhor com as questões de iluminação, o que é muito importante neste projeto, já que as condições de iluminação podem nem sempre ser ideais.

### **3 SISTEMA DESENVOLVIDO**

Neste capítulo estão detalhadas todas as etapas envolvidas no desenvolvimento do sistema. Primeiramente são descritas as ferramentas e tecnologias utilizadas nesta etapa, juntamente com a justificativa para o seu uso. Após, são descritos os passos envolvidos no processo de recuperação de imagens baseada em conteúdo do sistema gerado neste projeto. Além disso, algoritmos de refinamento dos resultados de pareamento são apresentados.

#### **3.1 Ferramentas e Tecnologias**

Nesta seção estão descritas as ferramentas e tecnologias escolhidas para o desenvolvimento do projeto, com justificativas para tais escolhas.

##### **3.1.1 Linguagem**

Dentre as diversas linguagens existentes que poderiam ser empregadas para realizar a tarefa de processamento de imagem, a escolhida foi C++. Existem vários motivos que justificam esta escolha. Primeiramente, C++ é uma linguagem simples de se aprender, sendo ela uma das primeiras linguagens abordadas em diversos cursos de graduação em computação. Outro fator importante é o grande número de bibliotecas existentes para esta linguagem. Isto é de grande ajuda quando é preciso desenvolver diversas funcionalidades em um sistema, podendo reaproveitar estas funções.

Existem outras linguagens que possuem as mesmas características citadas acima, como Java e Python. Entretanto, o fator velocidade favorece a linguagem C++. Por ser uma linguagem compilada, gerando código de baixo nível, que executa diretamente no processador da máquina, é possível se obter um ótimo desempenho. Como já vem sendo utilizada por vários anos, seu compilador já está bem otimizado. Por outro lado, tanto Java como Python são linguagens interpretadas, ou seja, seu código fonte é transformado em uma linguagem intermediária, que é interpretada pela máquina virtual da linguagem ao executar o programa, o que resulta em um desempenho mais lento.

### 3.1.2 OpenCV

Para o desenvolvimento do projeto utilizou-se a biblioteca OpenCV (ITSEEZ, 2015). Ela é uma biblioteca *open source* criada para o desenvolvimento de aplicativos na área de visão computacional. Por ser *open source*, é possível utilizar e fazer modificações no código sem nenhum custo.

Segundo (ITSEEZ, 2015) OpenCV é composta por mais de 2500 algoritmos otimizados, estando inclusos os algoritmos de detecção citados na Seção 2.3. Como possui código aberto, existe uma comunidade de mais de 47 mil colaboradores trabalhando para melhorar seus recursos e atualmente já possui 7 milhões de *downloads* (ITSEEZ, 2015). Além disso, é utilizada por diversas empresas consolidadas no mercado, como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, e Toyota. É suportada por linguagens como C++, C, Python, Java e MATLAB, sendo suportada nos sistemas operacionais Windows, Linux, Android e Mac OS.

### 3.1.3 Ambiente de Desenvolvimento

O projeto foi implementado em um computador que utiliza o sistema operacional Apple OS X El Capitan (APPLE, 2016a), versão 10.11.4. Este sistema é baseado em UNIX e é desenvolvido para funcionar em computadores Macintosh, produzidos pela empresa Apple Inc.

A IDE utilizada no projeto foi o Xcode 7.3.1 (APPLE, 2016b), também desenvolvido pela Apple Inc. Ele é um ambiente de desenvolvimento integrado que possui um conjunto de ferramentas de desenvolvimento de software desenhados para funcionar com o sistema operacional OS X.

Por fim, o compilador utilizado foi o Apple LLVM 7.1 (APPLE, 2012), padrão na instalação do Xcode. Este é baseado no projeto open source LLVM.org, e emprega uma abordagem diferenciada, construindo as tecnologias do compilador como um conjunto de bibliotecas. Múltiplos grupos de tecnologias, além da Apple Inc., são contribuidores ativos na comunidade LLVM, usando esta tecnologia para fazer as plataformas Apple ficarem mais rápidas e seguras.

### 3.2 Populando o Banco de Dados

Um importante elemento do sistema desenvolvido neste projeto é o banco de dados de pinturas digitalizadas. Através dele, imagens enviadas serão comparadas uma a uma para tentar encontrar um par com uma certa quantidade de características compatíveis. Entretanto, detectar e descrever todos os pontos de interesse de todas as imagens do banco pode consumir muito tempo, e fazer isso a cada execução do sistema é desnecessário. Como o resultado da detecção e descrição de uma imagem é sempre idêntico, guardar estas informações previamente pode salvar muito tempo.

Para isso, foi desenvolvida uma aplicação específica para criar a base de dados para o sistema principal. Ela utiliza a biblioteca OpenCV descrita na Seção 3.1.2, que possui uma implementação de ambos detector e descritor SURF, método escolhido para implementação.

Figura 3.1: Visualização dos pontos de interesse detectados em uma pintura digitalizada. Eles estão representados em cores diferentes apenas para sua distinção.



Fonte: Autor

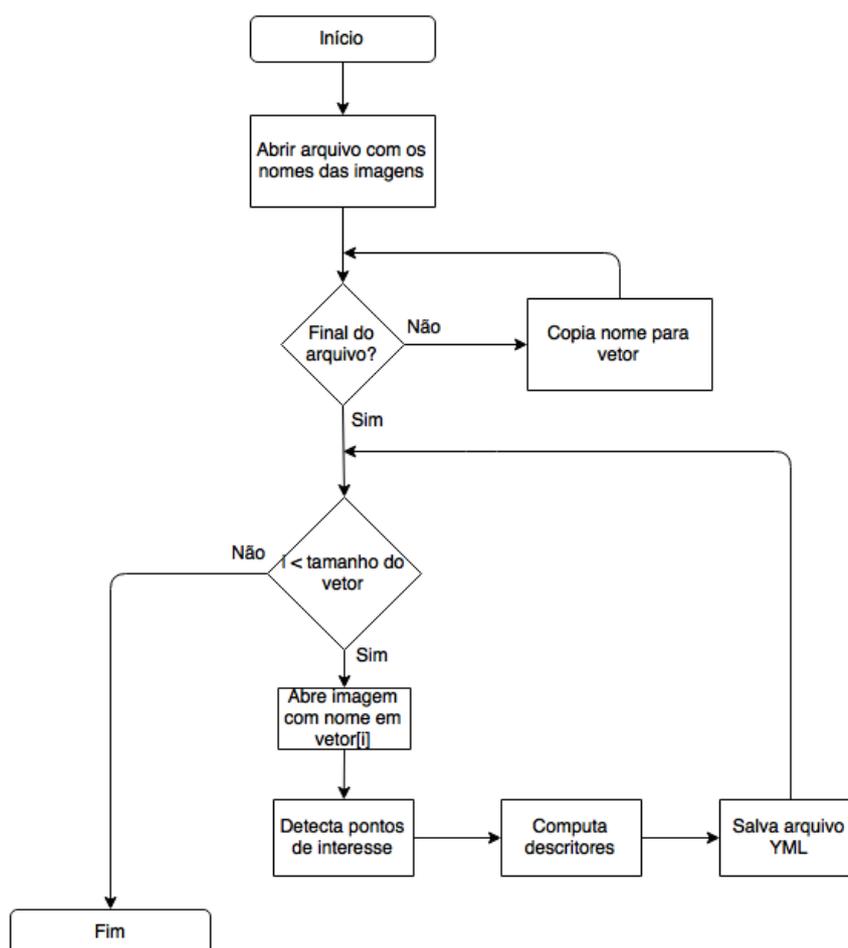
Esta aplicação usa um arquivo de texto como entrada, com os nomes dos arquivos das imagens digitalizadas para os quais se deseja gerar os dados. Durante a leitura deste arquivo, é utilizado a separação por linhas para criar um vetor dos nomes dos arquivos de imagem.

Para cada entrada deste vetor é carregada a sua respectiva imagem. Esta imagem é enviada como parâmetro para uma função que extrai as informações desejadas, retornando os pontos de interesse com seus devidos descritores. Na Figura 3.1 podem ser vistos os pontos de interesse detectados em uma pintura utilizando este processo. O próximo passo é guardar estes dados em um arquivo, contendo todos os dados retornados pela

função de extração. Ao final deste processo a base de dados com os pontos de interesse e seus descritores estará pronta para ser utilizada no sistema principal. Este processo está representado na Figura 3.2, com algumas simplificações.

Este arquivo de saída possui a extensão ".yaml". Este tipo de arquivo é codificado na linguagem YAML (BEN-KIKI; EVANS; NET, 2016), um acrônimo recursivo para *YAML Ain't Markup Language*. Esta linguagem foi desenvolvida para trabalhar com as diversas estruturas de dados utilizadas por outras linguagens. Com isso é possível criar arquivos que possuem estruturas personalizadas para a aplicação.

Figura 3.2: Fluxograma simplificado do algoritmo de população do banco de dados.



Fonte: Autor

### 3.3 Pareador de Imagens

Após a geração do banco de dados com os pontos de interesse e descritores previamente calculados, é possível realizar a busca por uma imagem de forma mais eficiente.

Assim, descrevemos nesta seção o funcionamento da principal funcionalidade deste projeto, juntamente com os algoritmos utilizados para refinar os resultados de correspondência entre os pontos de interesse.

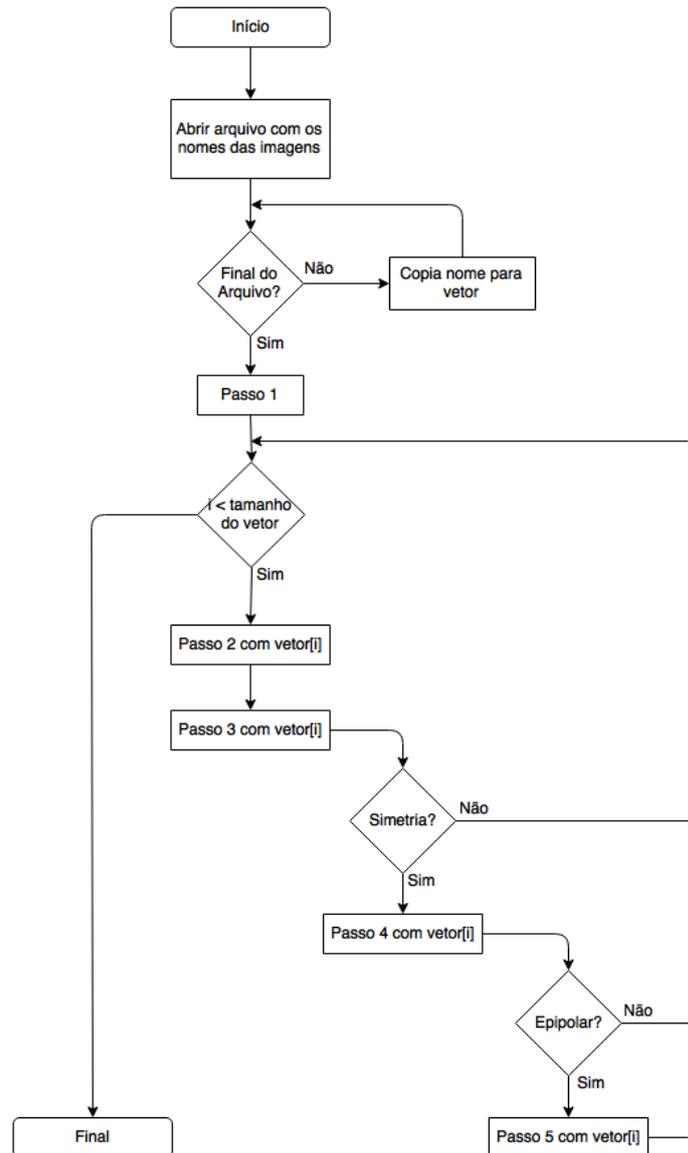
De forma semelhante ao software criado para popular o banco de dados, o Parea-dor de Imagens utiliza as mesmas ferramentas na sua implementação, além disso, também utiliza o mesmo arquivo de texto para saber quais são as pinturas previamente processadas. A partir deste arquivo, um vetor é criado com os nomes dos arquivos que serão utilizados, e com este vetor é feita a iteração que calcula a quantidade de pontos de interesse que cada imagem do banco tem em comum com a imagem de entrada.

Entretanto, antes de começar a iteração e verificar as correspondências, é necessário detectar os pontos de interesse na imagem de entrada, e gerar os descritores para cada um deles, esta etapa será chamada de Passo 1. Após obter os dados necessários, é iniciada a iteração, onde a função principal é chamada para encontrar as correspondências entre a imagem enviada e cada uma das imagens do banco de dados.

Os passos seguintes serão utilizados em cada uma das iterações para se obter o número de correspondências, e foram baseados em informações encontradas em (LAGA-NIÈRE, 2011), por conseguir bons resultados utilizando soluções simples, e ao mesmo tempo, por conter também soluções mais complexas, como visto será visto no último passo (Passo 5). Dentre eles, os dois últimos (Passos 4 e 5) são utilizados para refinar os resultados, removendo possíveis pareamentos falsos. A seguir uma breve descrição de todo o processo antes de entrar em mais detalhes.

O Passo 2 consiste no pareamento dos pontos utilizando o algoritmo de força bruta. Após, no Passo 3 é feito o teste de proporção para eliminar correspondências inválidas através da razão da distância dos dois melhores resultados para cada ponto. O Passo 4 consiste no teste de simetria das correspondências nos dois sentidos. E por fim, o Passo 5 é o teste de geometria epipolar, ou de visão estéreo, onde são removidas as correspondências que não respeitam as regras epipolares. Uma representação simplificada do algoritmo completo pode ser vista na Figura 3.3. Os Passos 4 e 5 podem ser ignorados, se necessário, como será visto no Capítulo 4. Isto é definido através de variáveis booleanas, mas mesmo assim devem ser executados na sequência correta.

Figura 3.3: Fluxograma do processo.

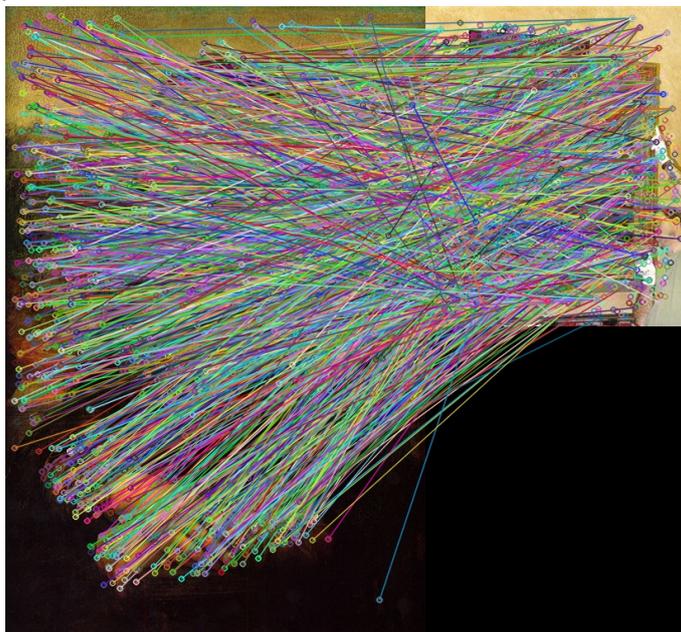


Fonte: Autor

### 3.3.1 Pareador de Força Bruta

Inicialmente, neste que será chamado de Passo 2, serão encontradas as correspondências entre os pontos de interesse utilizando um pareador de força bruta, com uma implementação que retorna as  $K$  correspondências mais próximas em ordem de aumento de distância entre os descritores. Foi utilizada uma implementação que está disponível na biblioteca OpenCV e, de acordo com (OPENCV, 2014), este pareador utiliza como padrão a norma Euclidiana para associar um número real não-negativo a cada vetor descritor. A norma Euclidiana, ou módulo, representa a raiz quadrada do produto interno do vetor por si mesmo.

Figura 3.4: Pareador de Força Bruta - Pareador utilizando apenas o Passo 2, obtendo 2281 correspondências.



Fonte: Imagem resultante do sistema desenvolvido.

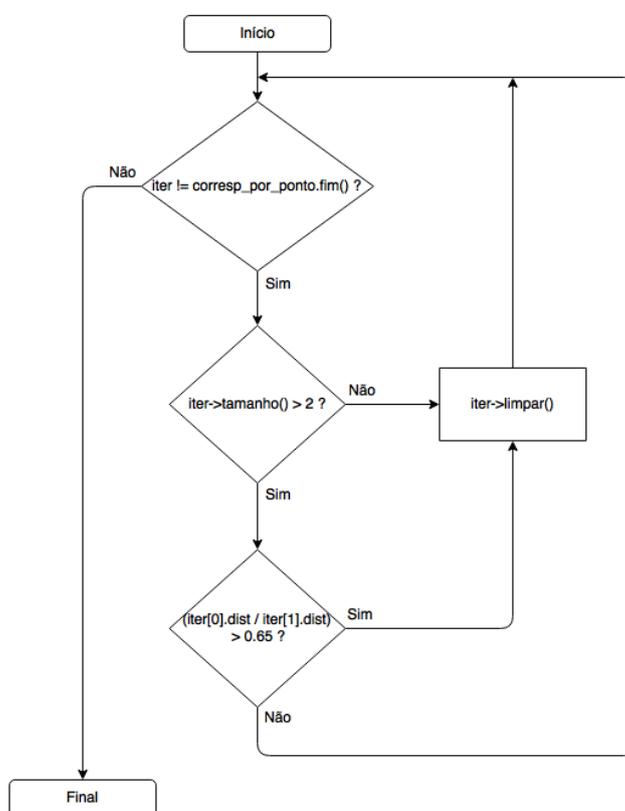
Este pareador compara todos os descritores de uma imagem com todos os da outra buscando as  $K$  melhores correspondências. Para a implementação deste sistema o valor de  $K$  foi definido como 2, pois precisamos de apenas as duas melhores correspondências para realizar o Teste de Proporção (seção 3.3.2), no Passo 3. Como o pareador utiliza números reais para quantificar os descritores, podemos assumir que as distâncias entre os descritores se mantêm as mesmas quando se inverte a ordem das imagens, então podemos dizer que o pareador é simétrico.

Além disso, este processo ocorre em duas direções, primeiro da imagem um para a imagem dois, após, da imagem dois para a um. Desta maneira, também obtemos os dois melhores candidatos na visão oposta, para cada ponto de interesse. É necessário dizer que ao utilizar apenas o pareador de força bruta, todos os pontos de interesse de uma imagem irão possuir uma correspondência na outra imagem, mesmo que estejam distantes. Então, possuindo apenas os dados de pareamento provenientes deste pareador não é possível se obter nenhum resultado que viabilize a recuperação de imagens sem que se tenha um refinamento dos mesmos. Um exemplo de imagem resultante desse processo pode ser visto na Figura 3.4.

### 3.3.2 Teste de Proporção

Utilizando as correspondências resultantes do Passo 2, será realizado um teste de proporção das distâncias entre as duas correspondências mais próximas, neste que será chamado de Passo 3. Se a distância medida é muito pequena para a melhor correspondência e muito maior para a segunda melhor, podemos aceitar a primeira como um bom resultado. No caso em que as duas são relativamente próximas, existe a possibilidade de cometer um erro ao selecionar uma à outra, rejeitando ambas neste caso, já que não se pode ter certeza se ambas estão corretas ou erradas.

Figura 3.5: Fluxograma simplificado do algoritmo do teste de proporção.

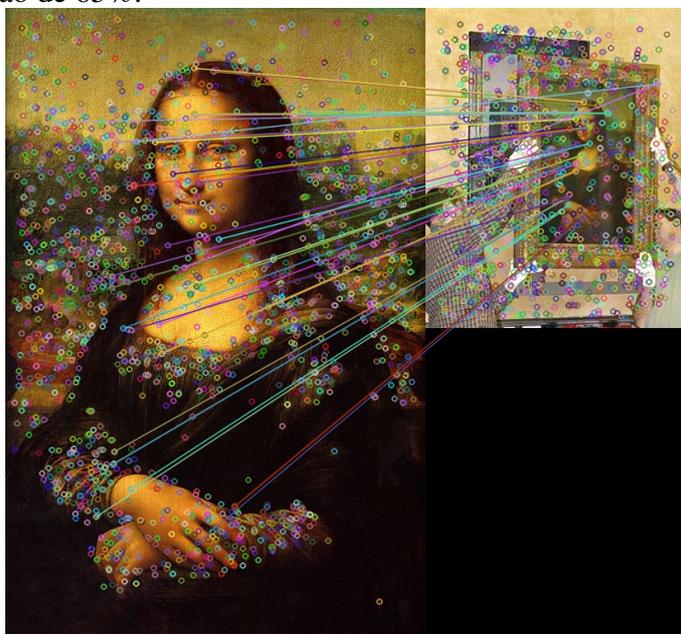


Fonte: Autor

Este processo é realizado para cada um dos resultados provenientes do Passo 2. A primeira etapa de cada iteração é conferir se foram encontradas pelo menos duas correspondências para cada ponto analisado, nos casos onde isso não ocorre, o ponto em questão será ignorado. A proporção entre distâncias é um parâmetro que pode ser alterado, se utilizarmos como exemplo o valor de 65%, a distância da melhor correspondência dividido pela distância da segunda melhor deve ser menor do que 0.65 para que seja considerada boa, isto para cada ponto de interesse. Caso seja maior, este ponto também será ignorado.

Quanto menor for o valor da proporção, mais distante estão os pontos, reduzindo ainda mais os resultados. Entretanto, dois pontos errados podem estar muito próximos, e isto pode gerar um resultado falso, então recomenda-se não diminuir muito a proporção e utilizar outros métodos de filtragem após este para se obter resultados mais precisos. Uma versão simplificada deste algoritmo está representada no fluxograma da Figura 3.5. Este processo é feito para os dois sentidos, da imagem um para dois e da imagem dois para um.

Figura 3.6: Teste de Proporção - Pareador até o Passo 3, reduzindo para 41 correspondências com proporção de 65%.



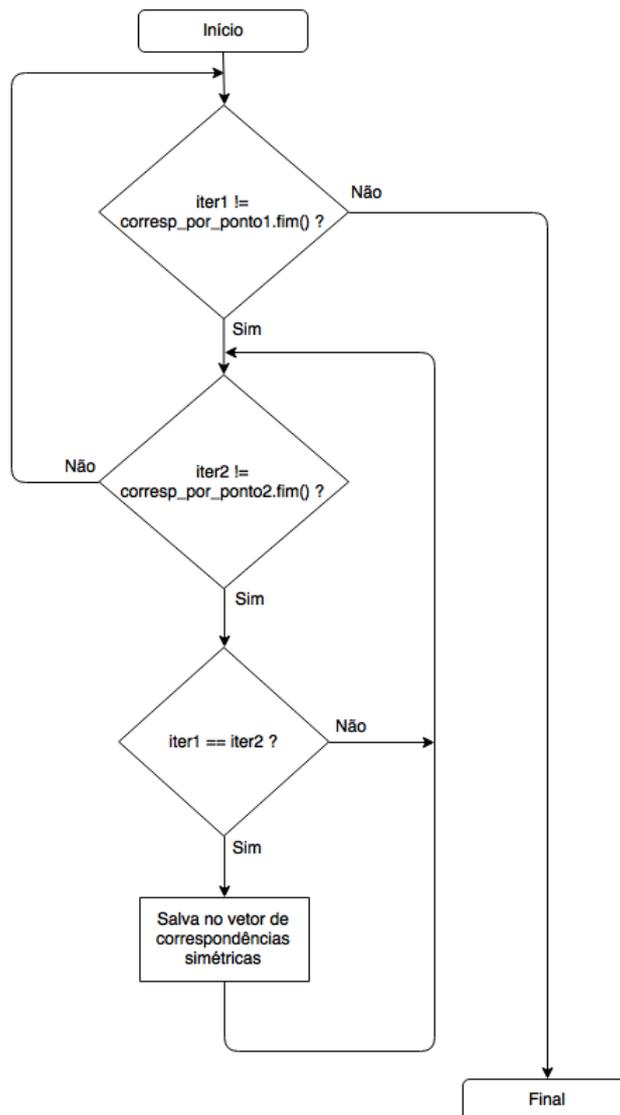
Fonte: Imagem resultante do sistema desenvolvido.

Após a execução do teste de proporção no Passo 3, será possível verificar que uma grande quantidade de correspondências ambíguas será removida, mas possivelmente ainda restarão resultados falsos. Entretanto, com a análise dos dados obtidos através deste passo, a recuperação de imagens já se torna viável, a Figura 3.6 ilustra o resultado deste passo. Para se tirar conclusões dos resultados neste ponto, deve-se levar em consideração o número de correspondências no sentido da imagem dois para a um, ou seja, do banco para a de entrada. Como a imagem do banco é de fato a obra, conteúdo que deve ser recuperado, este deve ser o sentido levado em consideração, mas o outro sentido também será utilizado no próximo passo para um refinamento ainda maior dos resultados.

### 3.3.3 Teste de Simetria

O motivo principal de realizar o pareamento em duas direções é o processamento realizado neste passo, que será chamado de Passo 4. O teste de simetria irá comparar os resultados do pareamento nos dois sentidos, com isso irá extrair apenas as correspondências que são encontradas em ambas. Este processo é feito utilizando os resultados provenientes do Passo 3.

Figura 3.7: Fluxograma simplificado do algoritmo do teste de simetria.

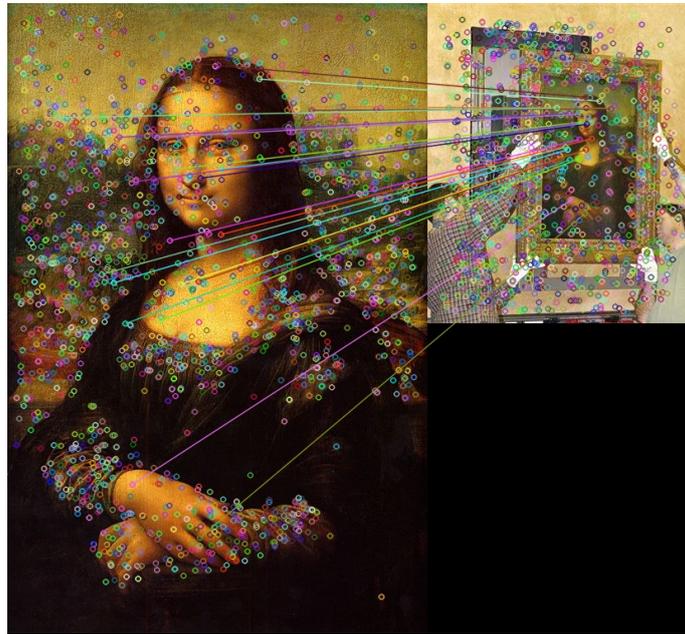


Fonte: Autor

Neste processo todos os resultados provenientes do pareamento entre a imagem enviada e a imagem do banco são comparados, um a um, com todos os resultados provenientes do pareamento no sentido contrário, da imagem do banco com a imagem enviada,

ignorando os resultados que foram removidos no Passo 3. Em cada uma destas comparações é verificado se a correspondência nos dois sentidos é igual, caso positivo, esta correspondência é salva. Uma versão simplificada deste algoritmo está representada no fluxograma da Figura 3.7. Este procedimento remove um grande número de falsos pareamentos entre as imagens, obtendo uma precisão ainda maior na recuperação da imagem correta. Utilizando novamente a imagem de exemplo, o resultado deste passo pode ser visto na Figura 3.8.

Figura 3.8: Teste de Simetria - Pareador até o Passo 4, reduzindo para 21 correspondências.



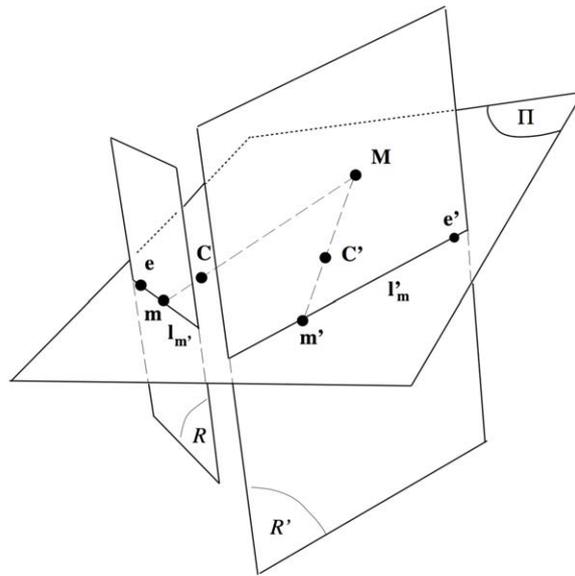
Fonte: Imagem resultante do sistema desenvolvido.

### 3.3.4 Matriz Fundamental

Antes de continuar para o próximo passo é importante revisar o conceito de Matriz Fundamental, segundo (LUONG; FAUGERAS, 1996), para melhor compreender o funcionamento do algoritmo.

Considerando o caso de duas câmeras olhando para a mesma cena. A geometria epipolar é a regra básica que resulta da existência de dois pontos de vista. Vamos considerar duas imagens em uma projeção linear em dois diferentes locais, como visto na Figura 3.9. Tomando  $C$  como o centro óptico da primeira câmera, e  $C'$  o centro óptico da segunda câmera. A linha  $(C, C')$  é projetada em um ponto  $e$  na primeira imagem  $R$ , e um ponto  $e'$  na segunda imagem  $R'$ , sendo os pontos  $e$  e  $e'$  os epipólos. As linhas que

Figura 3.9: Geometria Epipolar



Fonte: (LUONG; FAUGERAS, 1996)

cruzam  $e$  na primeira imagem e as linhas que cruzam  $e'$  na segunda imagem são as linhas epipolares. A regra epipolar é conhecida da seguinte forma na visão estéreo: para cada ponto  $m$  na primeira retina, seu ponto  $m'$  correspondente encontra-se em sua linha epipolar  $l'_m$ . De forma semelhante, para um dado ponto  $m'$  na segunda retina, seu ponto  $m$  correspondente encontra-se na sua linha epipolar  $l_m$ . As linhas  $l'_m$  e  $l_m$  são chamadas de linhas epipolares correspondentes.

O relacionamento entre as coordenadas da retina de um ponto  $m$  e sua linha epipolar  $l'_m$  correspondente é projetivo linear, por causa das relações entre  $m$  e  $(C, m)$ , e  $(C, m)$  e sua projeção  $l'_m$  são ambos projetivos lineares. A matriz  $3 \times 3$  que descreve esta correspondência é chamada de Matriz Fundamental.

### 3.3.5 Teste de Geometria Epipolar

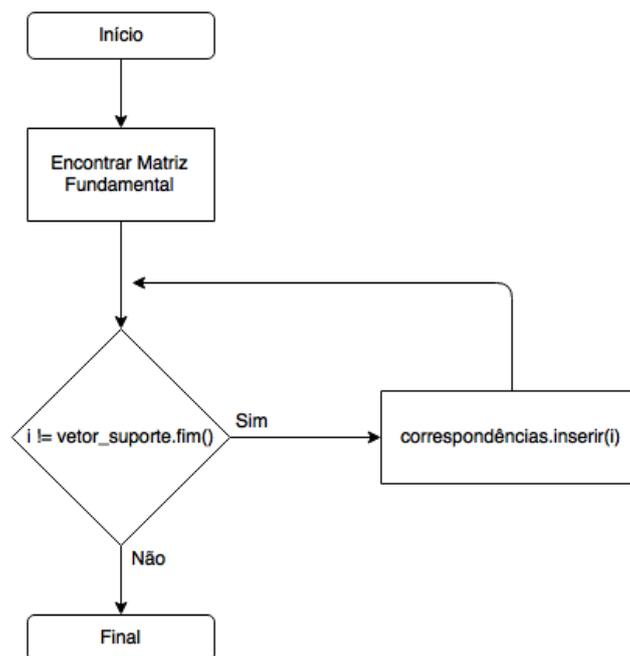
Finalmente, o último passo, que aqui será chamado de Passo 5, consiste em um teste de filtragem adicional que utiliza a matriz fundamental (seção 3.3.4) para rejeitar correspondências que não obedecem às regras da geometria epipolar, utilizando os dados provenientes do Passo 4. Com isso, tenta-se buscar pontos equivalentes nas duas imagens como se fossem a representação de uma mesma cena capturada por diferentes ângulos.

Este teste é baseado no método *Random Sampling Consensus* (RANSAC), capaz de calcular a matriz fundamental mesmo que alguns valores atípicos estejam presentes

no conjunto de correspondências. A ideia deste algoritmo é selecionar randomicamente alguns pontos do conjunto e realizar a estimativa da matriz apenas com eles. O número de pontos selecionados deve ser o número mínimo de pontos necessários para se estimar a matriz fundamental, que são 8 pares. Após ter a matriz estimada através destas 8 correspondências, todas as outras correspondências no conjunto são testadas com as regras epipolares derivadas desta matriz. Todas as correspondências que estiverem à uma distância pequena da sua linha epipolar serão identificadas. Estas correspondências formam o conjunto de suporte à matriz computada.

Quanto maior o número de correspondências presentes no conjunto de suporte, maior a probabilidade da matriz estimada ser a correta. Caso uma ou mais das correspondências selecionadas randomicamente for falsa, a matrix estimada provavelmente estará errada, então espera-se que seu conjunto de suporte seja pequeno. Este processo é repetido diversas vezes, no final do processo a matriz com o maior conjunto de suporte será mantida como a mais provável.

Figura 3.10: Fluxograma simplificado do algoritmo do teste de geometria epipolar.



Fonte: Autor

Então o objetivo é selecionar randomicamente 8 correspondências diversas vezes para que eventualmente todas sejam verdadeiras e gerar um grande conjunto de suporte. A dificuldade de se conseguir encontrar a matriz correta é proporcional ao número de correspondências falsas, por isso é importante executar outros passos de filtragem antes deste. Além disso, quanto maior o número de vezes que esta seleção é feita, maior será o

nível de confiança.

Para a implementação deste projeto foi utilizada uma função do OpenCV para buscar a matriz fundamental. Além de utilizar as correspondências das duas visões e o método RANSAC como parâmetros, esta função requer valores para distância e o nível de confiança. O valor informado para distância é o valor mínimo aceito para distância entre a correspondência e a linha epipolar. O nível de confiança é o número de iterações que serão feitas. Com isso, o maior conjunto de suporte será mantido como conjunto de correspondências aceitas. Uma versão simplificada do algoritmo pode ser vista na Figura 3.10. E a imagem mostrando as correspondências restantes após a execução deste passo está na Figura 3.11.

Figura 3.11: Teste de Geometria Epipolar - Pareador ao final do Passo 5, reduzindo para 14 correspondências.



Fonte: Imagem resultante do sistema desenvolvido.

## 4 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos através da execução do sistema para um conjunto de imagens teste de entrada. Também será discutido o conjunto de imagens que compõe o banco de dados. Por fim os resultados serão divididos em três seções, representando os passos de filtragem descritos na Seção 3.3.

### 4.1 Imagens de Entrada

Para o conjunto de imagens teste de entrada foram utilizadas fotografias de obras feitas por visitantes de galerias ou museus, disponíveis publicamente na internet. O objetivo aqui foi criar um conjunto de imagens que simulassem a experiência de um usuário do sistema, onde se apresentam várias situações não ideais para testar a capacidade do sistema. As seguintes imagens fazem parte do conjunto de imagens de entrada, com suas justificativas nas descrições.

Figura 4.1: Imagem 1 - Mona Lisa - Fotografia com pequeno ângulo, reflexo, informações desnecessárias e com resolução 427x640.



Fonte: (KAMPHUIS, 2008)

Figura 4.2: Imagem 2 - Mona Lisa - Fotografia com ângulo, afastada, muita informação desnecessária e resolução 1600x900.



Fonte: (MCKELVEY, 2013)

Figura 4.3: Imagem 3 - Mona Lisa - Fotografia direta, relativamente próxima, com obstruções, informações desnecessárias e resolução 2048x1024.



Fonte: (WELLINGTONANDSQUEAK, 2015)

Figura 4.4: Imagem 4 - Quarto em Arles - Fotografia direta, com resolução 1024x874.



Fonte: (NIELS, 2009)

Figura 4.5: Imagem 5 - Vaso com quinze girassóis - Fotografia direta, relativamente próxima, com obstruções, informações desnecessárias e resolução 810x1024.



Fonte: (BRUIJN, 2013)

Figura 4.6: Imagem 6 - Vaso com quinze girassóis - Fotografia direta, próxima, com obstruções, informações desnecessárias e resolução 1024x768.



Fonte: (WAGENAAR, 2009)

Figura 4.7: Imagem 7 - O Nascimento de Vênus - Fotografia com ângulo, distância média, informações desnecessárias e resolução 1024x765.



Fonte: (JOHNSTONE, 2014)

Figura 4.8: Imagem 8 - O Nascimento de Vênus - Fotografia com ângulo, próxima, contém reflexos e resolução 1024x768.



Fonte: (MAGGIE, 2010)

Figura 4.9: Imagem 9 - O Nascimento de Vênus - Fotografia direta, próxima, informações desnecessárias e resolução 1024x683.



Fonte: (CARDOSO, 2015)

## 4.2 Imagens do Banco de Dados

Para popular o banco de dados foram selecionadas 85 pinturas. Dentre estas, algumas foram inseridas apenas com a intenção de aumentar o tamanho do banco. Outras foram inseridas para que existissem correspondências para as imagens de entrada. Todas as imagens do banco estão em condições ideais, para que nenhuma informação desnecessária possa ser identificada erroneamente.

Primeiramente foram escolhidas algumas pinturas do conjunto de imagens utilizado em (BANSAL; DANILIDIS, 2013) e (HAUAGGE; SNAVELY, 2012), por conter imagens com diferenças de iluminação, idade e diferentes técnicas. Em seguida, foram selecionadas obras de artistas famosos, como Leonardo da Vinci, Van Gogh, Pablo Picasso e outros. As imagens destas obras foram obtidas através de (HARDEN, 2016).

## 4.3 Teste de Proporção

Nesta seção estão descritos os resultados obtidos utilizando apenas o algoritmo de filtragem do Passo 3, descrito na Seção 3.3.2. Foram escolhidos três valores para o parâmetro da proporção: 0,35; 0,65; 0,95, sendo que o valor de 65% é sugerido em (LAGANIÈRE, 2011), os outros dois valores correspondem à soma ou subtração de 0, 30 do valor sugerido. Os resultados obtidos estão na Tabela 4.1, juntamente com o tempo de execução a partir do Passo 2 até este passo.

Tabela 4.1: Teste de Proporção - Resultados

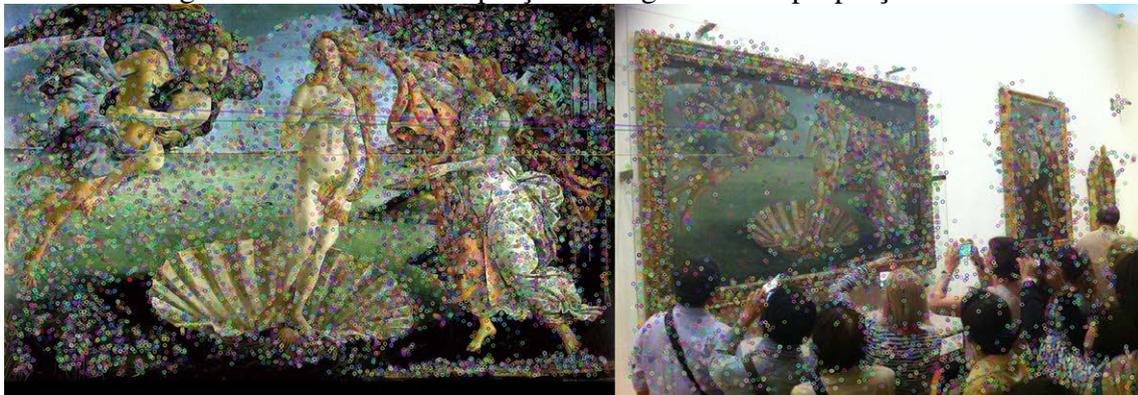
	Proporção 35%			Proporção 65%			Proporção 95%		
	Correspondências	Acerto	Tempo	Correspondências	Acerto	Tempo	Correspondências	Acerto	Tempo
Imagem 1	4	Sim	116,5s	65	Não	117,5s	5691	Não	117,8s
Imagem 2	2	Não	330,0s	25	Não	330,8s	5675	Não	331,2s
Imagem 3	0	Não	351,1s	26	Sim	352,3s	5408	Não	352,6s
Imagem 4	56	Sim	387,5s	502	Sim	388,1s	5501	Não	389,1s
Imagem 5	24	Sim	146,1s	158	Sim	146,8s	5723	Não	146,7s
Imagem 6	82	Sim	195,1s	510	Sim	195,8s	5607	Não	195,1s
Imagem 7	3	Sim	216,8s	139	Sim	217,7s	6022	Sim	218,8s
Imagem 8	1	Sim	235,8s	140	Sim	237,5s	6020	Sim	239,5s
Imagem 9	67	Sim	196,8s	656	Sim	197,8s	6362	Sim	197,2s

Fonte: Autor

Ao analisarmos os resultados obtidos utilizando a proporção 35%, podemos ver que os resultados foram muito bons para as imagens sem ângulos e com poucas informações desnecessárias. Para as imagens capturadas em situações menos ideais, o número de correspondências foi muito próximo de 0, podendo acontecer como no caso da Imagem 3, que acabou não encontrando nenhuma correspondência. No caso da Imagem 2, além das

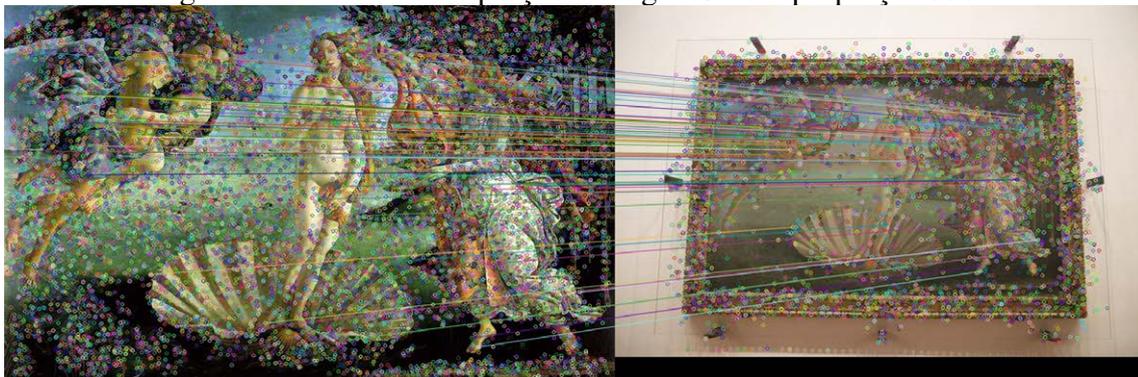
justificativas citadas acima, certamente é a imagem capturada com as piores condições, e não encontrou correspondências. Nas Figuras 4.10 e Figura 4.11 é possível ver resultados para esta proporção.

Figura 4.10: Teste de Proporção - Imagem 7 com proporção 35%.



Fonte: Imagem resultante do sistema desenvolvido.

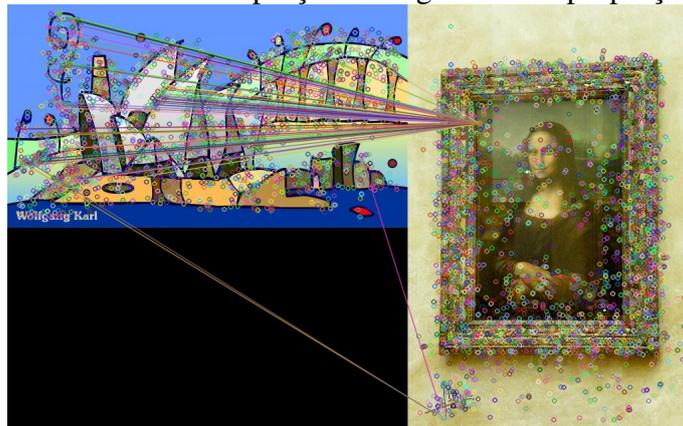
Figura 4.11: Teste de Proporção - Imagem 9 com proporção 35%.



Fonte: Imagem resultante do sistema desenvolvido.

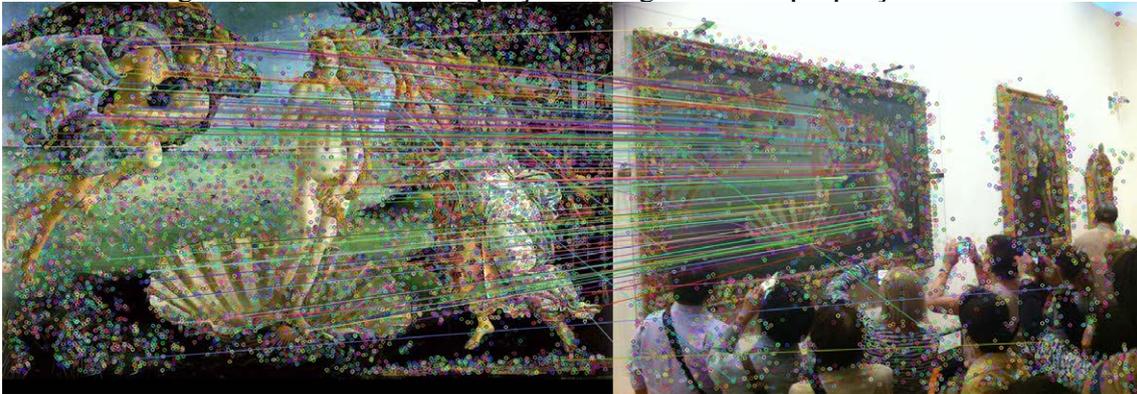
A proporção 65% obteve bons resultados em ambos os tipos de imagem, exceto para as Imagens 1 e 2. No caso da Imagem 1, é possível visualizar na Figura 4.12 um dos problemas de se utilizar apenas o teste de proporção, pois baseando-se apenas na proporção não eliminamos casos como este em que diversos pontos de uma imagem podem ter correspondência com apenas um ponto da outra. Para a Imagem 2, se aplicam as mesmas justificativas usadas na proporção 35%. Para as demais, imagens em condições mais próximas de ideais retornaram um grande número de correspondências e as capturadas em situações menos ideais obtiveram agora um número considerável de correspondências, encontrando também uma correspondência para a Imagem 3. Nas Figuras 4.13 e 4.14 é possível ver resultados para esta proporção.

Figura 4.12: Teste de Proporção - Imagem 1 com proporção 65%.



Fonte: Imagem resultante do sistema desenvolvido.

Figura 4.13: Teste de Proporção - Imagem 7 com proporção 65%.



Fonte: Imagem resultante do sistema desenvolvido.

Figura 4.14: Teste de Proporção - Imagem 9 com proporção 65%.



Fonte: Imagem resultante do sistema desenvolvido.

Utilizando a proporção 95% não é possível se obter resultados. A proporção é muito grande e acaba aceitando muitas correspondências falsas, isso explica o número enorme de correspondências encontradas. Então este valor de proporção não é confiável para ser utilizado. Na Figura 4.15 é possível ver resultados para esta proporção.

Figura 4.15: Teste de Proporção - Imagem 4 com proporção 95%.



Fonte: Imagem resultante do sistema desenvolvido.

Por fim, dentre os três casos analisados nesta seção para o teste de proporção, as duas melhores opções para se utilizar seriam 35% e 65%. Entretanto, por retornar uma maior quantidade de correspondências, a proporção 65% é mais aconselhada. E ainda assim é possível verificar que mesmo obtendo acertos, ainda restam correspondências falsas. Quanto às diferenças nos tempos de execução, só é possível notar diferenças consideráveis no tempo entre as imagens. Isso se dá devido às diferentes resoluções utilizadas nas imagens de entrada. A diferença de tempo entre as proporções para uma mesma imagem é desprezível.

#### 4.4 Teste de Simetria

Nesta seção estão descritos os resultados obtidos utilizando o algoritmos de filtragem até o Passo 4, descrito na Seção 3.3.3. Como o Passo 3 é executado antes deste, foram escolhidos os mesmos três valores para o parâmetro da proporção: 0,35; 0,65; 0,95. Com isso é possível comparar os resultados deste teste com os do Teste de Proporção da seção anterior. Os resultados obtidos estão na Tabela 4.2, juntamente com o tempo de execução a partir do Passo 2 até este passo.

Tabela 4.2: Teste de Simetria - Resultados

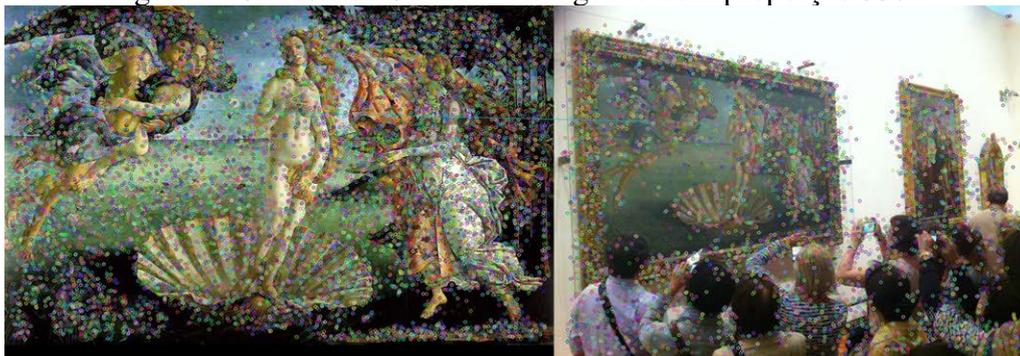
	Proporção 35%			Proporção 65%			Proporção 95%		
	Correspondências	Acerto	Tempo	Correspondências	Acerto	Tempo	Correspondências	Acerto	Tempo
Imagem 1	0	Não	116,8s	17	Sim	117,1s	451	Não	122,5s
Imagem 2	0	Não	330,8s	3	Não	329,5s	848	Não	344,8s
Imagem 3	0	Não	351,2s	15	Sim	351,4s	858	Não	367,6s
Imagem 4	38	Sim	387,6s	322	Sim	388,9s	1079	Sim	407,2s
Imagem 5	3	Sim	145,9s	68	Sim	147,4s	416	Não	153,8s
Imagem 6	33	Sim	195,0s	245	Sim	194,9s	824	Sim	204,2s
Imagem 7	1	Sim	216,3s	52	Sim	217,4s	849	Sim	228,9s
Imagem 8	1	Sim	237,4s	28	Sim	237,6s	987	Sim	249,0s
Imagem 9	18	Sim	196,5s	231	Sim	197,3s	1183	Sim	206,3s

Fonte: Autor

Os resultados obtidos com a proporção 35% foram razoavelmente bons, porém,

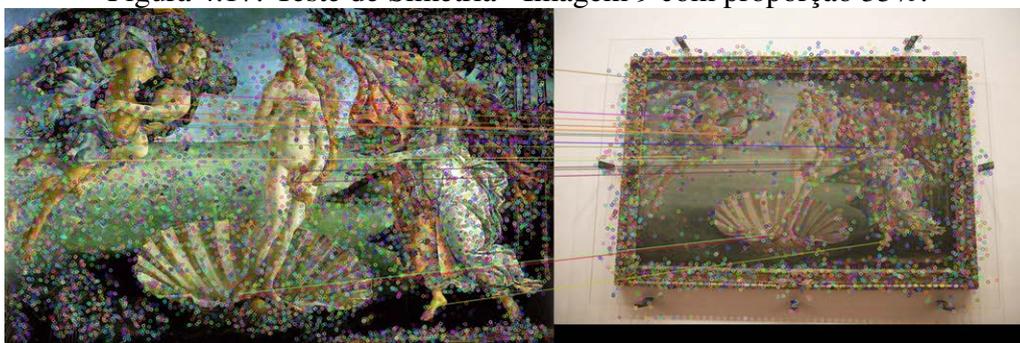
como vimos na seção 4.3, os casos em que as pinturas não foram capturadas de forma ideal resultam em poucas correspondências. Como o teste de simetria tende a reduzir ainda mais este valor, isso torna esta proporção menos apropriada para ser utilizada em conjunto com este teste. A redução no número de correspondências entre o teste de simetria e o teste de proporção para este caso foi em média 69%. Nas Figuras 4.16 e Figura 4.17 é possível ver resultados para esta proporção.

Figura 4.16: Teste de Simetria - Imagem 7 com proporção 35%.



Fonte: Imagem resultante do sistema desenvolvido.

Figura 4.17: Teste de Simetria - Imagem 9 com proporção 35%.

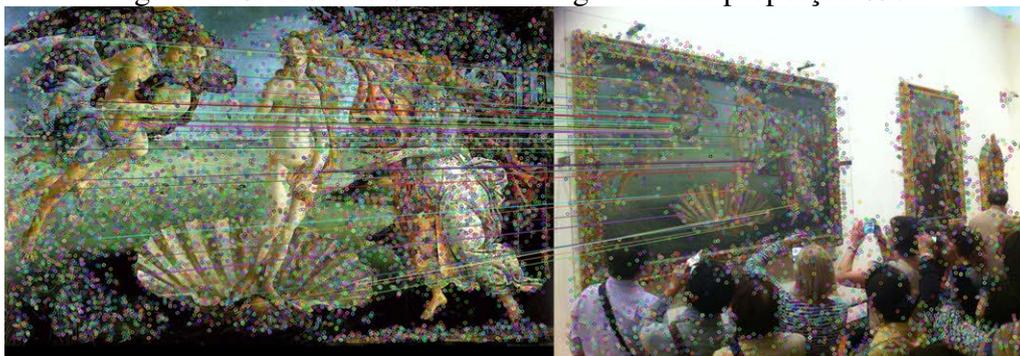


Fonte: Imagem resultante do sistema desenvolvido.

A proporção 65% obteve bons resultados em ambos tipos de imagem, exceto para a Imagem 2. No caso da Imagem 2, se aplicam as mesmas justificativas usadas na seção 4.3. Para as demais, imagens em condições mais próximas de ideais retornaram um bom número de correspondências e as capturadas em situações menos ideais obtiveram agora um número considerável de correspondências. Podemos notar que a Imagem 1 agora obteve acerto, diferente do resultado obtido utilizando apenas o teste de proporção. A redução no número de correspondências entre o teste de proporção e o teste de simetria para este caso foi em média 62%. Nas Figuras 4.18 e Figura 4.17 é possível ver resultados para esta proporção.

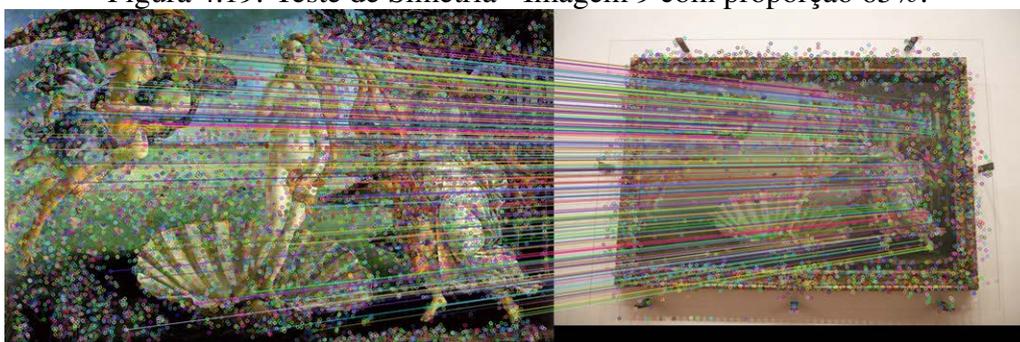
Para a proporção 95% houve uma pequena melhora nos resultados, mas ainda assim continua sendo pior que os demais. Agora foi possível identificar as Imagens 4 e 6,

Figura 4.18: Teste de Simetria - Imagem 7 com proporção 65%.



Fonte: Imagem resultante do sistema desenvolvido.

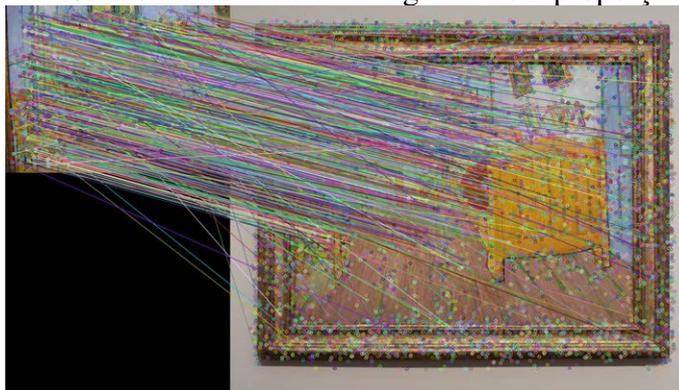
Figura 4.19: Teste de Simetria - Imagem 9 com proporção 65%.



Fonte: Imagem resultante do sistema desenvolvido.

diferente do que aconteceu no teste de proporção. Outra grande diferença é a redução no número de correspondências, que em média foi de 86% para esta proporção. Na Figura 4.20 é possível ver resultados para esta proporção.

Figura 4.20: Teste de Simetria - Imagem 4 com proporção 95%.



Fonte: Imagem resultante do sistema desenvolvido.

Por fim, dentre os três casos analisados nesta seção para o teste de simetria, as duas melhores opções para se utilizar continuam sendo 35% e 65%. Entretanto, desta vez a proporção 65% obteve um desempenho ainda superior, sendo assim o mais recomendável. E assim como no teste de proporção é possível verificar que mesmo obtendo acertos, ainda restam correspondências falsas. A diferença no tempo de execução entre as técnicas pode

ser desconsiderada, exceto para o caso em que a proporção é de 95% que obteve um pequeno aumento.

#### **4.5 Teste de Geometria Epipolar**

Nesta seção estão descritos os resultados obtidos utilizando todos os algoritmos de filtragem, incluindo agora o Passo 5, descrito na Seção 3.3.5. Como nos passos anteriores à este, foram escolhidos os mesmos valores para os parâmetros de proporção para fins de comparação: 0,35; 0,65; 0,95. Além disso, aqui também serão utilizados outros dois parâmetros. Primeiro o valor de distância da linha epipolar, variando entre os valores: 0,5; 1,0; 2,0, sendo que o valor sugerido em (LAGANIÈRE, 2011) para este parâmetro é de 1,0. O outro parâmetro a ser definido é o nível de confiança da aproximação da Matriz Fundamental, variando entre os valores: 0,39; 0,69; 0,99. Com o valor de 0,99 sugerido em (LAGANIÈRE, 2011). Desta vez os resultados serão divididos em seções de acordo com o parâmetro de proporção.

##### **4.5.1 Proporção 35%**

Aqui serão analisados os resultados que utilizam o valor 0,35 como parâmetro de proporção. Na Tabela 4.3 estão listados os resultados, juntamente com o tempo de execução a partir do Passo 2 até este passo.

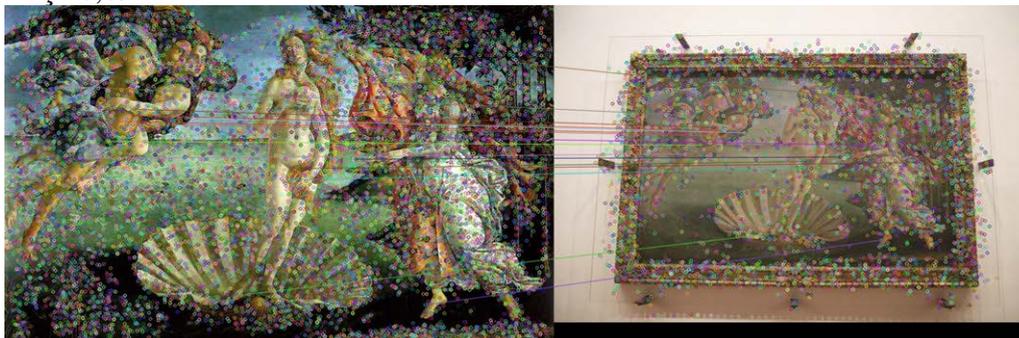
Ao analisar a tabela com os resultados do teste realizado com esta porcentagem, é fácil visualizar que apenas as imagens em condições mais próximas de ideais obtiveram algum resultado. Isso é consequência da redução das correspondências que acontece neste passo. Como no passo anterior os valores de correspondência já estavam baixos para imagens em condições não ideais, agora acabaram desaparecendo. Se observamos os valores das imagens que obtiveram acerto, é possível analisar que o aumento feito na distância obteve uma influência maior no número de correspondências do que a confiança. Então o teste de geometria epipolar não é aconselhável ao se utilizar esta proporção. Na Figura 4.21 é possível ver resultados para esta proporção.

Tabela 4.3: Teste de Geometria Epipolar - Resultados com proporção 35%

Distância	Confiança		Imagem 1	Imagem 2	Imagem 3	Imagem 4	Imagem 5	Imagem 6	Imagem 7	Imagem 8	Imagem 9
0.5	0.39	Correspondências	0	0	0	31	0	21	0	0	13
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	116.9	328.6	351.9	387.5	145.3	194.8	216.1	236.8	196.9
	0.69	Correspondências	0	0	0	31	0	21	0	0	13
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	117	328.5	351.6	387.2	145.7	194.6	216.7	237	196.8
0.99	Correspondências	0	0	0	31	0	21	0	0	13	
	Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim	
	Tempo(s)	117.9	328.8	350.5	386.9	146	194.7	217.6	237.5	196.6	
1	0.39	Correspondências	0	0	0	32	0	27	0	0	16
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	116.5	328.8	352.3	389.3	145.8	194.8	216.1	237.1	196.5
	0.69	Correspondências	0	0	0	34	0	27	0	0	16
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	116.8	328.6	351.2	388	146.1	194.4	214.9	237	196.5
0.99	Correspondências	0	0	0	36	0	27	0	0	16	
	Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim	
	Tempo(s)	116.9	329.1	351.7	387.6	145.8	194.9	217.6	237.3	196.5	
2	0.39	Correspondências	0	0	0	37	0	27	0	0	18
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	117	328.6	351.2	387.3	145.8	194.4	216.6	237.3	197
	0.69	Correspondências	0	0	0	37	0	27	0	0	18
		Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim
		Tempo(s)	116.4	328.5	351.6	387.4	145.8	195	215.8	237.5	196.7
0.99	Correspondências	0	0	0	37	0	31	0	0	18	
	Acerto	Não	Não	Não	Sim	Não	Sim	Não	Não	Sim	
	Tempo(s)	116.6	328.7	351.5	388.6	145.7	195.1	216	237.1	196.1	

Fonte: Autor

Figura 4.21: Teste de Geometria Epipolar - Imagem 9 com proporção 35%, distância 1, e confiança 0,69.



Fonte: Imagem resultante do sistema desenvolvido.

#### 4.5.2 Proporção 65%

Aqui serão analisados os resultados que utilizam o valor 0,65 como parâmetro de proporção. Na Tabela 4.4 estão listados os resultados, juntamente com o tempo de execução a partir do Passo 2 até este passo.

Com os resultados obtidos utilizando esta porcentagem, podemos verificar que o resultado de acerto da recuperação da imagem foi o mesmo para todas as variações de distância e confiança. Existe um aumento no número de correspondências conforme o valor destes parâmetros cresce, mas o resultado final se mantém. Esta estabilidade nos resultados é proveniente da qualidade da obtenção de dados já realizada na etapa anterior, apenas reduzindo ainda mais a quantidade de correspondências, mas ainda assim

Tabela 4.4: Teste de Geometria Epipolar - Resultados com proporção 65%.

Distância	Confiança		Imagem 1	Imagem 2	Imagem 3	Imagem 4	Imagem 5	Imagem 6	Imagem 7	Imagem 8	Imagem 9
0.5	0.39	Correspondências	12	0	9	197	28	107	24	17	96
		Acerto	Sim	Não	Sim						
		Tempo(s)	116.8	328.7	352.3	388.8	146.5	194.9	216.8	236.9	196.4
	0.69	Correspondências	12	0	10	197	32	107	24	19	96
		Acerto	Sim	Não	Sim						
		Tempo(s)	116.6	328.3	352.4	389.3	145.9	195.8	216.6	236.9	196.6
0.99	Correspondências	12	0	11	197	32	107	24	19	102	
	Acerto	Sim	Não	Sim							
	Tempo(s)	117.1	329.7	352	388.2	146.2	195.1	216.5	237.3	196.9	
1	0.39	Correspondências	13	0	11	251	39	144	31	19	138
		Acerto	Sim	Não	Sim						
		Tempo(s)	117.1	328.8	351.8	386.6	146	194.6	216.3	237.5	196.6
	0.69	Correspondências	13	0	12	251	40	147	31	23	140
		Acerto	Sim	Não	Sim						
		Tempo(s)	117	328.9	351.6	387.3	147.4	194.8	216.2	236.9	196.7
0.99	Correspondências	13	0	12	273	42	161	31	23	150	
	Acerto	Sim	Não	Sim							
	Tempo(s)	117.1	327.6	352.1	387.4	146.4	194.8	216.6	237.7	196.3	
2	0.39	Correspondências	14	0	12	301	48	187	39	22	187
		Acerto	Sim	Não	Sim						
		Tempo(s)	116.8	329	351.9	386.5	146.1	195	218.3	237.2	197.2
	0.69	Correspondências	17	0	12	301	53	209	39	23	192
		Acerto	Sim	Não	Sim						
		Tempo(s)	116.5	328	351.4	387.1	146.6	195	216.4	237	195.7
0.99	Correspondências	17	0	14	301	62	209	43	25	202	
	Acerto	Sim	Não	Sim							
	Tempo(s)	116.8	329.2	351.5	387.4	146	194.8	215.9	237.5	196.5	

Fonte: Autor

mantendo uma boa quantidade. Vale notar que a Imagem 2 não obteve acertos. Nas Figuras 4.22 e Figura 4.23 é possível ver resultados para esta proporção.

Figura 4.22: Teste de Geometria Epipolar - Imagem 7 com proporção 65%, distância 1, e confiança 0,69.



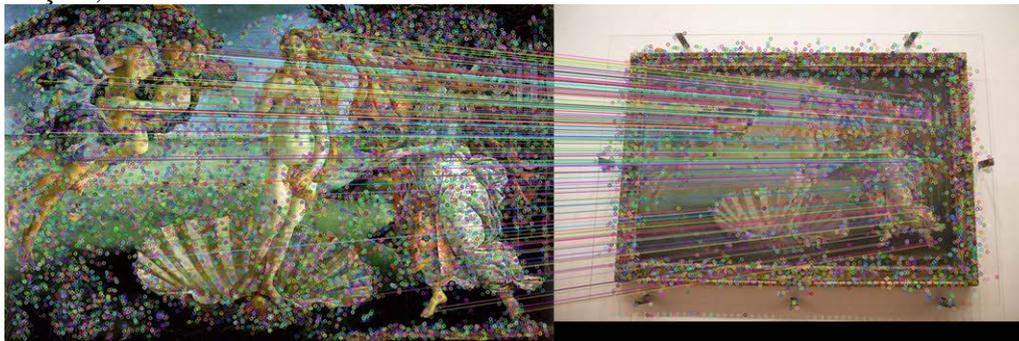
Fonte: Imagem resultante do sistema desenvolvido.

### 4.5.3 Proporção 95%

Aqui serão analisados os resultados que utilizam o valor 0,95 como parâmetro de proporção. Na Tabela 4.5 estão listados os resultados, juntamente com o tempo de execução a partir do Passo 2 até este passo.

Podemos observar que os resultados de acerto na recuperação de imagem obtidos utilizando esta proporção são tão bons quanto os da seção anterior. Aqui é possível notar

Figura 4.23: Teste de Geometria Epipolar - Imagem 9 com proporção 65%, distância 1, e confiança 0,69.



Fonte: Imagem resultante do sistema desenvolvido.

Tabela 4.5: Teste de Geometria Epipolar - Resultados com proporção 95%

Distância	Confiança		Imagem 1	Imagem 2	Imagem 3	Imagem 4	Imagem 5	Imagem 6	Imagem 7	Imagem 8	Imagem 9
0.5	0.39	Correspondências	19	12	13	502	43	217	66	73	232
		Acerto	Sim	Não	Sim						
		Tempo(s)	122.7	346.6	369	408.8	154.7	205.3	229.3	251.1	207.5
	0.69	Correspondências	19	12	13	502	43	217	66	73	232
		Acerto	Sim	Não	Sim						
		Tempo(s)	122.6	345.8	369.4	407.5	154.7	205.6	229.3	251	206.9
	0.99	Correspondências	19	12	13	502	43	217	66	73	232
		Acerto	Sim	Não	Sim						
		Tempo(s)	123.4	345.5	368.9	407.3	154.1	204.7	229.1	249.9	207
1	0.39	Correspondências	28	14	18	621	76	330	122	123	410
		Acerto	Sim	Não	Sim						
		Tempo(s)	123.3	345.8	369.2	410.8	155.8	205.6	229.6	250.3	207.6
	0.69	Correspondências	28	14	18	621	76	330	122	123	410
		Acerto	Sim	Não	Sim						
		Tempo(s)	123.2	345.4	369.7	407.2	154.7	205.6	228.8	250.1	207.1
	0.99	Correspondências	28	14	18	677	76	338	122	123	410
		Acerto	Sim	Não	Sim						
		Tempo(s)	123.3	345.2	368.6	407.9	154.4	207	228.3	250.2	207.1
2	0.39	Correspondências	45	16	35	792	127	453	176	218	615
		Acerto	Sim	Não	Sim						
		Tempo(s)	122.8	345.5	369.8	407.5	154.2	205.8	228.9	250.4	206.8
	0.69	Correspondências	45	16	35	792	127	453	176	218	615
		Acerto	Sim	Não	Sim						
		Tempo(s)	123	346.1	370.9	407.6	155.7	205	228.3	250.3	206.7
	0.99	Correspondências	45	16	35	809	127	480	176	218	630
		Acerto	Sim	Não	Sim						
		Tempo(s)	123.4	345.1	371.8	407.6	154.2	205.7	229.9	249.6	207.1

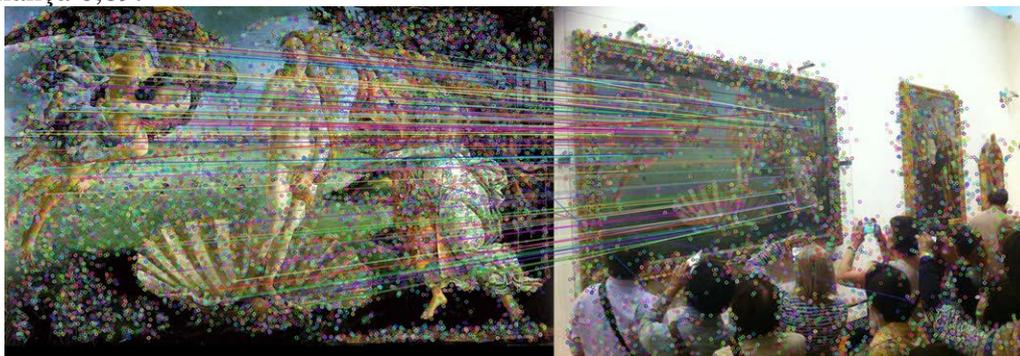
Fonte: Autor

a real capacidade do teste de geometria epipolar, já que se conseguiu bons resultados através do refinamento dos dados obtidos no passo anterior, que foram ruins (ver Seção 4.4). Novamente a Imagem 2 não obteve acertos, entretanto, desta vez obteve algumas correspondências falsas. Nas Figuras 4.24 e Figura 4.25 é possível ver resultados para esta proporção.

#### 4.5.4 Análise Entre as Proporções

Como foi possível ver, para a proporção 35% este método acaba reduzindo demais o número de correspondências. Por outro lado, para as outras proporções obteve bons resultados, mesmo para a proporção 95% que não desempenhou bem nos passos an-

Figura 4.24: Teste de Geometria Epipolar - Imagem 7 com proporção 95%, distância 1, e confiança 0,69.



Fonte: Imagem resultante do sistema desenvolvido.

Figura 4.25: Teste de Geometria Epipolar - Imagem 9 com proporção 95%, distância 1, e confiança 0,69.



Fonte: Imagem resultante do sistema desenvolvido.

teriores. Quanto ao tempo de processamento, este passo não teve influência significativa nos tempos de execução do sistema em relação ao passo anterior.

#### 4.6 Discussão

Com os resultados apresentados neste capítulo, utilizando as diferentes técnicas apresentadas neste trabalho, podemos ver que existe uma grande mudança no desempenho do sistema se variarmos o parâmetro de proporção. No caso do teste de geometria epipolar em que outros dois parâmetros são adicionados, a variação dos mesmos não apresentou mudanças tão expressivas nos resultados.

O sistema foi capaz de identificar as pinturas em diversos dos casos apresentados. Para imagens onde se apresentam condições mais próximas de ideais, podemos ver que a solução mais simples, o teste de proporção, já apresenta bons resultados. Nos casos aonde existem mais elementos que interferem na identificação, técnicas mais avançadas são necessárias. Dentre estas, o teste de simetria se mostrou bastante preciso, porém,

pela diferença inexistente no tempo de execução entre este e o teste de geometria epipolar, recomenda-se utilizar o último, pois consegue obter melhor precisão mesmo com a variação do parâmetro de proporção.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho apresentamos um sistema capaz de realizar a recuperação de pinturas através de fotografias digitais. Estas fotografias podem ser provenientes de diferentes dispositivos, e nem sempre foram concebidas em condições ideais, aonde a pintura está representada de forma clara na imagem. Para que isso fosse possível, foram analisados trabalhos relacionados em busca de métodos de recuperação de imagens através de conteúdo, aonde foi escolhido o detector SURF por se adequar melhor às situações de uso desta aplicação. Para que fosse possível eliminar ocorrências de correspondências falsas e melhorar os resultados, foram selecionadas técnicas de filtragem dos resultados, como o teste de proporção, o teste de simetria, e o teste de geometria epipolar.

Com a análise dos resultados provenientes da execução do sistema CBIR implementado neste trabalho, podemos concluir que o sistema é capaz de recuperar a pintura correta através de uma imagem de entrada, mesmo com imagens de entrada em condições não ideais. Com isto, este trabalho conseguiu verificar que os algoritmos de filtragem encontrados em (LAGANIÈRE, 2011) são capazes de identificar e recuperar pinturas, mesmo com mudanças em seus parâmetros. Isto ocorre tanto para fotografias em condições ideais, como para fotografias com reflexos, obstruções, informações desnecessárias e em ângulo. Apenas não foi possível recuperar a pintura existente na Imagem 2 (Figura 4.2), que representa um caso extremo, onde a quantidade de informações desnecessárias é muito maior do que a representação da pintura na imagem, assim não restam informações suficientes da pintura para que o detector consiga identificar uma boa quantidade de pontos de interesse dela, para a resolução da imagem.

Dentre as diferentes técnicas, como visto na Seção 4.4, já é possível obter uma boa precisão com o Teste de Simetria. Quanto ao Teste de Proporção, ele obteve bons resultados, mas acaba gerando muitas correspondências falsas. Entretanto, o Teste de Geometria Epipolar se mostrou bastante estável mesmo com as mudanças de proporção, desde que não se utilize uma proporção muito pequena. Sendo então recomendado utilizar o Teste de Geometria Epipolar.

Os tempos de execução do sistema têm pouca variação dentre as diferentes técnicas utilizadas. Porém, são tempos altos para uma aplicação que deve ser utilizada em tempo real. Parte disso se deve à capacidade de processamento do computador utilizado para rodar os testes. Como trabalho futuro está o aumento na eficiência do algoritmo, para diminuir o tempo de execução. Outra implementação futura seria o desenvolvimento

de um aplicativo para dispositivos móveis, capaz de enviar imagens para um servidor no qual este sistema estaria rodando e receber como resposta a obra identificada.

Por fim, vale citar que durante o desenvolvimento deste projeto foi estudado um acordo de cooperação com a Fundação Iberê Camargo, aonde seriam providas imagens das pinturas contidas em seu acervo digital, juntamente com fichas técnicas das mesmas. Estas imagens seriam utilizadas para popular o banco de dados, e as imagens de entrada seriam capturadas em uma visita a sede da fundação, das obras em exposição. Isto não foi possível devido ao tempo necessário para resolver as questões burocráticas envolvidas na cooperação, porém, isto pode ser concretizado em implementações futuras.

## REFERÊNCIAS

- APPLE. **LLVM Compiler Overview**. 2012. <https://developer.apple.com/library/mac/documentation/>.
- APPLE. **OS X - What is OS X - Apple**. 2016. <http://www.apple.com/osx/what-is/>.
- APPLE. **Xcode - IDE - Apple Developer**. 2016. <https://developer.apple.com/xcode/ide/>.
- BANNOUR, H.; HLAOUA, L.; AYEB, B. el. Survey of the adequate descriptor for content-based image retrieval on the web: Global versus local features. In: **CORIA**. LSIS-USTV, 2009. p. 445–456. ISBN 2-9524747-1-0. Available from Internet: <<http://dblp.uni-trier.de/db/conf/coria/coria2009.html#BannourHA09>>.
- BANSAL, M.; DANIILIDIS, K. Joint spectral correspondence for disparate image matching. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2013. p. 2802–2809.
- BAY, H. et al. Speeded-up robust features (surf). **Computer Vision and Image Understanding**, v. 110, n. 3, p. 346 – 359, 2008. ISSN 1077-3142. Similarity Matching in Computer Vision and Multimedia. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S1077314207001555>>.
- BEN-KIKI, O.; EVANS, C.; NET, I. dot. **YAML Specification Index**. 2016. <http://yaml.org/spec/>.
- BRUIJN, W. d. **Admiring Sunflowers by Vincent van Gogh**. 2013. <https://www.flickr.com>. Some rights reserved by Wouter de Bruijn.
- CARDOSO, N. **Boticelli, The birth of Venus, Galleria degli Uffizi, Florence**. 2015. <https://www.flickr.com>. Some rights reserved by NunoCardoso.
- HARDEN, M. **The Artchive**. 2016. <http://artchive.com/>.
- HAUAGGE, D. C.; SNAVELY, N. Image matching using local symmetry features. In: IEEE. **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**. [S.l.], 2012. p. 206–213.
- ITSEEZ. **OpenCV Website**. 2015. <http://opencv.org/about.html>.
- JOHNSTONE, A. **The Birth of Venus, Uffizi**. 2014. <https://www.flickr.com>. Some rights reserved by AJoStone.
- JUAN, L.; GWON, O. A comparison of SIFT, PCA-SIFT and SURF. **International Journal of Image Processing (IJIP)**, v. 3, n. 4, p. 143–152, 2009. Available from Internet: <<http://www.cscjournals.org/csc/manuscript/Journals/IJIP/volume3/Issue4/IJIP-51.pdf>>.
- KAMPHUIS, J. **Paris - Musée du Louvre - La Joconde / Portrait de Mona Lisa**. 2008. <https://www.flickr.com>. All rights reserved by glanerbrug.info.
- KATO, T. Database architecture for content-based image retrieval. In: **Proceedings of SPIE Image Storage and Retrieval Systems**. San Jose, CA, USA: SPIE, 1992. p. 112–123.

LAGANIÈRE, R. **OpenCV 2 Computer Vision Application Programming Cookbook**. Packt Publ., 2011. (Packt open source). ISBN 9781849513241. Available from Internet: <<https://books.google.com.br/books?id=REmTmAEACAAJ>>.

LOWE, D. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691. Available from Internet: <<http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94>>.

LUONG, Q.-T.; FAUGERAS, O. D. The fundamental matrix: Theory, algorithms, and stability analysis. **International journal of computer vision**, Springer, v. 17, n. 1, p. 43–75, 1996.

MAGGIE, M. **Botticelli Birth of Venus Uffizi Gallery, Florence Italy**. 2010. <https://www.flickr.com>. All rights reserved by Muggy Maggie.

MCKELVEY, D. **The Mona Lisa, Salle des Etats, Denon Wing, Musée du Louvre, Paris**. 2013. <https://www.flickr.com>. Some rights reserved by David McKelvey.

NIELS. **De slaapkamer, Vincent van Gogh (1888)**. 2009. <https://www.flickr.com>. Some rights reserved by Pachango.

OPENCV. **OpenCV - Feature Matching**. 2014. [http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_matcher/py\\_matcher.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html).

SMEULDERS, A. W. M. et al. Content-based image retrieval at the end of the early years. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 12, p. 1349–1380, Dec 2000. ISSN 0162-8828.

TERRIBERRY, T. B.; FRENCH, L. M.; HELMSEN, J. Gpu accelerating speeded-up robust features. In: CITESEER. **Proc. Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT)**. [S.l.], 2008. p. 355–362.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: A survey. **Found. Trends. Comput. Graph. Vis.**, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 3, p. 177–280, jul. 2008. ISSN 1572-2740. Available from Internet: <<http://dx.doi.org/10.1561/06000000017>>.

WAGENAAR, M. **Vincent van Gogh 1889 Sunflowers**. 2009. <https://www.flickr.com>. Some rights reserved by Minke Wagenaar.

WELLINGTONANDSQUEAK. **Why is she smiling???** 2015. <https://www.flickr.com>. All rights reserved by wellingtonandsqueak.