



**Universidade de Brasília  
Faculdade de Tecnologia**

**Otimização de algoritmo de rastreamento de  
agulhas flexíveis em imagens de ultrassom  
para aplicações a tempo de execução em  
robótica cirúrgica**

Ada Carine Santos de Oliveira

TRABALHO DE GRADUAÇÃO  
CONTROLE E AUTOMAÇÃO

Brasília  
2022

**Universidade de Brasília  
Faculdade de Tecnologia**

**Otimização de algoritmo de rastreamento de  
agulhas flexíveis em imagens de ultrassom  
para aplicações a tempo de execução em  
robótica cirúrgica**

Ada Carine Santos de Oliveira

Trabalho de Graduação submetido como re-  
quisito parcial para obtenção do grau de Enge-  
nheiro em Controle e Automação.

Orientador: Prof. Dr. Claudia Patricia Ochoa Diaz

Brasília  
2022

S237o Santos de Oliveira, Ada Carine.  
Otimização de algoritmo de rastreamento de agulhas flexíveis em imagens de ultrassom para aplicações a tempo de execução em robótica cirúrgica / Ada Carine Santos de Oliveira; orientador Claudia Patricia Ochoa Diaz. -- Brasília, 2022.  
78 p.

Trabalho de Graduação em Controle e Automação -- Universidade de Brasília, 2022.

1. Ultrassom. 2. Agulhas Flexíveis. 3. Robótica Cirúrgica. 4. Aplicação Online. I. Ochoa Diaz, Claudia Patricia, orient. II. Título

**Universidade de Brasília  
Faculdade de Tecnologia**

**Otimização de algoritmo de rastreamento de agulhas flexíveis em imagens de ultrassom para aplicações a tempo de execução em robótica cirúrgica**

Ada Carine Santos de Oliveira

Trabalho de Graduação submetido como requisito parcial para obtenção do grau de Engenheiro em Controle e Automação.

Trabalho aprovado. Brasília, 11 de maio de 2022:

---

**Prof. Dr. Claudia Patricia Ochoa Diaz,**  
**UnB/FGA**  
Orientadora

---

**Prof. Dr. Roberto de Souza Baptista,**  
**UnB/FGA**  
Examinador interno

---

**Prof. Dr. Guillermo Alvarez Bestard,**  
**UnB/FGA**  
Examinador interno

Brasília  
2022

# Agradecimentos

À Carmen Célia Santos de Oliveira, por ser meu maior amor, inspiração e fonte de coragem diária.

À Antonia do Socorro Santos, por ser minha segunda mãe e encher minha vida de amor e bondade.

À Maria do Carmo Silva Santos, a eterna fonte da doçura que me habita.

À Maria das Dores Borges, um dos meus maiores exemplos de força.

A Ana Elisa, João Pedro, Adão Júnior e Kilwson Fernandes, meus irmãos tão amados, parceiros de caminhada e a verdadeira herança que tenho.

À Francisca Maria Santos, por sempre me dizer que estudar é amargo, mas os frutos são doces.

A Djalma Fernandes, por me acolher de braços abertos.

A Daniel Ervilha Lopes, por sempre ser apoio e força incondicionais.

A Cecília Maria, Letícia Ploeg, Bárbara Peres, Isabella Bianchi, Bianca Sales, Amanda Karolyne, Gustavo Almeida, Elmar Brígido, Jorge Simeão, Sarah Albuquerque, Thiago Holanda, Ricardo Bauchspiess, Danilo Li, Artur Pio, Thais Eillen, Thainá Hellen, Alessandra Ribeiro, Ravena Leite, Lorena Diniz, Andreza Bona, Bruno Andreggetti, Sara Gomes, Amanda Alina, Iracema Mendes, Rebeca Sousa e Gabriel Pires, meus amigos tão amados, cada um especial e fundamental a seu modo.

A Gabriela Haya e Nathalia Henriques, pelo suporte para chegar até aqui.

A Mariana Bernardes e Claudia Ochoa, por toda a orientação, compreensão e afeto durante todo este processo.

À Lisbela, por fazer de tudo mais leve e cheio de carinho.

E a Adão Borges de Oliveira... Não satisfeito com tudo que fez por mim em vida, me mostra diariamente que o amor é maior que tudo. Transcende a vida, transcende a presença física, transcende a dor, transcende a saudade.

Muito obrigada!

# Resumo

Este trabalho visa a otimização de um sistema já existente de rastreamento da posição da ponta de uma agulha flexível em procedimentos clínicos ou cirúrgicos minimamente invasivos. São utilizadas imagens da movimentação de uma agulha flexível em *Phantoms*, simuladores de tecido biológico, adquiridas por meio de um sistema de imageamento de ultrassom com interface de pesquisa, conectado via Ethernet a um computador periférico com aplicação cliente. Ao longo deste trabalho, são propostas melhorias às técnicas de segmentação de imagens outrora implementadas. Também faz-se o uso de uma filtragem estocástica via Filtro de Kalman para que sejam geradas estimativas mais precisas, suaves e, principalmente, robustas. Por fim, leva-se em consideração questões tangentes ao custo computacional do projeto, visando torná-lo compatível com uma aplicação online.

**Palavras-chave:** Ultrassom. Agulhas Flexíveis. Robótica Cirúrgica. Aplicação Online.

# Abstract

This work aims to optimize an existing position tracking system of the tip of a flexible needle in minimally invasive clinical or surgical procedures. Images of the movement of a flexible needle in Phantoms, biological tissue simulators, are used, acquired through an ultrasound imaging system with research interface, connected via Ethernet to a peripheral computer with a client application. Throughout this work, improvements are suggested to the techniques of segmentation of images formerly implemented. A Kalman Filter is also implemented to generate more accurate, smoother and, above all, robust estimates. Finally, consideration is given to issues related to the cost computation of the project, aiming to make it compatible with an online application.

**Keywords:** Ultrasound. Flexible Needles. Surgical Robotics. Online Application.

# Lista de ilustrações

Figura 1 – Sistema Robótico Zeus. . . . .	14
Figura 2 – Aplicação do filtro de mediana em uma imagem para máscaras de tamanho 3, 5, 9, 15 e 35, respectivamente, sobre uma imagem de $500 \times 500$ pixels. . . . .	19
Figura 3 – Para $c = 1$ e valores de $\gamma$ diferentes, curvas da Equação $s = c \times r^\gamma$ . . . . .	21
Figura 4 – Imagem $A$ , elemento estruturante $B$ e o resultado da erosão de $A$ por $B$ . . . . .	22
Figura 5 – Imagem $A$ , elemento estruturante $B$ e o resultado da dilatação de $A$ por $B$ . . . . .	23
Figura 6 – Abertura e fechamento morfológicos. . . . .	24
Figura 7 – Segmentação por <i>watershed</i> . . . . .	26
Figura 8 – Representação ecográfica de uma agulha flexível imersa em simulador translúcido de tecido biológico, contendo lóbulos laterais como artefatos. . . . .	30
Figura 9 – Representação ecográfica de uma agulha flexível imersa em simulador fibroso de tecido biológico, contendo uma sombra acústica como artefato. . . . .	30
Figura 10 – Representação ecográfica de uma agulha flexível imersa em banho d’água, contendo uma cauda de cometa formada por artefatos de reverberação. . . . .	31
Figura 11 – Ciclo de filtro de Kalman discreto em andamento. . . . .	33
Figura 12 – Equipamento de ultrassom SonixTOUCH Research. . . . .	36
Figura 13 – Transdutores . . . . .	36
Figura 14 – Esquemático do sistema de imageamento via ultrassom. . . . .	38
Figura 15 – Exemplos de simuladores de tecido biológico de alta fidelidade. . . . .	39
Figura 16 – <i>Phantoms</i> de banho d’água, gelatina translúcida e gelatina opaca. . . . .	40
Figura 17 – Resumo do processo de estimação da ponta da agulha. . . . .	41
Figura 18 – Imagem de ultrassom representativa da posição de uma agulha flexível em phantom opaco, com forte presença de ruído do tipo <i>speckle</i> . . . . .	43
Figura 19 – Imagem de ultrassom representativa da posição de uma agulha flexível em phantom translúcido. . . . .	44
Figura 20 – Resumo do processo de estimação da ponta da agulha após propostas de alteração. . . . .	51
Figura 21 – Regiões de interesse de dimensões $200 \times 200$ , $150 \times 150$ e $100 \times 100$ pixels de imagem adquirida em <i>phantom</i> opaco . . . . .	54
Figura 22 – Aplicação de filtros de mediana e filtros bilaterais em região de interesse de imagem adquirida em <i>phantom</i> opaco. . . . .	55
Figura 23 – Aplicação da correção gama para valores de $\gamma$ diferentes em imagem de <i>phantom</i> opaco submetida a filtro de mediana. . . . .	57
Figura 24 – Aplicação da correção gama para valores de $\gamma$ diferentes em imagem de <i>phantom</i> opaco submetida a filtro bilateral. . . . .	58



Figura 25 – Aplicação de gradiente para realce de bordas após correção gama. . . . .	59
Figura 26 – Execução da operação de limiarização, que precede a segmentação da imagem. . . . .	59
Figura 27 – Execução de <i>Watershed</i> . . . . .	60
Figura 28 – Execução de Watershed via componentes conectados. . . . .	61
Figura 29 – Execução de <i>Watershed</i> baseado em código de cadeia. . . . .	62
Figura 30 – Rastreamento Eletromagnético da ponta da agulha utilizando o sensor NDI Aurora. . . . .	63
Figura 31 – Medição simultânea do sensor eletromagnético, da transformada de Hough e do máximo local de cinza da posição da ponta da agulha em um simulador de gelatina translúcida. . . . .	64
Figura 32 – Resumo do processo final de estimação da ponta da agulha. . . . .	73

# Lista de tabelas

Tabela 1 – Velocidade de propagação do som em alguns materiais e tecidos. . . . .	28
Tabela 2 – Cabeçalho de uma mensagem transmitida por meio do protocolo OpenIGTLink. . . . .	38
Tabela 3 – Operações executadas com as imagens adquiridas a partir de cada phantom.	41
Tabela 4 – Tabela de erros de medição do ensaio parado para ROI de $200 \times 200$ pixels.	65
Tabela 5 – Tabela de erros de medição do ensaio horizontal para ROI de $200 \times 200$ pixels. . . . .	65
Tabela 6 – Tabela de erros de medição do ensaio vertical para ROI de $200 \times 200$ pixels.	66
Tabela 7 – Tabela de erros de medição do ensaio parado para ROI de $150 \times 150$ pixels.	66
Tabela 8 – Tabela de erros de medição do ensaio horizontal para ROI de $150 \times 150$ pixels. . . . .	66
Tabela 9 – Tabela de erros de medição do ensaio vertical para ROI de $150 \times 150$ pixels.	67
Tabela 10 – Tabela de erros de medição do ensaio parado para ROI de $100 \times 100$ pixels.	67
Tabela 11 – Tabela de erros de medição do ensaio horizontal para ROI de $100 \times 100$ pixels. . . . .	67
Tabela 12 – Tabela de erros de medição do ensaio vertical para ROI de $100 \times 100$ pixels.	68
Tabela 13 – Tempo de execução da definição de regiões de interesse. . . . .	69
Tabela 14 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de $200 \times 200$ pixels. . . . .	70
Tabela 15 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de $150 \times 150$ pixels. . . . .	70
Tabela 16 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de $100 \times 100$ pixels. . . . .	71
Tabela 17 – Tabela comparativa do tempo de execução do algoritmo antes das modificações e após as mesmas para nova ROI de $150 \times 150$ pixels. . . . .	72
Tabela 18 – Tabela comparativa do tempo de execução do algoritmo antes das modificações e após as mesmas para nova ROI de $100 \times 100$ pixels. . . . .	72

# Lista de abreviaturas e siglas

CPU	Unidade Central de Processamento . . . . .	37
CTA	Cauda de Cometa . . . . .	31
FPS	<i>Frames Per Second</i> . . . . .	51
LARA	Laboratório de Robótica e Automação . . . . .	15
OpenCV	<i>Open Source Computer Vision Library</i> . . . . .	42
PLUS	<i>Public Software Library for Ultrasound</i> . . . . .	38
ROI	<i>Region Of Interest</i> . . . . .	16
SDK	<i>Software Development Kit</i> . . . . .	37
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> . . . . .	37
UnB	Universidade de Brasília . . . . .	15
URI	<i>Ultrasound Research Interface</i> . . . . .	35

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Contextualização</b>	<b>13</b>
<b>1.2</b>	<b>Definição do problema</b>	<b>14</b>
<b>1.3</b>	<b>Objetivos do projeto</b>	<b>16</b>
<b>1.4</b>	<b>Apresentação do manuscrito</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
<b>2.1</b>	<b>Processamento Digital de Imagens</b>	<b>17</b>
2.1.1	Processos de baixo nível	18
2.1.1.1	Filtros espaciais de suavização	18
2.1.1.2	Transformações de intensidade	20
2.1.1.3	Filtros espaciais de aguçamento	20
2.1.2	Processos de nível médio	22
2.1.2.1	Erosão e Dilatação	22
2.1.2.2	Abertura e Fechamento	23
2.1.2.3	Limiarização	24
2.1.3	Processos de alto nível	25
2.1.3.1	<i>Watershed</i>	26
2.1.3.2	Aprendizado de máquina	27
<b>2.2</b>	<b>Imagens de ultrassom</b>	<b>27</b>
2.2.1	Princípios básicos	27
2.2.2	Artefatos	28
2.2.2.1	Lóbulos Laterais	29
2.2.2.2	Sombra acústica	30
2.2.2.3	Reverberação	31
<b>2.3</b>	<b>Filtro de Kalman</b>	<b>31</b>
2.3.1	Princípios básicos	32
2.3.2	Filtro de Kalman Discreto	32
2.3.2.1	Predição	33
2.3.2.2	Correção	33
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>35</b>
<b>3.1</b>	<b>Sistema de Imageamento Por Ultrassom</b>	<b>35</b>
3.1.1	Equipamento de ultrassom	35
3.1.2	Sistema de aquisição de imagens	37
3.1.3	Phantoms	38

<b>3.2</b>	<b>Pré-processamento</b>	<b>41</b>
3.2.1	Definição da ROI	42
3.2.2	Filtros de suavização	42
3.2.3	Correções de contraste	43
3.2.4	Operações morfológicas	44
<b>3.3</b>	<b>Segmentação</b>	<b>45</b>
3.3.1	Algoritmo de segmentação	45
3.3.1.1	<i>Watershed</i> Tradicional	45
3.3.1.2	<i>Watershed</i> baseado em componentes conectados	46
3.3.1.3	<i>Watershed</i> baseado em código de cadeia	48
<b>3.4</b>	<b>Estimação da posição da ponta da agulha</b>	<b>50</b>
3.4.1	Filtro de Kalman	50
<b>3.5</b>	<b>Custo Computacional</b>	<b>51</b>
<b>4</b>	<b>RESULTADOS</b>	<b>53</b>
<b>4.1</b>	<b>Análise qualitativa das técnicas de processamento de imagens</b>	<b>53</b>
4.1.1	Pré-processamento	54
4.1.1.1	Filtros de suavização	54
4.1.1.2	Operações de realce	56
4.1.1.2.1	Correção gama	56
4.1.1.2.2	Gradiente	58
4.1.2	Segmentação	59
4.1.2.1	<i>Watershed</i> Tradicional	59
4.1.2.2	<i>Watershed</i> baseado em componentes conectados	60
4.1.2.3	<i>Watershed</i> baseado em código de cadeia	61
<b>4.2</b>	<b>Eficácia de medição</b>	<b>62</b>
<b>4.3</b>	<b>Custo Computacional</b>	<b>68</b>
4.3.1	Aquisição e transmissão de imagens	69
4.3.2	Diminuição da ROI	69
4.3.3	Medição dos tempos de execução das tarefas	69
<b>5</b>	<b>CONCLUSÕES</b>	<b>74</b>
<b>5.1</b>	<b>Perspectivas Futuras</b>	<b>75</b>
	<b>REFERÊNCIAS</b>	<b>76</b>

# 1 Introdução

## 1.1 Contextualização

A palavra "percutâneo" é oriunda do latim "*per cutis*" e, em tradução literal, quer dizer "através da pele". Em procedimentos percutâneos, o acesso a órgãos e tecidos se dá por meio de punções através da pele, feitas por objetos tubulares bem finos, como agulhas ou cateteres. Os procedimentos percutâneos mais conhecidos são, certamente, a aplicação de medicação intravenosa e coleta de sangue, mas pode-se citar exemplos, também, de biópsias e várias modalidades de cirurgias renais (ARVELLOS, 2011), ortopédicas (ORTOPEDIA ESPECIALIZADA, 2018), oncológicas (RANGEL; TELLES, 2012) e em outras diversas áreas da Medicina.

As aplicações cirúrgicas desse tipo de intervenção são uma alternativa muito eficiente quando comparadas às convencionais cirurgias abertas, nas quais ocorrem a exposição dos órgãos, grandes cicatrizes remanescentes e maiores tempos de internação e recuperação. Sua aplicabilidade tem o potencial de grande expansão com o uso de novas ferramentas como as agulhas flexíveis, cuja necessidade é originária de eventuais dificuldades de acesso ao órgão a ser tratado. Muitas vezes, outros órgãos, ossos ou vasos podem atuar como obstáculos dificultando o acesso e, nesse caso, agulhas rígidas não são uma opção viável. Agulhas flexíveis tem a capacidade de realizar trajetórias curvilíneas dentro do corpo humano e desviar de tais obstáculos, preservando ao máximo os tecidos saudáveis. Além disso, por realizarem o mínimo de interferência no organismo, resultam em menores cicatrizes remanescentes e também menores tempos de internação, melhorando a qualidade de vida do paciente no pós-operatório e reduzindo os riscos de eventuais infecções hospitalares.

Entretanto, agulhas flexíveis requerem o uso de equipamento robótico para sua inserção. Fazê-los manualmente torna-se algo de grande complexidade devido à precisão que esse tipo de procedimento demanda, dificuldade de acesso a órgãos e tecidos alvo e visibilidade comprometida. O avanço dessa tecnologia deve ser, portanto, aliado ao avanço da robótica cirúrgica. Robôs são capazes de realizar tarefas com rapidez, precisão e excelente repetibilidade, além de não terem como variáveis fatores de erro intrinsecamente humanos, como stress, cansaço, fadiga e condições ambientais.

As últimas décadas foram de muitos avanços na área de robótica cirúrgica. Um exemplo disso é o sistema robótico ZEUS (ETO; NAITO, s.d.), constituído de um sistema mestre-escravo subdividido em dois subsistemas separados, *surgeon-side* ou "lado do cirurgião" e *patient-side*, ou "lado do paciente", vistos na Figura 1. O subsistema do cirurgião recebe os comandos do cirurgião via console e o subsistema do paciente inclui dois braços ro-

bóticos e o sistema de endoscópio automatizado para posicionamento ideal. Desse modo, em 2001 este sistema foi capaz de realizar uma colecistectomia laparoscópica em animais tendo sua central de controle em Nova York, nos Estados Unidos, e os animais em Estrasburgo, na França (MARESCAUX, 2001). Desde então, a laparoscopia se popularizou bastante e tem uma série de aplicações entre humanos.



Figura 1 – Sistema Robótico Zeus.

Fonte: (ETO; NAITO, s.d.)

Assim, aliando a robótica cirúrgica a procedimentos percutâneos, é necessário combinar a inserção da agulha com algum sistema de rastreamento automatizado para fechar a malha de controle. Este rastreamento é normalmente associado a sistemas de imageamento, que permitem uma boa visualização da agulha e do meio em que se encontra.

No trabalho desenvolvido em (MELLO, G. G. N. de et al., 2001), por exemplo, faz-se um estudo sobre procedimentos percutâneos orientados por imagem no tratamento de coleções hepáticas, que são abscessos de pus ou de material infectado geralmente decorrentes de uma infecção localizada no fígado. Nele, compara-se punções puramente aspirativas com drenagens percutâneas orientadas por ultrassonografia, tomografia computadorizada e ressonância magnética. Em coleções submetidas exclusivamente à punção aspirativa, obteve-se sucesso em 57,1%, enquanto nas submetidas à drenagem percutânea orientada o sucesso foi de 100%, comprovando a eficácia desse tipo de abordagem.

## 1.2 Definição do problema

No presente trabalho, focar-se-á no rastreamento de agulhas flexíveis por meio de imagens de ultrassom. Exemplos dessa modelagem podem ser encontrados nas referências (HONG, 2004) e (MATIAS, 2012). As vantagens do imageamento por ultrassom são várias, sendo algumas delas, por exemplo, o fornecimento de imagens em tempo real, a ausência de radiação, o que gera mais segurança ao paciente, baixo custo relativo às demais tecnologias de imagem e a ausência de restrições relevantes no tocante aos materiais utilizados em

dispositivos de inserção. Em sistemas de imageamento por raio X e ressonância magnética, estruturas metálicas comprometem a boa realização dos procedimentos.

No entanto, há também uma grande desvantagem: se comparado a outros métodos de aquisição de imagens, como ressonância magnética e tomografia computadorizada, a aquisição de imagens via ultrassom apresenta uma baixa resolução de imagem, ainda contando esta com ruídos e uma série de artefatos intrínsecos à tecnologia de ultrassom que precisam ser tratados. Desse modo, as imagens adquiridas devem passar por ajustes de modo que as informações necessárias sejam corretamente extraídas.

Em trabalhos anteriores, baseados em (MATIAS, 2012) e (GERALDES, 2015), foram propostos métodos para o rastreamento da ponta da agulha flexível em imagens de ultrassom durante inserções em simuladores de tecido biológico. Entretanto, não foram levadas em consideração restrições de tempo computacional para o rastreamento intraoperatório que tornem o projeto compatível com uma aplicação online. Deve-se, portanto, buscar o compromisso entre a rapidez do rastreamento e a precisão demandada quando lida-se com vidas. Em procedimentos percutâneos, a depender do órgão, a precisão necessária é da ordem de milímetros ou até mesmo micromilímetros para que seja preservada a integridade de órgãos e tecidos vitais (ABOLHASSANI, 2007) (NEEDLE..., 2007).

O trabalho aqui proposto pode ser descrito como um projeto de otimização de uma aplicação já existente baseada em (MATIAS, 2012) e (GERALDES, 2015), descrita em (KIYOKU, 2015) e já sequenciada em (S. DOS REIS, 2019). O sistema de rastreamento da ponta da agulha flexível foi desenvolvido no Laboratório de Robótica e Automação da Universidade de Brasília (LARA UnB) e é composto por um sistema de imageamento de ultrassom conectado via Ethernet a um computador periférico, no qual ocorre o processamento das imagens adquiridas.

Em (KIYOKU, 2015), foram desenvolvidas as técnicas de aquisição, transmissão e processamento das imagens da movimentação da agulha em simuladores de tecido biológico. Entretanto, não foram considerados requisitos temporais e comprovou-se que o algoritmo utilizado para a segmentação das imagens é demasiadamente lento e, portanto, incompatível com um processamento online de imagens.

Posteriormente, o trabalho foi sequenciado em (S. DOS REIS, 2019) e, para melhorar a estimação da ponta da agulha, introduziu-se um Filtro de Kalman ao algoritmo para ajustes de posição e predição de posições futuras, além de novos algoritmos de estimação da ponta da agulha. Entretanto, apesar da melhoria na estimação da posição da ponta da agulha e de algumas considerações referentes ao tempo de execução, não foram implementadas soluções que reduzissem o custo computacional do projeto de modo significativo e compatível com um processamento a tempo de execução.



## 1.3 Objetivos do projeto

O objetivo deste trabalho é reduzir o custo computacional desse sistema e torná-lo mais rápido, desenvolvendo técnicas que permitam que o rastreamento da ponta da agulha acompanhe a aquisição de imagens já existente, permitindo que as imagens adquiridas sejam lidas, pré-processadas, segmentadas, a posição da ponta da agulha seja estimada e este cálculo seja retornado a tempo de execução. O processamento já acontece, em sua grande maioria, em escala de cinza, o que já diminui o custo computacional por permitir o trabalho com apenas um canal de 8 bits por pixel, ao invés de 3 canais. Além disso, deseja-se minimizar a interferência do usuário no processo, tornando automáticos processos que antes eram fixos e manuais. A otimização deste projeto a fim de executar o processamento de imagens em tempo de execução é feita, portanto, através das seguintes tarefas:

1. Cálculo dos tempos de aquisição e transmissão de imagens, outrora desprezados;
2. Substituição do algoritmo *Watershed* utilizado na segmentação de imagens por alternativas que reduzam o custo computacional do projeto;
3. Automatização da definição da Região de Interesse da imagem (ROI - *Region Of Interest*), utilizando-se da estimação da posição da ponta da agulha para isso;
4. Diminuição da quantidade de frames de imagem processados, considerando o *trade-off* entre precisão e rapidez do processo;
5. Análise da filtragem e do cálculo final da posição da ponta da agulha.

## 1.4 Apresentação do manuscrito

O Capítulo 2 conta com a fundamentação teórica da formação de imagens de ultrassom, as principais técnicas de processamento de imagem utilizadas e a teoria de Filtro de Kalman. O Capítulo 3 apresenta as fases de desenvolvimento do trabalho, desde as propostas de alteração no algoritmo de processamento de imagens à implementação do Filtro de Kalman e delimitação da ROI. No Capítulo 4, analisa-se os resultados obtidos com o desenvolvimento proposto e no Capítulo 5 tem-se as conclusões feitas e perspectivas futuras do projeto.

## 2 Fundamentação Teórica

Neste capítulo são introduzidos os conceitos necessários à compreensão e ao desenvolvimento do trabalho aqui realizado. Primeiramente, introduz-se conceitos de processamento digital de imagens, necessários ao tratamento de imagens de ultrassom. Em seguida, abordam-se os conceitos que envolvem imagens de ultrassom, destacando os princípios básicos de sua aquisição e a formação de artefatos, que são eventuais defeitos de imagem. Por fim, fala-se sobre filtro de Kalman, que foi utilizado para a melhoria das estimativas no trabalho aqui proposto.

### 2.1 Processamento Digital de Imagens

De acordo com (GONZALEZ, 2010), uma imagem pode ser definida como uma função bidimensional  $f(x, y)$ , sendo  $x$  e  $y$  coordenadas espaciais. A amplitude de  $f$  em qualquer par  $(x, y)$  é chamada de intensidade da imagem, definida em escala de cinza. Tem-se uma imagem digital quando  $x$ ,  $y$  e os valores de intensidade de  $f$  são quantidades finitas e discretas. E cada um desses elementos, tendo localização e intensidades específicas, são os chamados pixels.

Em relação às operações com imagens digitais, existe todo um vasto campo de estudo e aplicação, que se inicia no processamento de imagens de mais baixo nível e segue até a visão computacional e soluções que emulem a visão humana. Os limites entre as áreas de processamento de imagens, análise de imagens e visão computacional são pouco claros, mas imaginando uma linha contínua, partindo do processamento de imagens e encerrando-se na visão computacional, a definição de três tipos de processos computacionais envolvendo imagens é muito útil à compreensão do papel das operações realizadas.

Tem-se, em primeiro lugar, os processos de baixo nível, que consistem em operações básicas que visam melhorias na imagem a fim de facilitar outros processos e extração de informações. Exemplos dessas operações são filtros para a suavização de ruídos e operações de realce. Nesses processos, tanto a entrada quanto a saída são imagens.

Em seguida, processos de nível médio são definidos como aqueles nos quais, a partir de uma imagem de entrada, obtém-se atributos extraídos da imagem em questão como saída, como bordas, contornos e a identidade de objetos. Encaixam-se aqui operações morfológicas, que trabalham diretamente na morfologia da imagem e algumas das primeiras etapas da segmentação, ou seja, a separação de uma imagem em objetos e/ou regiões, e a adequação dessas imagens e/ou regiões ao processamento computacional.

Por fim, processos de alto nível são aqueles que emulam a visão humana, ou seja, dão sentido a objetos reconhecidos. Partindo das etapas finais da segmentação, imagens

de entrada tem como saída informações a serem utilizadas de acordo com a finalidade do projeto.

No presente trabalho, o início se dá por meio da aquisição por meio de um sistema de imageamento de ultrassom associado a um computador periférico. As imagens adquiridas, então, passam por um pré processamento, consistente na aplicação de filtros responsáveis pela redução de ruídos e aumento do contraste. Este processo é realizado de modo a facilitar o processo de identificação de bordas e objetos feito em sequência, realizado através de operações morfológicas e de segmentação para que a agulha seja encontrada na imagem. Por fim, por meio de cálculos, estimações e filtros estocásticos, extraímos informação, obtendo a posição da ponta da agulha em um simulador de tecido biológico.

### 2.1.1 Processos de baixo nível

Operações no domínio espacial são aquelas referentes ao próprio plano da imagem, caracterizadas pela manipulação direta dos valores dos pixels da imagem para a obtenção do efeito desejado. Em filtros espaciais, tem-se o pixel, sua vizinhança (composta pelos pixels vizinhos a ele, geralmente formando um pequeno retângulo, a ser chamado de máscara do filtro, com a dimensão desejada pelo desenvolvedor) e uma operação pré-definida a ser realizada sobre o pixel e sua vizinhança. A partir de uma imagem de entrada, tem-se, portanto, uma nova imagem como saída, gerada pela movimentação da máscara pelos pixels da imagem realizando a operação pré-definida. Esta nova imagem é composta por elementos suavizados ou realçados de acordo com a aplicação desejada, de modo a facilitar novos processos que venham a seguir.

#### 2.1.1.1 Filtros espaciais de suavização

Filtros espaciais de suavização têm por finalidade principal o borramento e a redução de ruído em imagens. É utilizado de modo a suavizar imperfeições, remover pequenos detalhes e ligar eventuais discontinuidades em linhas e curvas antes de realizar operações de mais alto nível, como a extração de bordas, contornos ou objetos.

O mais comum dentre os filtros de suavização não-lineares é o *filtro de mediana*. Seu funcionamento consiste em substituir o valor do pixel central pela mediana dos valores de intensidade dos pixels da vizinhança. Este filtro é muito utilizado por geralmente entregar uma redução de ruído satisfatória com um borramento consideravelmente menor que os de filtros lineares de média com máscaras similares, nos quais o valor de um pixel da imagem é simplesmente substituído pela média dos valores dos pixels de sua vizinhança dentro da máscara. O filtro de mediana é conhecido, também, por se destacar diante da presença de ruídos do tipo sal e pimenta, composto por pontos brancos e pretos sobrepostos.

A eficácia do filtro de mediana na redução de ruídos com menor borramento se dá

pela própria natureza da mediana, que tende a forçar pontos com níveis de intensidade distintos para serem mais semelhantes aos seus vizinhos. Utilizando do exemplo dado em (GONZALEZ, 2010), em uma vizinhança  $3 \times 3$  com valores (10, 20, 20, 20, 15, 20, 20, 25, 100), os valores são ordenados como (10, 15, 20, 20, 20, 20, 20, 25, 100) e a mediana resultante é 20. Desse modo, o filtro favorece agrupamentos de valores de intensidade mais próximos de pixels e tende a cancelar intensidades muito distoantes, eliminando a transição brusca que é o ruído e preservando o valor de intensidade predominante. A Figura 2 mostra o resultado da aplicação do filtro de mediana em uma imagem para máscaras de tamanho 3, 5, 9, 15 e 35, respectivamente, sobre uma imagem de  $500 \times 500$  pixels.

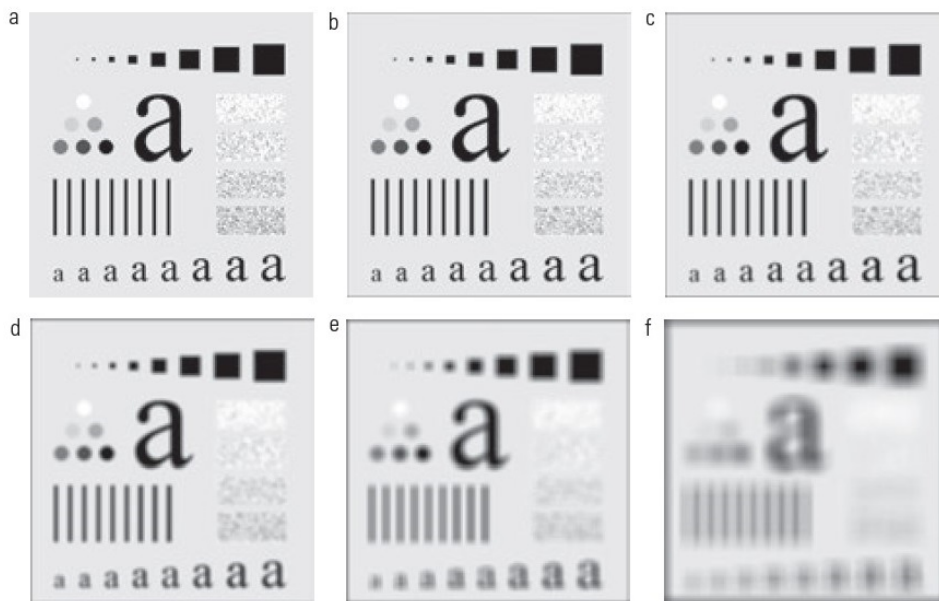


Figura 2 – Aplicação do filtro de mediana em uma imagem para máscaras de tamanho 3, 5, 9, 15 e 35, respectivamente, sobre uma imagem de  $500 \times 500$  pixels.

Fonte: Processamento Digital de Imagens, (GONZALEZ, 2010)

No entanto, apesar do sucesso na suavização de ruídos, existe uma outra transição abrupta que filtros de média ou mediana suavizam e talvez não deveriam: bordas. As bordas dos objetos de uma imagem são, quase sempre, uma característica importante a ser preservada, e um filtro de suavização tende a reduzir a nitidez das mesmas. Para contornar tal problema, esses filtros são comumente associados a operações de aguçamento ou realce que devolvam esses contornos desejados após a eliminação ou redução dos ruídos.

Antes disso, uma outra opção de filtragem eficaz na suavização de ruídos é o *filtro bilateral*. Assim como o filtro de mediana, é um filtro de suavização não linear, mas com superior preservação de bordas e redução de ruído para imagens. Seu funcionamento se dá por meio da substituição da intensidade de cada pixel por uma média ponderada dos valores de intensidade dos pixels próximos.

A filtragem bilateral consiste em uma dupla filtragem gaussiana, sendo a primeira

uma filtragem gaussiana no espaço e a segunda um filtro gaussiano que é uma função da diferença de pixels. A função gaussiana do espaço garante que apenas os pixels próximos sejam considerados para o efeito *blur*, enquanto a função gaussiana da diferença de intensidade garante que apenas pixels com intensidades semelhantes ao pixel central sejam considerados para desfoque. Além disso, dois pixels estão próximos um do outro não apenas se ocuparem localizações espaciais próximas, mas também se tiverem alguma semelhança. Assim, preserva as bordas, pois os pixels presentes nas mesmas não terão grande variação de intensidade. Conta com três parâmetros:

1.  $d$ , correspondente ao diâmetro da vizinhança de um pixel;
2.  $\sigma_{color}$ , correspondente ao valor de  $\sigma$  no espaço de cores. Quanto maior o valor, mais valores de intensidades mais discrepantes tendem a se misturar.
3.  $\sigma_{space}$ , correspondente ao valor de  $\sigma$  no espaço de coordenadas. Quanto maior o seu valor, mais pixels serão misturados.

#### 2.1.1.2 Transformações de intensidade

Transformações de intensidade são operações das mais simples, caracterizadas por um valor  $r$  de intensidade anterior do pixel e um valor  $s$  de intensidade posterior, associados através de uma função  $s = T(r)$ . Os três tipos básicos de transformações de intensidade são linear, logarítmica e de potência.

A *correção gama* é uma transformação de potência representada, de modo simples, pela Equação 2.1 abaixo, sendo  $c > 0$  e  $\gamma > 0$ :

$$s = c \times r^\gamma \quad (2.1)$$

A Figura 3 mostra curvas de intensidade mapeadas pela transformação gama. Percebe-se que, quando  $c = \gamma = 1$ , tem-se a transformação de identidade, na qual a imagem de entrada é igual a imagem de saída. Para  $\gamma < 1$ , é mapeada uma faixa estreita de valores escuros de entrada para uma ampla faixa de valores de saída, e o oposto se dá para valores mais altos de níveis de entrada.

Essa transformação afeta, portanto, o brilho da imagem de saída. Para  $\gamma > 1$ , observa-se que ela realça pontos de maior intensidade de valores dos pixels, o que pode ser interessante para a identificação de objetos mais claros ou brilhantes.

#### 2.1.1.3 Filtros espaciais de aguçamento

Sendo o cálculo de média análogo a integração e responsável por suavizar transições abruptas e reduzir a nitidez da imagem, pode-se assumir que a diferenciação no domínio do

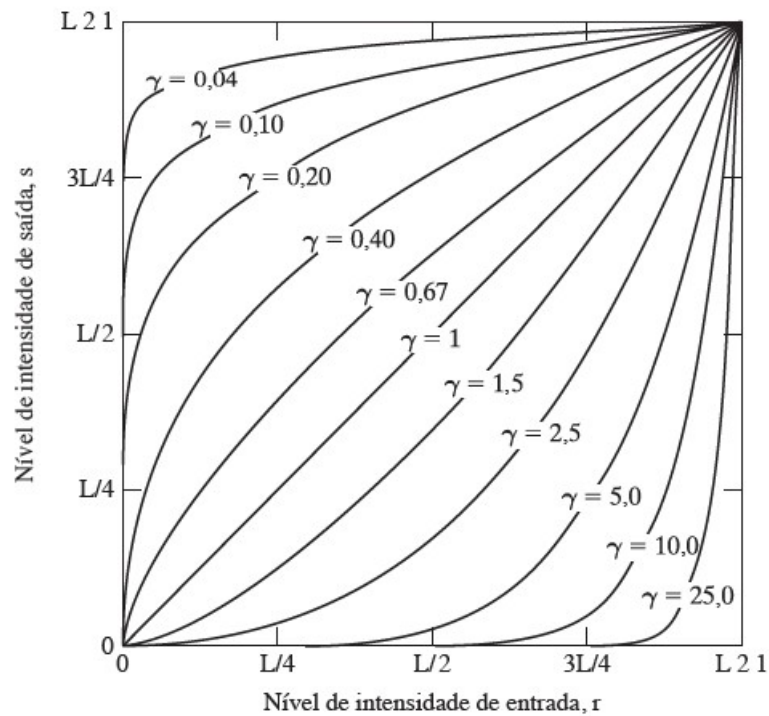


Figura 3 – Para  $c = 1$  e valores de  $\gamma$  diferentes, curvas da Equação  $s = c \times r^\gamma$ .

Fonte: Processamento Digital de Imagens, (GONZALEZ, 2010)

espaço tem o poder de ressaltar bordas e outras transições de intensidade. Desse modo, após filtros de suavização, aplicar filtros espaciais de realce pode ser muito eficaz na recuperação de bordas perdidas e na atenuação de áreas com pouca variação de intensidade (GONZALEZ, 2010).

O gradiente é um filtro espacial de aguçamento que faz uso da derivada de primeira ordem. O gradiente de  $f(x,y)$  no ponto  $(x,y)$  é definido por:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.2)$$

Esse vetor aponta na direção da maior taxa de variação de  $f$  na posição  $(x,y)$ . A partir disso, a magnitude do vetor é o valor em  $(x,y)$  da taxa de variação na direção do vetor gradiente, e é expressa através da seguinte Equação 2.3:

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (2.3)$$

A máscara do filtro é criada, então, a partir da magnitude. A partir deste ponto, a máscara é implementada do mesmo modo que as demais máscaras de processos lineares, percorrendo os pixels da imagem.

## 2.1.2 Processos de nível médio

Após o pré-processamento de imagens, feito majoritariamente com a aplicação de filtros como os acima descritos, tem-se o processo de extração de informação. Este, por sua vez, tem início nas operações morfológicas, que recebem esse nome por fazerem uso de morfologia matemática para alterar a estrutura dos elementos presentes na imagem de modo a extrair componentes úteis na representação de uma região, como bordas, fronteiras e esqueletos. Estas operações tem por base duas operações primárias, chamadas *erosão* e *dilatação*.

Fundamentadas na teoria de conjunto, a erosão e a dilatação têm como entrada uma imagem de duas dimensões e um elemento estruturante, também de duas dimensões. Trata-se de duas operações opostas, a serem descritas a seguir.

### 2.1.2.1 Erosão e Dilatação

Utilizada para eliminar pequenos detalhes desprezíveis, entende-se a erosão da imagem  $A$  por um elemento estruturante  $B$  como o conjunto de todos os pontos  $z$  de forma que  $B$ , transladado por  $z$ , está contido em  $A$ . Isso quer dizer que, ao alinhar o centro de  $B$  com dado pixel em  $A$ , para que este pixel esteja presente em  $z$ ,  $B$  tem que, necessariamente, estar contido em  $A$ . Este processo é descrito pela Equação 2.4 e pode ser melhor visto na Figura 4.

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2.4)$$



Figura 4 – Imagem  $A$ , elemento estruturante  $B$  e o resultado da erosão de  $A$  por  $B$ .

Fonte: Produzido pelo autor.

Contrária à erosão, tem-se na dilatação uma ferramenta para o preenchimento de eventuais lacunas. Entende-se por dilatação de  $A$  pelo elemento estruturante  $B$ , ambos de duas dimensões, o conjunto de todos os deslocamentos  $z$  tal que a reflexão de  $B$  se sobrepõe a  $A$  por pelo menos um elemento. Este processo é descrito pela Equação 2.5 e pode ser melhor visto na Figura 5.

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (2.5)$$

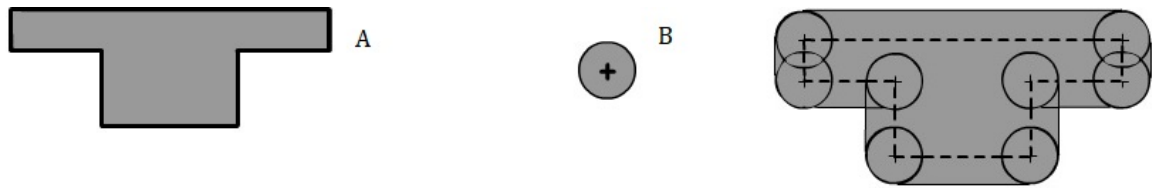


Figura 5 – Imagem  $A$ , elemento estruturante  $B$  e o resultado da dilatação de  $A$  por  $B$ .

Fonte: Produzido pelo autor.

### 2.1.2.2 Abertura e Fechamento

Trata-se de duas importantes operações morfológicas, opostas tais quais erosão e dilatação e resultantes da combinação das duas. A abertura geralmente suaviza o contorno de um objeto, rompe pequenas ligações que provavelmente não deveriam existir e elimina saliências finas. O fechamento também tende a suavizar contornos, mas, por sua vez, geralmente funde discontinuidades, elimina pequenos buracos e preenche as lacunas em um contorno.

A abertura do conjunto  $A$  pelo elemento estruturante  $B$  é definida pela erosão de  $A$  por  $B$  seguida de uma dilatação de  $A$  por  $B$ .

$$A \circ B = (A \ominus B) \oplus B \quad (2.6)$$

Enquanto isso, o fechamento é composto por uma dilatação de  $A$  por  $B$  seguida de uma erosão de  $A$  por  $B$ .

$$A \bullet B = (A \oplus B) \ominus B \quad (2.7)$$

As duas operações podem ser melhor vistas na Figura 6.



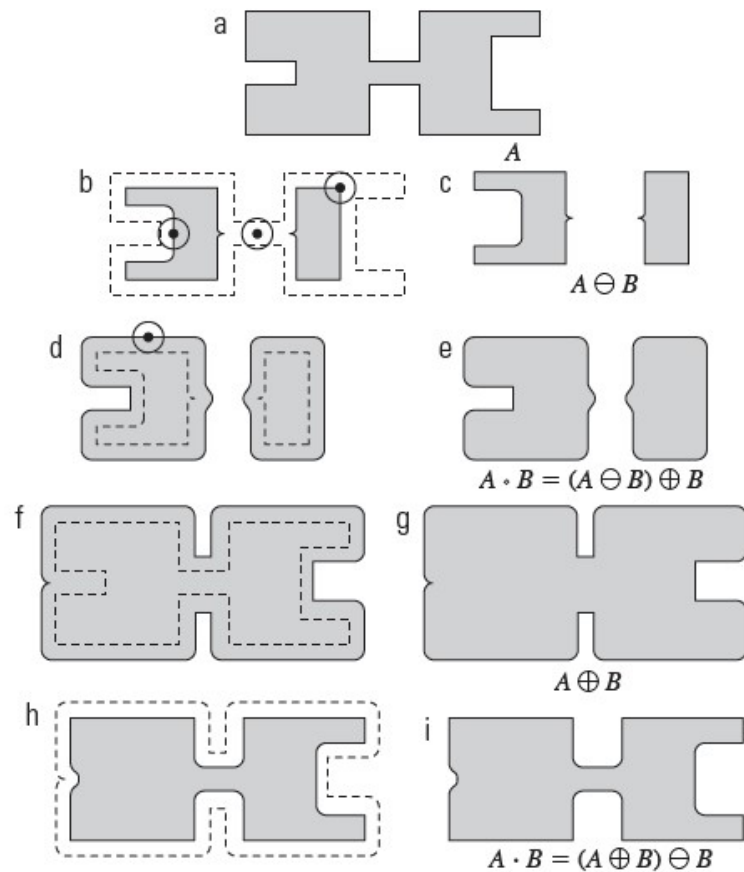


Figura 6 – Abertura e fechamento morfológicos.

Fonte: Processamento Digital de Imagens, (GONZALEZ, 2010)

### 2.1.2.3 Limiarização

Entende-se por limiarização uma técnica de divisão de imagens em regiões tendo por base os valores de intensidade e/ou as propriedades desses valores dos pixels. Em resumo, utilizando dos conceitos presentes em (GONZALEZ, 2010), supondo que uma imagem  $f(x, y)$  seja composta por objetos claros sobre um fundo escuro de tal forma que os pixels do objeto e do fundo tenham valores de intensidade que possam ser agrupados em dois grupos dominantes, este agrupamento pode ser feito através da escolha de um limiar  $T$  responsável por separar estes grupos. A partir da escolha de  $T$ , qualquer ponto  $(x, y)$  na imagem em que  $f(x, y) > T$  é chamado de ponto do objeto. Caso contrário, o ponto é chamado ponto de fundo. Em outras palavras, a imagem segmentada,  $g(x, y)$  é dada por

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (2.8)$$

Dentre as diversas possibilidades de limiarização, destaca-se a *Limiarização de Otsu*. Ela não depende da subjetividade do usuário para a definição de um limiar ótimo para o

agrupamento de objetos, sendo este limiar definido pelas etapas a seguir, de acordo com (GONZALEZ, 2010):

1. Calcular o histograma normalizado da imagem de entrada. Designar os componentes do histograma como  $p_i, i = 1, 2, 3, \dots, L - 1$ ;
2. Calcular as comas acumuladas,  $P_i(k)$  para  $k = 1, 2, 3, \dots, L - 1$ ;
3. Calcular as médias acumuladas,  $m(k)$  para  $k = 1, 2, 3, \dots, L - 1$ ;
4. Calcular a média global,  $m_G$ ;
5. Calcular a variância entre classes  $\sigma_B^2$  para  $k = 1, 2, 3, \dots, L - 1$ ;
6. Obter o limiar de Otsu,  $k^*$  como o valor de  $k^*$ . Se a máxima não for única, obter  $k^*$  através da média dos valores de  $k$  que correspondem aos diversos valores máximos detectados.
7. Obter a medida de separabilidade,  $\eta^*$ , avaliando  $\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$  em  $k = k^*$

### 2.1.3 Processos de alto nível

Após submeter uma imagem de entrada a um pré-processamento à realização de operações morfológicas, o próximo passo em direção à extração de informação de imagens é a segmentação da mesma. Entende-se por segmentação o processo de subdivisão de uma imagem em objetos ou partes constituintes, com nível de detalhamento a depender da imagem e do problema a ser resolvido. A saída, portanto, é composta por uma imagem rotulada a partir do reconhecimento de cada objeto, feito através da atribuição de um mesmo rótulo a pixels que constituem uma mesma região. É esta imagem segmentada que submetida como entrada a processos de análise computadorizada, nos quais os objetos recebem uma interpretação por parte do desenvolvedor.

Utilizando de conceitos da teoria de conjuntos, pode-se definir como  $R$  toda a região espacial de uma imagem, segmentada em subconjuntos  $R_i, i = 1, 2, \dots, n$  tais que:

1.  $\cup R_i = R$ ;
2.  $R_i$  é um conjunto convexo, ou conectado;
3. Para todo  $i \neq j, R_i \cap R_j = \emptyset$ ;
4. Sendo  $Q(R_i)$  uma propriedade lógica definida sobre os pontos de um conjunto  $R_i$ ,  $Q(R_i) = True$  para todo e qualquer conjunto  $R_i$  e  $Q(R_i \cup R_j) = False, i \neq j$ .

A partir da definição, existem vários métodos de segmentação desenvolvidos de modo a satisfazer dada condição  $Q$ , podendo ser eles interativos ou automáticos, a depender da interação do desenvolvedor com o algoritmo, e utilizando diferentes estratégias para a representação de objetos.

Para imagens monocromáticas, a segmentação geralmente é baseada em uma de duas categorias básicas relacionadas às propriedades dos valores de intensidade. A primeira é a descontinuidade, na qual a imagem é dividida de acordo com mudanças bruscas de intensidade, como as bordas. A segunda é a similaridade, na qual divide-se a imagem em regiões que sejam semelhantes de acordo com critérios predefinidos.

### 2.1.3.1 *Watershed*

O *Watershed* é um algoritmo de segmentação de imagens pela divisão em regiões baseado numa analogia com bacias hidrográficas. Interpretando a imagem de modo "topográfico", três tipos de ponto são considerados: pontos pertencentes a um mínimo regional; pontos nos quais gotas d'água, ao serem despejadas neles, cairiam quase que certamente em um mínimo simples; e pontos em que a água tem a mesma probabilidade de cair em mais de um mínimo. Os pontos prováveis a cair em um dado mínimo regional são interpretados como a bacia hidrográfica do mesmo. Pontos com probabilidade de cair em mais de um mínimo são linhas de crista sobre a superfície topográfica.

Estas linhas, chamadas de linhas de divisão ou linhas de *watershed*, são o foco do algoritmo. Imagina-se, de acordo com (GONZALEZ, 2010), um orifício sendo perfurado em cada mínimo regional e a topografia inteira sendo inundada de baixo para cima, deixando a água subir pelos orifícios de modo uniforme. Quando a água acumulada nas diversas *watersheds* está prestes a se juntar, uma barragem é construída para impedir a fusão. A inundação acabará por chegar a uma fase em que apenas os topos das barragens são visíveis acima da linha d'água. Esses limites da barragem, as linhas de *watershed*, são as fronteiras extraídas pelo algoritmo. A Figura 7 ilustra esta analogia.

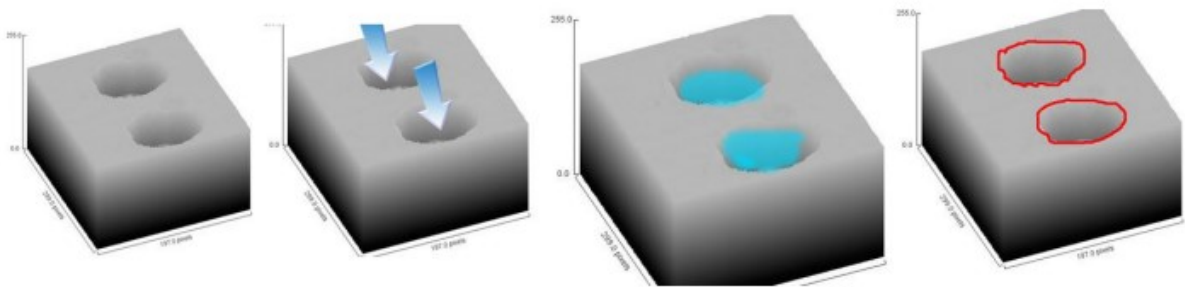


Figura 7 – Segmentação por *watershed*.

Fonte: "Segmentação baseada em Textura e *watershed* aplicada a imagens de Pólen", (ANDRADE, 2011)

Este algoritmo, apesar de eficaz, é, no entanto, demasiadamente lento. Por isso,

alternativas a ele foram propostas desde o seu desenvolvimento e algumas delas serão aplicadas neste trabalho.

### 2.1.3.2 Aprendizado de máquina

Existem técnicas de segmentação de imagem baseadas em região realizadas através de modelos de contornos ativos. São alternativas eficazes, particularmente, na segmentação de imagens ruidosas e/ou com limites mal definidos. Apesar disso, pecam na segmentação de imagens com pouca homogeneidade de intensidade. Do lado oposto, algoritmos de segmentação por aprendizado de máquina são excelentes para o tratamento de heterogeneidades, mas pecam na atribuição de rótulos na presença de ruídos excessivos. A técnica proposta em (PRATONDO, 2016) propõe um *framework* que integra *machine learning* com um modelo de contorno ativo baseado em região. *Scores* de probabilidade de classificação de algoritmos de *machine learning*, aplicados através de funções não lineares, são usados para substituir a intensidade do pixel em valores de menor intensidade. O resultado é melhor precisão e menos sensibilidade a alterações de valores e intensidades.

## 2.2 Imagens de ultrassom

### 2.2.1 Princípios básicos

O som é uma onda mecânica propagada através da vibração das partículas do meio no qual se movimenta de modo longitudinal. Caracteriza-se através de uma velocidade de propagação, comprimento de onda e frequência de vibração, cuja relação é descrita pela Equação 2.9:

$$v = \lambda \times f \quad (2.9)$$

Quando fala-se em ultrassom, refere-se necessariamente à faixa de frequência de onda superior à captada pelo ouvido humano, ou seja,  $f > 20$  KHz. As ondas são em geral longitudinais, podendo também se propagar de modo transversal em corpos sólidos como ossos. Dada sua natureza mecânica, a necessidade de matéria para a propagação faz com que a velocidade esteja diretamente relacionada com a densidade  $\rho$  e o módulo volumétrico de elasticidade  $B$  do meio de propagação através da Equação 2.10:

$$c = \sqrt{\frac{B}{\rho}} \quad (2.10)$$

Tem-se na Tabela 1 a velocidade de propagação do som em alguns materiais e tecidos (PAPALEO, 2019). Observa-se que a velocidade de propagação do som na água é quase 5

vezes superior à velocidade de propagação no ar. Este é o motivo do uso de gel nos exames de ultrassonografia realizados.

Tabela 1 – Velocidade de propagação do som em alguns materiais e tecidos.

Material	c (m/s)
Ar	330
Água (a 20 °C)	1.482
Tecido mole médio	1.540
Sangue	1.580
Gordura	1.450
Fígado	1.550
Rim	1.560
Músculo	1.585
Pulmão	950
Osso (típico)	3.200
Coração (músculo)	1.560
Cristais dos transdutores piezoelétricos	4.000

Fonte: (PAPALEO, 2019)

Na ultrassonografia, utiliza-se de feixes sonoros pulsados de alta frequência (de 1 a 15 MHz) gerados por cristais piezoelétricos excitados por sinais elétricos para obter imagens do corpo humano. Cada vez que o feixe sonoro encontra em seu caminho interfaces acústicas, caracterizadas por alterações na densidade ou elasticidade do meio, uma fração da energia sonora é refletida. A onda refletida retorna ao cristal, sendo detectada através da deformação mecânica do mesmo e convertida em um sinal elétrico relacionado ao tempo de propagação de ida e volta e à distância do cristal ao objeto identificado. Posteriormente, este sinal é convertido em uma escala de tons de cinza, de acordo com a amplitude do sinal demodulado e amplificado. (PAPALEO, 2019)

### 2.2.2 Artefatos

Artefatos podem ser descritos como "defeitos de imagem". Tipo de imagem, tipo de aquisição, tecnologia utilizada, protocolo seguido com a aquisição e erros de usuário são alguns dos diversos motivos que levam uma imagem final a apresentar artefatos, resultando em informações perdidas e/ou mascaradas na imagem obtida. No caso do ultrassom, artefatos são definidos como quaisquer alterações da imagem ecográfica não correspondentes à estrutura examinada. Muitos artefatos na imagem são induzidos por erros na técnica de escaneamento ou pelo uso impróprio do instrumento, mas também podem resultar da

violação das suposições feitas pelo instrumento para a aquisição ideal da imagem. De acordo com (KREMKAU, 1986), as principais suposições são:

1. As ondas sonoras se propagam em linhas retas;
2. Cada interface gera apenas um eco ou reflexão;
3. Ecos são formados apenas por objetos localizados no eixo do transdutor;
4. A intensidade dos ecos retornados é diretamente relacionada à dispersão do objeto;
5. A velocidade do som e a atenuação são homogêneas e conhecidas a priori;
6. Qualquer eco detectado é devido ao pulso transmitido mais recente.

Em via de regra, o compromisso é não permitir que um artefato atrapalhe o processamento da imagem e a identificação das estruturas. Dentre os principais artefatos de imagens de ultrassom, destaca-se aqui lóbulos laterais, sombra acústica e artefatos de reverberação. Apesar do potencial de comprometimento do exame e diagnóstico, eles também podem ser fontes de informação, caso saiba-se explorá-los.

#### 2.2.2.1 Lóbulos Laterais

Para uma modelagem ideal, assume-se que cada cristal piezoelétrico do transdutor mapeia uma linha vertical do espaço e a composição do mapeamento por todos os cristais forma uma imagem bidimensional. No entanto, tem-se em realidade o feixe de ultrassom com propagação tridimensional. Assim, a Suposição 3 é violada e existem ecos fora do eixo do transdutor. Além disso, estruturas muito refletoras são capazes de produzir ecos fortes a ponto de serem detectados pelo transdutor e representados na imagem como uma estrutura inexistente.

Um exemplo de imagem com lóbulos laterais pode ser visto na Figura 8. Nela, tem-se uma agulha flexível imersa em um simulador translúcido de tecido biológico. A agulha é muito refletora e a diferença de velocidade entre ela e o meio faz com que os ecos se potencializem, dando origem às estruturas vistas na Figura, além da ponta da agulha. O ideal, neste contexto, é que se execute uma operação morfológica de abertura para que os lóbulos sejam separados da agulha para fins de segmentação da imagem.

Para fins diagnósticos, de acordo com (LAING, 1982), lóbulos laterais ocorrem perto de superfícies fortes, curvas e muito refletoras, como o diafragma, ou perto de grandes massas císticas, como a bexiga urinária ou a vesícula biliar. Artefatos difusos originam-se de gases intestinais adjacentes a estruturas císticas. Desse modo, percebe-se que saber explorá-los fornece informações médicas de grande valia.

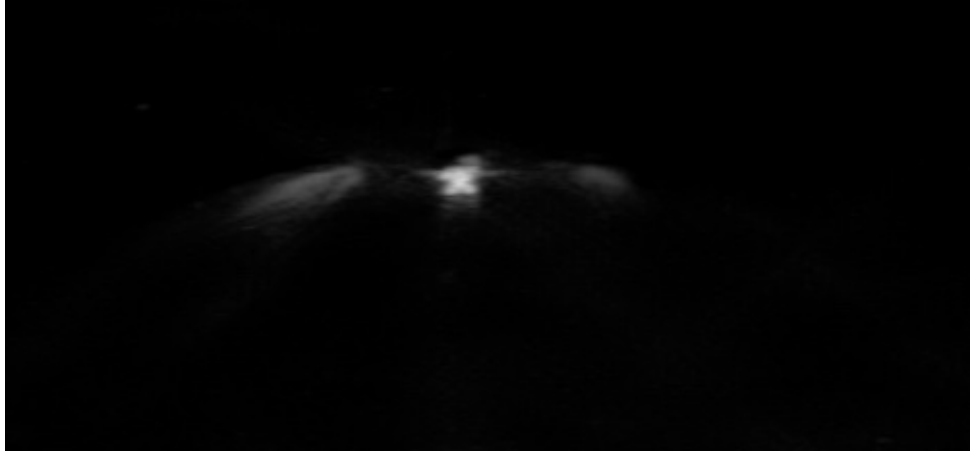


Figura 8 – Representação ecográfica de uma agulha flexível imersa em simulador translúcido de tecido biológico, contendo lóbulos laterais como artefatos.

Fonte: (KIYOKU, 2015)

#### 2.2.2.2 Sombra acústica

A sombra acústica é causada pela mudança do meio de propagação da onda sonora. Na fronteira entre dois meios, parte da energia da onda é refletida e parte é transmitida. A razão entre a reflexão e a transmissão é determinada pela diferença de impedância entre os meios e diretamente proporcional a quantidade de energia refletida. Objetos muito refletivos como os metálicos são geralmente representados como estruturas muito claras e brilhantes. Depois deles, a energia transmitida é muito pequena, e com isso podem ocorrer sombreamentos na imagem logo após a representação dessas estruturas. Um exemplo de sombra acústica pode ser visto na Figura 9. Em um meio fibroso, a agulha é representada de modo brilhante e, abaixo dela, dada a alta diferença de impedância entre o metal e a fibra, ocorre um sombreamento.

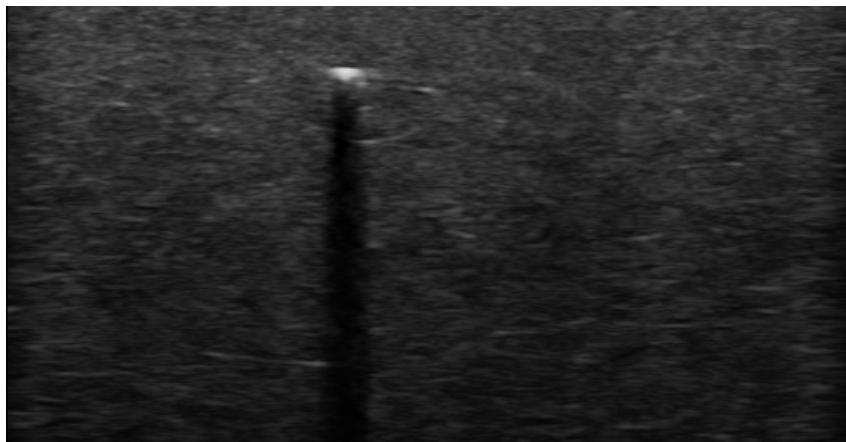


Figura 9 – Representação ecográfica de uma agulha flexível imersa em simulador fibroso de tecido biológico, contendo uma sombra acústica como artefato.

Fonte: (KIYOKU, 2015)

Na medicina, a sombra acústica pode ser explorada por ser característica de lesões

que apresentam uma forte reflexão, permitindo, dessa forma, esclarecer dúvidas que venham a ocorrer. (LIMA, 2013)

### 2.2.2.3 Reverberação

A reverberação se dá por meio da violação da Suposição 2, que infere que cada interface gera apenas um eco, e ocorre dada a grande impedância acústica dos meios que constituem as interfaces refletoras. A questão, aqui, é a reprodução de falsos ecos. A onda é refletida por repetidas vezes entre duas superfícies paralelas e altamente reflexivas antes de retornar ao transdutor. Este, por sua vez, interpreta as duas primeiras reflexões como os objetos que são e as demais acabam por gerar um artefato de imagem.

Uma forma de reverberação chamada de *Cauda de Cometa* (CTA) pode ser vista na Figura 10. Tem-se nela uma agulha posicionada de modo perpendicular ao plano imersa em banho d'água e, devido às dimensões diminutas da agulha e à alta velocidade de propagação do som no metal, consta na imagem uma linha vertical formada pela reverberação da onda na agulha, que forma artefatos com uma distância muito pequena entre eles. Observa-se na imagem também a presença de lóbulos laterais, já explicados anteriormente.



Figura 10 – Representação ecográfica de uma agulha flexível imersa em banho d'água, contendo uma cauda de cometa formada por artefatos de reverberação.

Fonte: (KIYOKU, 2015)

## 2.3 Filtro de Kalman

De acordo com (AGUIRRE, 2007) e (BISHOP, 2006), entende-se por filtro de Kalman o conjunto de equações recursivas utilizadas para estimar e corrigir os estados de um sistema dinâmico proposto por Rudolf Kalman em 1960. Desde então, com o avanço da tecnologia digital, o filtro de Kalman tem sido objeto de extensas pesquisa e aplicação. Sua aplicabilidade se estende ao processamento de imagens, sendo ele bastante utilizado para a melhoria da



estimação da posição de objetos e reconstrução de imagens médicas, como aplicado, por exemplo, em (MELLO, E. B. de, 1998).

### 2.3.1 Princípios básicos

O filtro de Kalman tenta estimar o estado de um processo controlado em tempo discreto governado por uma Equação de diferença estocástica linear. Pode-se representar o Filtro de Kalman, então, através do seguinte modelo linear discreto:

$$\begin{cases} x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ z_k = Hx_k + v_k \end{cases} \quad (2.11)$$

No modelo 2.11, tem-se em  $x_k$  o próximo passo temporal no processo, em  $x_{k-1}$  o passo atual,  $u_k$  uma entrada de controle opcional e as matrizes  $A$ , de dimensões  $n \times n$ , e  $B$ , de dimensões  $n \times 1$ , são as chamadas matrizes de transição e de controle, respectivamente, responsáveis por relacionar o estado e a entrada de controle atuais com as do passo seguinte. Em  $z_k$  tem-se a medição no passo  $k$  e a matriz  $H$ , de dimensões  $m \times n$  e chamada de matriz de medição, relaciona o estado com a medição  $z_k$ .

$w$  e  $v$  são variáveis aleatórias independentes, de média nula e representam, respectivamente, o ruído de processo, ou multiplicativo, e medição, ou aditivo. São consideradas independentes entre si, ruídos brancos e com distribuições de probabilidade normal. Sendo  $Q$  e  $R$  suas respectivas matrizes de covariância, tem-se:

$$\begin{cases} w \sim N(0, Q) \\ v \sim N(0, R) \end{cases} \quad (2.12)$$

### 2.3.2 Filtro de Kalman Discreto

O filtro de Kalman Discreto estima um processo usando um controle com *feedback*: o filtro estima o estado do processo em dado momento e, em seguida, obtém *feedback* na forma de medições que, por sua vez, são ruidosas. Pode-se, portanto, dividir as equações do filtro de Kalman em dois grupos:

1. Equações de atualização de tempo;
2. Equações de atualização de medida.

As primeiras são responsáveis por projetar no tempo estimativas de estado atual e covariância no de erro, visando estimar, também, o próximo passo de tempo. Já as segundas são responsáveis pelo *feedback*, ou seja, por incorporar uma nova medição na estimativa do estado atual visando uma estimativa a posteriori melhorada.

Diante deste cenário, pode-se pensar nas primeiras equações como equações de previsão e nas segundas como equações de correção ou atualização. Estabelece-se, então, um algoritmo preditor-corretor para a resolução de problemas numéricos.

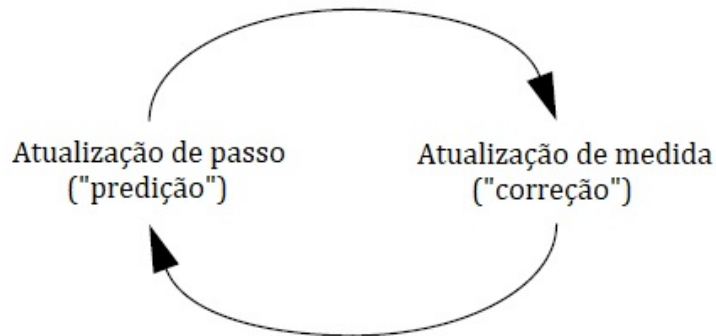


Figura 11 – Ciclo de filtro de Kalman discreto em andamento.

Fonte: (BISHOP, 2006)

### 2.3.2.1 Predição

Nesta etapa, o filtro gera uma estimativa  $\hat{x}_k^-$  baseada na estimativa do passo atual  $\hat{x}_{k-1}^-$  e nos ajustes feitos por meio da entrada opcional  $\hat{u}_{k-1}$ . A atualização do tempo projeta a estimativa do estado atual com antecedência e a atualização de medição ajusta a estimativa projetada por uma medição real naquele momento, gerando o ciclo presente na Figura 11. Tem-se, a partir daí, as equações de atualização temporal, representadas pelo modelo 2.13 abaixo:

$$\begin{cases} \hat{x}_k^- = \hat{A}x_{k-1}^- + Bu_{k-1} \\ P_k^- = AP_{k-1}A^T + Q \end{cases} \quad (2.13)$$

A covariância do erro de estimativa do passo anterior  $P_k^-$  é calculada a partir da matriz de transição  $A$  e da matriz de covariância do ruído do processo  $Q$ . Esta medida indica o grau de incerteza da estimativa em relação ao processo.

### 2.3.2.2 Correção

A etapa de correção ocorre com a chegada da medição na iteração  $k$ . Com essa nova informação, é possível corrigir o vetor de estados previamente à chegada da medição  $x_k^-$ , para dar origem ao vetor de estados corrigido e definitivo  $x_k^+$ . Isso é feito através da modelagem matemática 2.14 a seguir:

$$\begin{cases} K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \\ x_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k = (I - K_k H)P_k^- \end{cases} \quad (2.14)$$

---

O primeiro cálculo a se fazer durante a atualização da medição é calcular o ganho de Kalman  $K_k$ . Em seguida, é realmente medir o processo a fim de obter  $z_k$  e, então, gerar uma estimativa de estado a posteriori incorporando a medição. O passo final é obter uma estimativa de covariância de erro a posteriori.

A matriz de ganho de Kalman  $K_k$  indica a confiabilidade a ser dada a cada medição ruidosa  $z_k$  e o quão rápido as estimativas irão convergir para a medida a cada passo. O ganho é definido majoritariamente a partir de  $Q$  e  $R$ , sendo que este último, por sua vez, indica o quão confiáveis são as medidas obtidas por *feedback* para o filtro. Para  $R$  elevado, tem-se medidas pouco confiáveis e pouca influência das medições sobre as estimativas. Já para baixos valores de  $R$ , tem-se alta confiabilidade e estimativas que convergem às estimações rapidamente.

Deve-se considerar também que um grau de incerteza de medição muito elevado pode tornar o filtro demasiadamente lento, mas um valor muito baixo pode fazer do filtro instável e gerador de medidas extrapoladas. Se as matrizes  $Q$  e  $R$  forem constantes, as covariâncias de erro e ganho tendem a se manterem constantes depois de algumas iterações. Esses parâmetros podem ser determinados antes da implementação do filtro, com informações oriundas de estudos do problema a ser resolvido e testes.

## 3 Desenvolvimento

Neste capítulo, é feita a descrição do desenvolvimento do projeto. O primeiro passo foi a familiarização com o sistema de imageamento por ultrassom, seu funcionamento e com o método de aquisição das imagens da movimentação da agulha. Posteriormente, considerando que o objetivo do projeto é a sua otimização, desenvolvendo técnicas que o tornem mais rápido e compatível com uma aplicação em tempo de execução, foi feita a análise das técnicas escolhidas para o pré-processamento, segmentação e estimação da posição da ponta da agulha, bem como a aplicação do Filtro de Kalman ao fim da estimação. A partir desta análise, foram desenvolvidas melhorias e alternativas ao que se julgou ser incompatível com a aplicação e seus objetivos. Por fim, calculou-se o custo computacional do projeto, de modo a validá-lo como compatível com uma aplicação em tempo de execução.

### 3.1 Sistema de Imageamento Por Ultrassom

O sistema de imageamento por ultrassom aqui utilizado foi desenvolvido em (KIYOKU, 2015) é composto por um equipamento de ultrassom, responsável pela aquisição em tempo real das imagens, conectado via *Ethernet* a um computador periférico, no qual ocorre a o processamento das mesmas. Através deste sistema, obtém-se imagens de uma intervenção com guiagem de uma agulha flexível em simuladores de tecido biológico mole denominados *phantoms*.

#### 3.1.1 Equipamento de ultrassom

Conta-se, nesse projeto, com um equipamento de ultrassom SonixTOUCH, desenvolvido pela empresa *Ultrasonix* (ULTRASONIX, 2013a). Tem-se, nele, o sistema operacional *Windows XP* embarcado e uma *Interface de Pesquisa de Ultrassom* ou, do inglês, uma *URI*. Aliadas, elas geram grandes possibilidades por permitir o trabalho em modo de pesquisa, dando ao desenvolvedor acesso a informações que um sistema puramente clínico não daria. Assim, tem-se mais liberdade de trabalho sobre as imagens adquiridas, o armazenamento de dados em diversos formatos. Uma grande vantagem deste equipamento é a possibilidade de conexão em rede local com outros sistemas para aquisição de dados.

Para a aquisição de imagem, conta-se com dois possíveis transdutores: um convexo, modelo C5-2, e um linear, modelo L14-5W. O equipamento de ultrassom e os transdutores podem ser vistos nas Figuras 12 e 13.

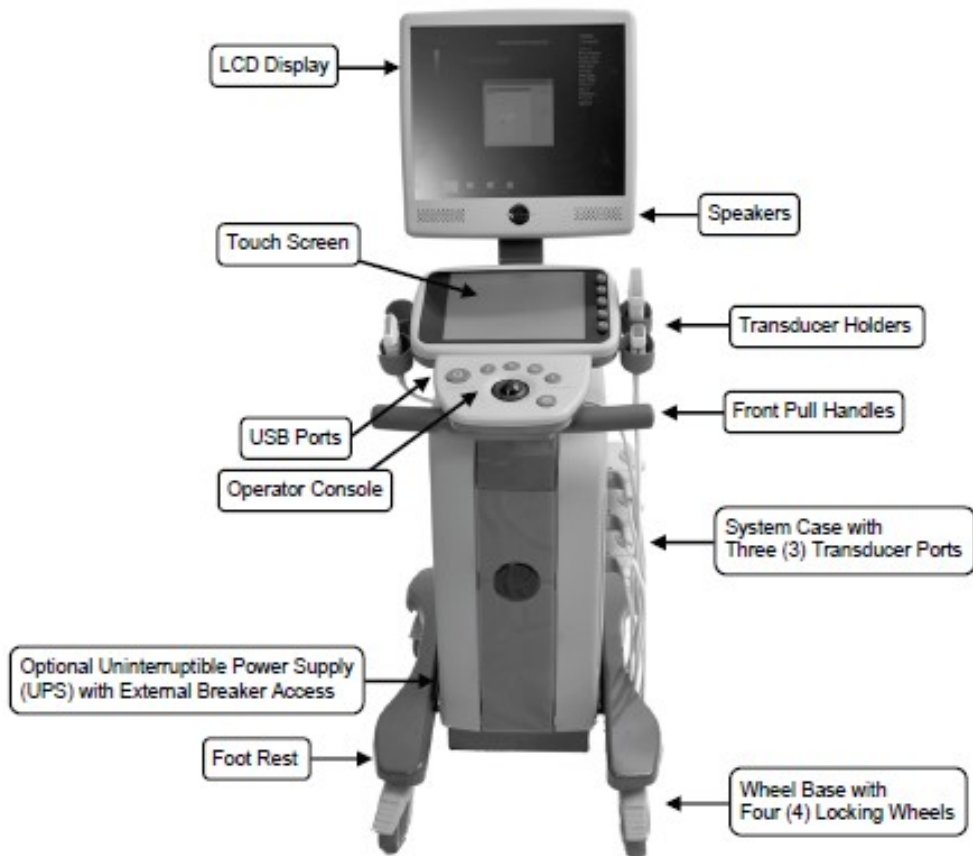


Figura 12 – Equipamento de ultrassom SonixTOUCH Research.

Fonte: (ULTRASONIX, 2013a)



(a) Transdutor convexo.



(b) Transdutor linear.

Figura 13 – Transdutores

Fonte: (ULTRASONIX, 2013b)

A partir deste equipamento, foram analisadas as diversas possibilidades de implemen-

tação para a aquisição de imagens. A *Ultrasonix* fornece uma série de *Kits de Desenvolvimento de Software*, ou, do inglês SDKs. Um deles, o *SDK Ulterius*, traz consigo a possibilidade de estabelecer uma conexão TCP/IP com a URI, podendo o programa cliente ser executado no próprio equipamento ou em um computador periférico. Além disso, a *Ultrasonix* também fornece um conjunto de ferramentas para o MATLAB chamadas de *SonixDataTools*, que permite uma boa visualização dos dados. No entanto, estes SDKs pecam em portabilidade, por estarem disponíveis apenas para o sistema operacional *Windows*.

Diante disso, o trabalho desenvolvido em (KIYOKU, 2015) não prosseguiu com o uso dos SDKs próprios da *Ultrasonix*. Priorizou-se a modularidade e portabilidade, que facilita futuras modificações e alterações, bem como a interoperabilidade em um sistema clínico real, no qual diversos equipamentos, cada um com suas configurações e parâmetros, podem vir a operar juntos. Diante disso, buscou-se alternativas aos SDKs, encontradas em protocolos de código aberto.

### 3.1.2 Sistema de aquisição de imagens

Feitas as considerações descritas na seção anterior e visando uma comunicação em rede portátil, dada a grande variedade de sistemas operacionais e arquiteturas de CPU e a possibilidade de conexão com diferentes ambientes de programação, optou-se por desenvolver uma solução para conexão TCP/IP utilizando o protocolo *OpenIGTLink*.

De acordo com (TOKUDA et al., 2009), tem-se no *OpenIGTLink* de um protocolo projetado para uso na camada de aplicação da pilha do TCP/IP, enquanto permite aos desenvolvedores implementá-lo para outros modelos de rede, como o *Data User Protocol*. O *OpenIGTLink* não inclui mecanismos para estabelecer e gerenciar um sessão. A mensagem, unidade mínima de dados do protocolo, contém todas as informações necessárias para interpretação pelo receptor. Começa com um cabeçalho de 58 bytes, comum a todos os tipos de dados e organizado de acordo com a Tabela 2. Este cabeçalho é seguido pelo corpo da mensagem, cujo formato varia de acordo com o tipo de dado especificado no cabeçalho. Uma vez que qualquer receptor compatível pode interpretar o cabeçalho, que contém o tamanho e o tipo de dados do corpo, cada receptor pode lidar com qualquer mensagem, mesmo aquelas com tipo de dados desconhecido. Portanto, essa estrutura de duas seções permite que os desenvolvedores definam seu próprio tipo de dados.

Para o envio de imagens, o tipo de dado *'IMAGE'* é especificado no cabeçalho e, com ele, há suporte para a transmissão de imagens 2D e 3D. Tem-se, então, o cabeçalho da imagem, com informações como o tamanho da imagem, tipo de sistema de coordenadas, posição e orientação; e o corpo da imagem, que contém sua matriz de pixels (TOKUDA et al., 2009).

Neste ponto, então, tem-se um protocolo para a transmissão de imagens. Para prepa-

Tabela 2 – Cabeçalho de uma mensagem transmitida por meio do protocolo OpenIGTLink.

Informação do cabeçalho	Finalidade	Tamanho
Número da versão	Número da versão do protocolo OpenIGTLink	2 bytes
Tipo de dado	Nome do tipo de dado a ser transmitido	12 bytes
Nome do dispositivo	Nome do dispositivo de origem	20 bytes
Selo de Tempo	Momento no qual o dado foi gerado	8 bytes
Tamanho do corpo	Tamanho da seção de corpo da mensagem	8 bytes
CRC	Verificação de redundância cíclica para a seção do corpo	8 bytes

Fonte: Produzido pelo autor.

rar estas imagens para a transmissão, foi adotada a biblioteca *PLUS (Public software Library for Ultrasound)*. Descrita em (LASSO et al., 2014), trata-se de uma biblioteca gratuita e de código aberto para a prototipagem de sistemas de procedimentos guiados por imagens de ultrassom. Traz consigo, também, uma aplicação denominada *PlusServer*, com a capacidade de aquisição de dados de dispositivos e envio para os clientes conectados. Desse modo, o sistema de aquisição de imagens é composto pela implementação do *PlusServer* no equipamento de ultrassom *SonixTOUCH*, sendo as imagens adquiridas com os transdutores já descritos transmitidas via Ethernet, em tempo real, a um computador periférico com uma aplicação cliente compatível com o *OpenIGTLink*. Pode-se ver um esquemático do sistema de imageamento via ultrassom na Figura 14.

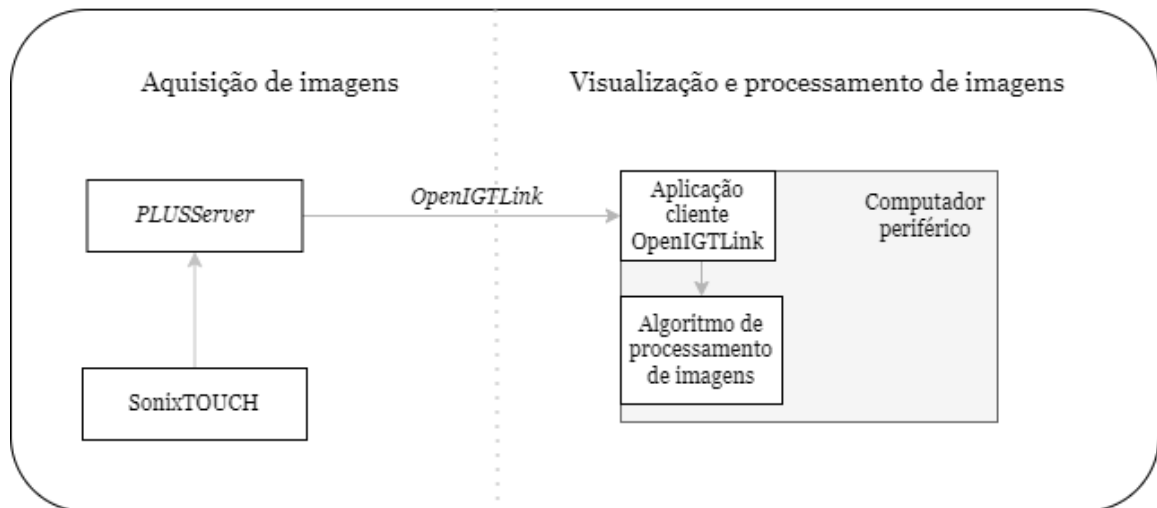


Figura 14 – Esquemático do sistema de imageamento via ultrassom.

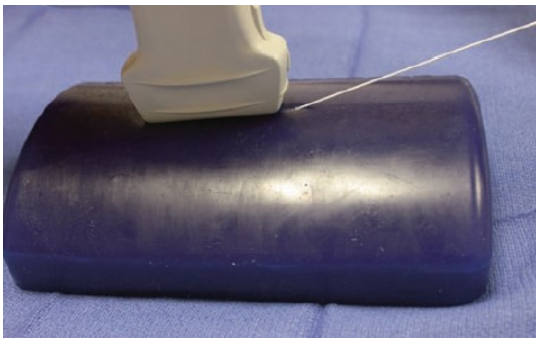
Fonte: Produzido pelo autor.

### 3.1.3 Phantoms

*Phantoms* são simuladores de tecido biológico desenvolvidos com a finalidade de treinar operadores manuais e pesquisar novas técnicas. Podem imitar propriedades acústicas

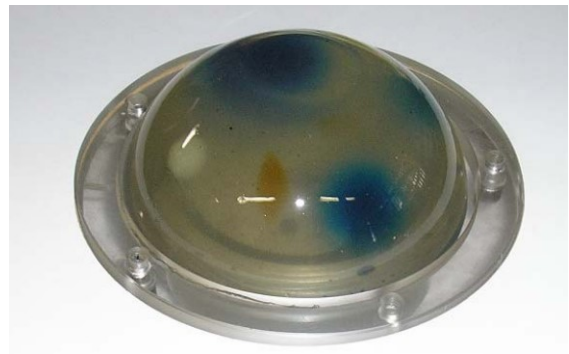
do tecido como a velocidade de propagação do som ou atenuação e são capazes, também, de reproduzir a aparência do tecido nas imagens adquiridas. São fundamentais no ambiente de pesquisa e estudos, dado que inconveniências e desconforto ao paciente podem ser reduzidos se o profissional realizar treinamentos e simulações eficazes usando um ambiente o mais realista possível.

Esse tipo de simulador pode ser classificado de acordo com a fidelidade de reprodução da textura do tecido. Exemplos de *phantoms* de alta fidelidade são o *blue phantom* (NAROUZE, 2011), um dos mais realistas e duráveis simuladores de tecido biológico disponíveis, e o *phantom* desenvolvido em (VIEIRA, 2005), com geometria equivalente a da mama e inclusões representando nódulos, voltado ao treinamento de biópsias de mama guiadas por ultrassom. Ambos podem ser vistos na Figura 15.



(a) Blue phantom.

Fonte: (NAROUZE, 2011)



(b) Phantom de mama.

Fonte: (VIEIRA, 2005)

Figura 15 – Exemplos de simuladores de tecido biológico de alta fidelidade.

O *data base* deste projeto conta com imagens adquiridas através de ensaios da movimentação de uma agulha flexível utilizando três tipos de simulação de tecido biológico de baixa fidelidade, porém em níveis de fidelidade diferentes, presentes na Figura 16. Existem duas abordagens possíveis para o alinhamento entre o transdutor e o plano de inserção da agulha. Na primeira delas, *fora do plano*, o plano de inserção da agulha é perpendicular ao plano da imagem, fazendo com que a agulha apareça na imagem como um ponto brilhante. Na segunda delas, a abordagem *no plano*, o plano de inserção da agulha é alinhado ao plano da imagem, de modo que a agulha aparece na imagem como uma linha. A primeira abordagem foi a escolhida.



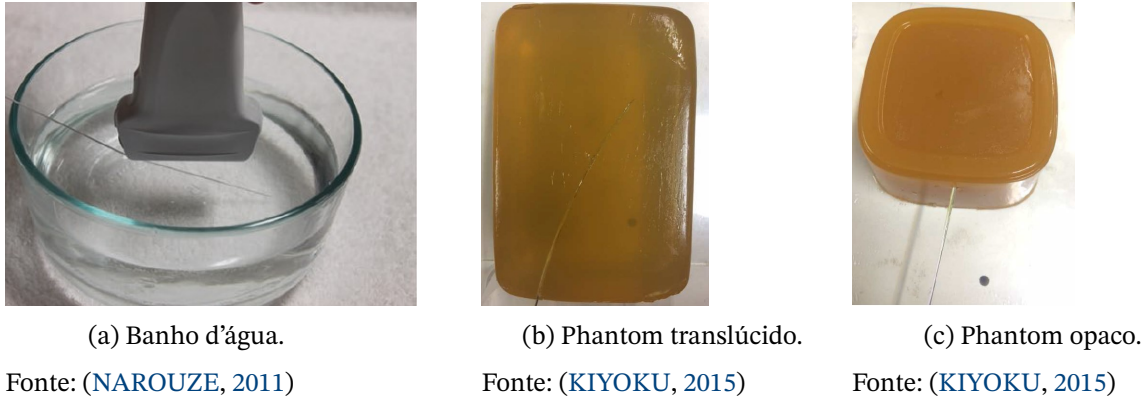


Figura 16 – *Phantoms* de banho d'água, gelatina translúcida e gelatina opaca.

O primeiro *phantom* utilizado foi um banho d'água, que consiste simplesmente na movimentação da agulha em um recipiente com água. A vantagem deste simulador é a simplicidade de confecção do mesmo e também a visibilidade da movimentação da agulha com a qual o desenvolvedor pode contar. No entanto, as propriedades acústicas e morfológicas não se assemelham às de tecidos moles.

As demais imagens que compõem o *data base* do projeto foram adquiridas por meio da movimentação da agulha em dois simuladores de gelatina diferentes, sendo o primeiro composto por uma gelatina translúcida e sem partículas suspensas e o segundo feito com fibras de *Psyllium*, que visam reproduzir o ruído causado pela dispersão de tecidos moles. No primeiro, ainda há a translucidez que gera facilidade ao desenvolvedor em acompanhar a movimentação da agulha, apesar da pouca fidelidade a tecidos biológicos. No segundo, a visibilidade é perdida, mas ganha-se em fidelidade. Imagens de ultrassom obtidas através de ensaios com os *phantoms* de banho d'água, translúcido e fibroso podem ser vistas, respectivamente, nas Figuras 10, 8 e 9, presentes no Capítulo 2. Observam-se variações de ruído, contraste e artefatos presentes nas imagens adquiridas de um *phantom* para outro, sendo a gelatina opaca a mais fiel ao tecido.

A partir deste ponto, a Figura 17 traz o resumo das etapas às quais a imagem é submetida até a estimação da ponta da agulha. As próximas seções explorarão cada etapa do processamento, destacando o que foi mantido e, principalmente, as propostas de alteração feitas. Importante destacar que, dado que as imagens resultantes da movimentação em cada tipo de phantom têm suas peculiaridades em relação às demais, algumas operações variam de acordo com o tipo de phantom correspondente à imagem. A Tabela 3 mostra quais operações são realizadas de acordo com cada tipo de imagem.

No presente trabalho, o processamento de imagens adquiridas por meio de *phantoms* de banho d'água não é considerado. Essa escolha foi feita dada a baixíssima similaridade deste meio com tecidos moles, servindo ele apenas ao estudo de técnicas de processamento. A gelatina translúcida ainda é considerada, mas o foco é direcionado ao *phantom* opaco, dado que as fibras de *Psyllium* fazem com que a dispersão do meio seja mais similar a de

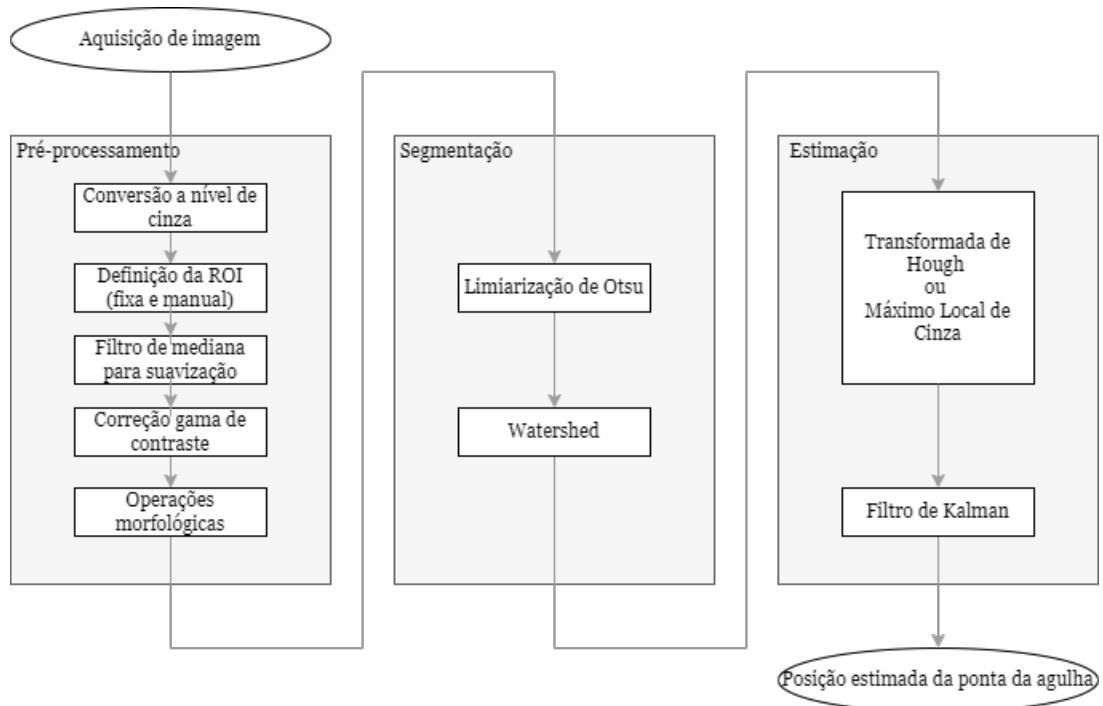


Figura 17 – Resumo do processo de estimação da ponta da agulha.

Fonte: Produzido pelo autor.

Tabela 3 – Operações executadas com as imagens adquiridas a partir de cada phantom.

Etapa do processamento	Operação		
	Banho d'água	Gelatina Translúcida	Gelatina Opaca
Filtro de suavização	Filtro de mediana	Filtro de mediana	Filtro de mediana
Correção de contraste	Correção gama	Correção gama	Correção gama
Operações morfológicas	Abertura	Abertura	Gradiente
Operações morfológicas	-	-	Fechamento
Limiarização	Limiarização de Otsu	Limiarização de Otsu	Limiarização de Otsu
Segmentação	<i>Watershed</i>	<i>Watershed</i>	<i>Watershed</i>
Estimação	Transformada de Hough	Transformada de Hough	Transformada de Hough
Estimação	Máximo local de cinza	Máximo local de cinza	Máximo local de cinza

Fonte: Produzida pelo autor.

tecidos biológicos que a dos demais *phantoms*.

## 3.2 Pré-processamento

Antes do pré-processamento, a primeira etapa é a conversão da imagem em escala de cinza, visto que, se tratando de imagens de ultrassom, já em preto e branco, este procedimento não gera nenhuma perda ao processo, apenas economia no custo computacional ao longo do processamento, já que se passa a trabalhar com apenas um canal de pixels, ao invés de

três canais. A partir deste ponto o pré processamento da imagem é, então, realizado, visando a suavização de ruídos e a máxima preservação das bordas. Importante ressaltar que todas as operações de imagem são feitas em *Python* e utilizando *OpenCV*, a principal biblioteca de código aberto para a visão computacional, processamento de imagem e aprendizagem de máquina, e podem ser encontradas no repositório do projeto<sup>1</sup>.

### 3.2.1 Definição da ROI

A primeira observação feita sobre o processamento é a de que a ROI era definida manualmente. O desenvolvedor ou usuário era o responsável por fornecer manualmente as coordenadas de seu centro e, com isso, uma região de  $200 \times 200$  pixels era definida e se mantinha fixa ao longo de todo o processamento, independentemente da movimentação da agulha. Nesta configuração, era possível chegar a um momento no qual a agulha sequer estaria localizada dentro da região de interesse.

Diante dessas circunstâncias, algumas modificações foram propostas. Primeiramente, visando eliminar a interferência do usuário, a primeira opção foi definir o centro da ROI a partir do ponto de maior entropia da imagem, tendo em vista que o máximo local de cinza da imagem concentra-se na região da agulha e, conseqüentemente, a entropia se torna maior neste local. No entanto, observou-se nas imagens em *phantom* opaco que o alto teor de ruído impede esse processo, dada a elevada entropia da imagem como um todo.

A segunda alternativa seria selecionar a ROI a partir do histograma da imagem, dado que, novamente, os pixels com maior valor de cinza estão na área da agulha. É uma boa alternativa mas, a priori, desnecessária. Optou-se por uma alternativa simples. A partir da aquisição da imagem e, com ela, de suas dimensões, o algoritmo centraliza a primeira região de interesse no centro da imagem. Assume-se que, dado que o objetivo do usuário é acompanhar a movimentação da agulha com o transdutor, ela sempre estará localizada próxima ao centro da imagem. A primeira estimativa da posição da ponta da agulha é feita e, a partir de então, o centro da ROI seguinte é localizado na posição da estimativa anterior. Desse modo, a movimentação da região de interesse acompanha a movimentação da agulha.

Visando a redução de custo computacional, testes também foram realizados com ROIs menores, de tamanhos  $150 \times 150$  e  $100 \times 100$  pixels, para fins de comparação de eficácia na localização da agulha com economia do custo computacional.

### 3.2.2 Filtros de suavização

Imagens de ultrassom contam, em sua vasta maioria, com um ruído do tipo speckle. De acordo com (ROSA, 2012), trata-se de um ruído característico de imagens que utilizam fontes coerentes, tais como sistemas ópticos ou acústicos (sonares e ultrassom), dentre outros.

<sup>1</sup> Repositório presente no link <https://github.com/lara-unb/NeedleSteering/tree/master/ustracking>

O ruído speckle é o resultado do espalhamento difuso de ondas em superfícies naturais, que chegam fora de fase ao sensor. Pode ser visto, em especial, em imagens obtidas por meio da movimentação da agulha no simulador opaco como, por exemplo, a Figura 18.

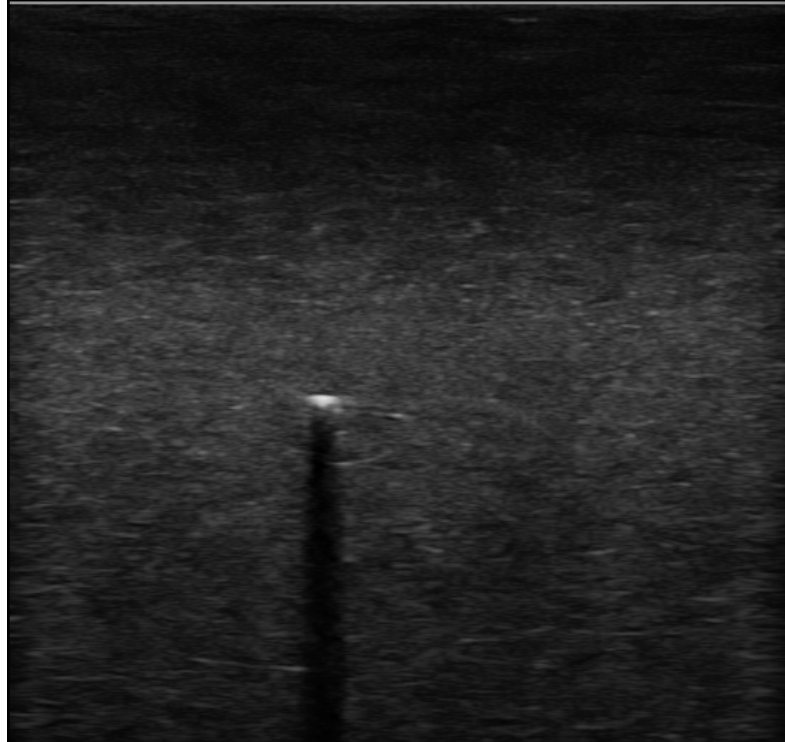


Figura 18 – Imagem de ultrassom representativa da posição de uma agulha flexível em phantom opaco, com forte presença de ruído do tipo *speckle*.

Fonte: (KIYOKU, 2015)

A suavização deste ruído é feita através da implementação de um filtro de mediana. Além dele, analisou-se a possibilidade da implementação de um filtro de média simples, porém, como já descrito no Capítulo 2, o filtro de mediana consegue a mesma suavização de ruído que um filtro de média com maior preservação das bordas, o que é importante para este trabalho. Outra implementação possível é a implementação de um filtro bilateral, também proposto em (ROSA, 2012), dentre outros. Escolheu-se, aqui, o filtro bilateral para análise de sua suavização e custo computacional em relação ao filtro de mediana.

### 3.2.3 Correções de contraste

Como proposto em (KIYOKU, 2015), após a suavização dos ruídos e consequente suavização das bordas, uma transformação de potência foi implementada para que se recuperasse o brilho e o contraste da imagem. Isto favorece a localização da agulha, representada como um ponto brilhante dada a escolha da abordagem *fora do plano* para a aquisição de imagens. A correção gama foi mantida e analisou-se, apenas, qual seria o melhor valor para  $\gamma$ , alterando esta variável no intervalo  $1.3 < \gamma < 1.7$ .

### 3.2.4 Operações morfológicas

As operações morfológicas aqui executadas tem como maior objetivo o tratamento da imagem de modo que eventuais descontinuidades, pequenas ligações e os artefatos presentes não interfiram negativamente na segmentação.

Na imagem adquirida em meio translúcido, que pode ser vista na Figura 19, percebe-se a presença de lóbulos laterais, que são separados de modo bem sucedido através de uma operação de abertura.

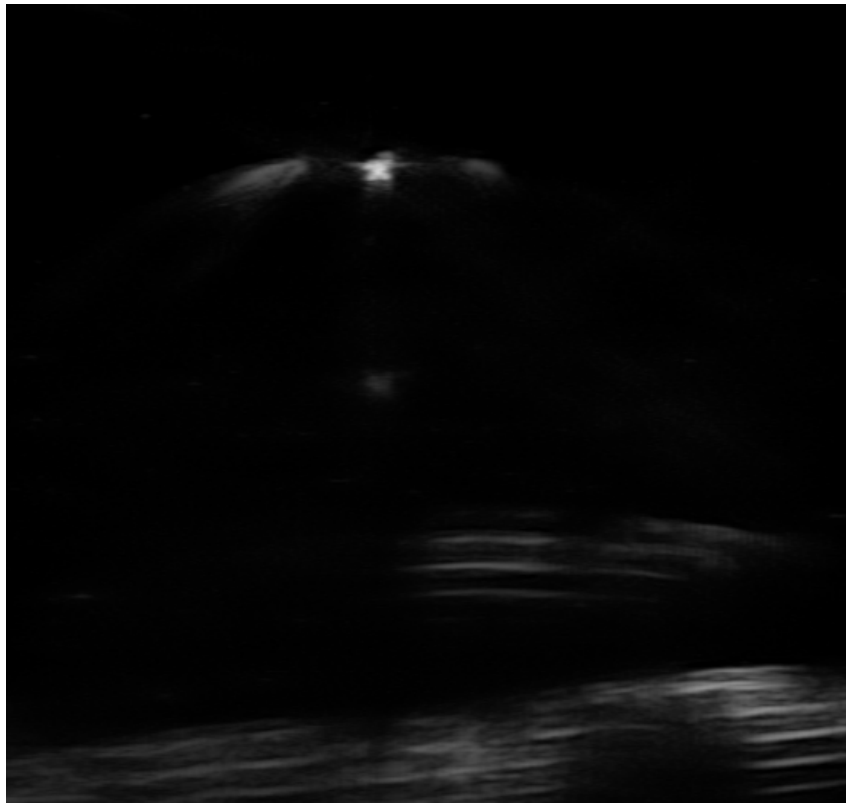


Figura 19 – Imagem de ultrassom representativa da posição de uma agulha flexível em phantom translúcido.

Fonte: (KIYOKU, 2015)

Já na imagem adquirida em meio opaco (Figura 18), a sombra acústica não gera grandes prejuízos à segmentação e a ausência de lóbulos laterais torna a abertura desnecessária. Desse modo, opta-se pela aplicação de um filtro espacial de realce, o gradiente, que ressalta os contornos da agulha. Por fim, para que descontinuidades não prejudiquem a segmentação, é realizada uma operação de fechamento. No presente trabalho, avalia-se a real necessidade desta operação de fechamento. A partir de então, as imagens ficam prontas para a etapa de segmentação.

## 3.3 Segmentação

A segmentação é o passo seguinte da estimação da posição da agulha. Nela, após o preparo da imagem realizado na segmentação, a imagem é dividida em regiões. Primeiramente, utiliza-se da limiarização de Otsu de modo a facilitar a segmentação. Dentre as diversas possibilidades de limiarização, ela foi considerada a melhor alternativa por não depender da subjetividade do usuário para a definição de um limiar ótimo para o agrupamento de objetos. Em seguida, descreve-se aqui as técnicas de segmentação utilizadas no presente trabalho.

### 3.3.1 Algoritmo de segmentação

Após a limiarização da imagem, a segmentação é feita por meio do algoritmo de *Watershed*. Será comprovado no Capítulo 4 que este algoritmo é demasiadamente lento, apesar de eficaz na estimação da ponta da agulha. Por conta disso, algumas alternativas são propostas e serão comparadas no que diz respeito ao custo computacional e à precisão que acarretam à estimativa da posição da agulha. A seguir, serão apresentados o algoritmo de *Watershed* tradicional e mais dois algoritmos baseados nele, mas com a proposta de serem mais eficientes em termos de custo computacional.

#### 3.3.1.1 *Watershed* Tradicional

De acordo com (GONZALEZ, 2010), o algoritmo de *Watershed* tradicional determina que qualquer imagem em tons de cinza pode ser vista como uma superfície topográfica na qual alta intensidade denota picos e colinas, ou máximos locais, e baixa intensidade denota vales, ou mínimos locais. A cada vale é atribuído um rótulo diferente. À medida que "a água sobe", a água de diferentes vales tende a se fundir. Para que isso não aconteça, barreiras são construídas entre eles, sendo as mesmas o resultado da segmentação.

Dado que toda a implementação feita neste trabalho é feita com o uso do OpenCV, segue-se, aqui, a sua abordagem, na qual marcadores são especificados para determinar quais pontos de vale devem ser mesclados e quais não devem. Isso é estabelecido por meio de uma segmentação interativa de imagens. Primeiramente, recebem um rótulo de intensidade regiões que estão certamente no primeiro plano. Depois, recebem um segundo rótulo regiões que se tem certeza que estão no plano de fundo e, por fim, rotula-se regiões de incerteza com o valor zero. Com esse cenário estabelecido, aplica-se o comando de *watershed* para os "divisores de águas". Estes assumirão o valor -1 e, assim, a segmentação de objetos está concluída. Isto pode ser visto no Algoritmo 1.

---

**Algorithm 1** Algoritmo de *Watershed* tradicional.
 

---

- 1: Definição do kernel = mascara 3x3 branca
  - 2: *opening* = abertura(imagem, kernel) ▷ imagem = roi pós limiarização
  - 3: *sure\_bg* = dilatacao(*opening*,kernel) ▷ definição do plano de fundo
  - 4: aplicação de transformada de distância na imagem *sure\_bg*
  - 5: *sure\_fg* = aplicação de threshold na transformada de distancia de *sure\_bg*
  - 6: *unknown* = *sure\_bg* – *sure\_fg* ▷ região de incerteza
  - 7: *n, markers* = cv2.connectedComponents(*sure\_fg*) ▷ n = número de regiões
  - 8: *markers* = *markers* + 1 ▷ *sure\_bg* = 1
  - 9: *markers*[*unknown*==255] = 0 ▷ *unknown* = 0
  - 10: *markers* = int(*markers*)
  - 11: *rgb\_roi* = conversão da ROI de cinza a RGB
  - 12: *markers* = cv2.watershed(*rgb\_roi*,*markers*) ▷ aplicação de watershed para detecção de bordas utilizando OpenCV
  - 13: *rgb\_roi2*[*markers* == -1] = [255,0,0]
  - 14: *original*[*x2*:*x2*+200, *y2*:*y2*+200] = *rgb\_roi2* ▷ aplicação de bordas à imagem original
- 

### 3.3.1.2 *Watershed* baseado em componentes conectados

Em (A BIENIEK, 1999), é proposto um algoritmo de bacia hidrográfica baseado em componentes conectados. Diferentemente do tradicional, este algoritmo é implementado tendo como analogia um processo de chuva, não uma inundação. Assume-se que, ao cair de uma gota de chuva na superfície da imagem, ela deve fluir ao longo da encosta até atingir o vale correspondente. Tendo isso como base, entende-se que o caminho feito pela gota de chuva é um componente conectado e, também, o caminho mais íngreme entre o ponto de queda e o vale. Logo, todos os componentes conectados que levam ao mesmo vale formam uma bacia de captação associada a ele. Este algoritmo pode rotular todas as bacias de captação percorrendo a imagem inteira quatro vezes. Sendo o *Watershed* tradicional um processo interativo, esta alternativa é mais rápida e também eficaz para a segmentação da imagem. O esquemático do algoritmo pode ser visto abaixo, no Algoritmo 2. Para isso, algumas definições são necessárias:

1. Entende-se por *imagem* a imagem de entrada, aqui o resultado da Limiarização de Otsu;
2. Entende-se por *output* a imagem segmentada, saída do algoritmo;
3. Chama-se de *p* o pixel analisado dentro da imagem;
4. Chama-se de *pixel* um pixel a ser comparado com *p*;
5. Tem-se por *q* um valor de intensidade a ser guardado;
6. A função *find* é uma função auxiliar responsável por encontrar um valor de pixel dentro de uma imagem.

---

**Algorithm 2** Algoritmo de *Watershed* baseado em componentes conectados.
 

---

```

1: def find(img, u):
2:   r = u
3:   rows, cols = img.shape
4:   for p in img do
5:     if img[i][j] != r:
6:       r = img[i][j]
7:       w = u
8:       if w != r:
9:         tmp = img[i][j]
10:        img[i][j] = r
11:        w = tmp
12:   end for
13:   return r
14:
15: for p in imagem do
16:   q ← p
17:   for pixel in vizinhança do
18:     if f(pixel) < f(q)
19:       q ← p'
20:   end for
21:   if q != p:
22:     output[p] ← q
23:   else:
24:     output[p] ← -1
25: end for
26:
27: for p in imagem do
28:   if output(p) = -1:
29:     for pixel in vizinhança do
30:       if (output(p) != -1 and imagem(p) == imagem(pixel)):
31:         adicionar pixel na fila
32:     end for
33: end for
34:
35: while fila_empty() = False:
36:   fila.append(p);
37:   for pixel in N(p) do
38:     if output(pixel) == -1:
39:       output(pixel) ← p
40:       fila.append(pixel)
41:   end for
42:
43: for p in imagem do
44:   if output[p] = -1:
45:     output[p] ← p
46:     for pixel in vizinhança do
47:       if f(pixel) = f(p):
48:         r ← find(output, p)
49:         s ← find(output, pixel)
50:         output(r) = min(r,s)
51:         output(s) = min(r,s) end for

```



### 3.3.1.3 *Watershed* baseado em código de cadeia

Esta abordagem adota código em cadeia para representar os componentes conectados. Assim como a proposta anterior, se baseia em uma analogia de chuva, mas a simula para gerar componentes conectados usando código de cadeia ao invés do endereço de pixel. Em seguida, simula inundações para as rotular bacias de captação rastreando códigos de cadeia. Também percorre a imagem apenas quatro vezes e pode segmentar a imagem completamente. De acordo com (HAN SUN JINGYU YANG, 2005), se comparada à anterior, pode descrever componentes conectados mais claramente e fornecer mais informações para etapas futuras do processamento.

Seja  $f(p)$  o valor cinza do pixel atual  $p$ ,  $f(p')$  o valor de cinza do pixel vizinho  $p'$ , e  $N(p)$  ser o conjunto de pixels vizinhos. Então, o vizinho mais íngreme  $q$  é definido como  $q = \{p' \in N(p) | f(p') = \min_{p'' \in N(p)} f(p''), f(p') < f(p)\}$ . Ou seja, o código de cadeia do pixel atual  $p$  deve apontar para o vizinho mais inclinado  $q$ .

Os pixels restantes não têm vizinhos inferiores, então são designados de modo diferente. Esses pixels são localizados em dois lugares: planaltos mínimos e não mínimos. Os mais escuros recebem um rótulo de código de cadeia de pontos. Logo, os pixels em não mínimos podem ser rotulados propagando-se a partir desses pixels de borda por meio de uma fila. Escolhe-se o primeiro pixel em cada mínimo, atribuindo um valor de cinza a ele e a seus vizinhos. Em seguida, esses vizinhos têm código de cadeia point-out e podem ser encaixados na categoria anterior. A lógica implementada está presente no Algoritmo 3 e o algoritmo pode ser descrito em quatro etapas, sendo elas:

1. Rotular os códigos de cadeia *point in* e *point out* que tenham pixels em planaltos mínimos;
2. Rotular os códigos de cadeia *point in* e *point out* que tenham pixels em planaltos não mínimos;
3. Rotular os códigos de cadeia *point in* e *point out* que tenham pixels localizados nos planaltos mínimos;
4. Rotular os códigos de cadeia *point in* e *point out* que tenham pixels localizados nos planaltos não mínimos.

---

**Algorithm 3** Algoritmo de *Watershed* baseado em código de cadeia.
 

---

```

1: inicializar PointOut() em 8;
2: inicializar PointIn() em 0.
3:
4: for p in imagem do
5:   if q existe {
6:     PointOut(p) = PO_Code( $p \rightarrow q$ )
7:     PointIn(q) = PI_Code( $p \leftarrow q$ ) }
8: end for
9:
10: for p in imagem do
11:   if (PointOut(p) != 8 and PointOut(pixel) == 8 and imagem(p) == imagem(pixel))
12:     adicionar p na fila
13: end for
14:
15: for p in imagem do
16:   While (fila_empty() == False) {
17:     p = fila_get()
18:     for pixel in vizinhança do
19:       if(f(p) == f(pixel) and PointOut(pixel) == 8) {
20:         PointOut(pixel) = PO_Code( $pixel \rightarrow p$ );
21:         PointIn(p) = PI_Code( $p \leftarrow pixel$ )
22:         adicionar pixel à fila
23:       } end for }
24: end for
25:
26: for p in imagem do
27:   for pixel in vizinhança do
28:     If (PointOut(p) == 8 and PointOut(pixel) == 8 and imagem(p) == imagem(pixel)) {
29:       PointOut(pixel) = PO_Code( $pixel \rightarrow p$ )
30:       PointIn(p) = PI_Code( $p \leftarrow pixel$ )
31:       fila_push(pixel) }
32: end for
33:
34: label = 1
35: for p in imagem do
36:   for pixel in vizinhança do
37:     If (PointOut(p) == 8)
38:       fila_push(p)
39:       while (fila_empty() == False){
40:         pixel = fila_pop();
41:         segmentada(pixel) = label }
42:   end for
43:   label = label + 1
44: end for

```

---

## 3.4 Estimação da posição da ponta da agulha

O primeiro método de estimação da ponta da agulha é a *Transformada de Hough*. Trata-se de uma técnica para detecção de linhas, escolhida para explorar a existência da CTA nas imagens obtidas por meio do *phantom* em banho d'água. Como estas imagens não são o foco deste trabalho, optou-se por não seguir adiante com este método de estimação, priorizando métodos direcionados às imagens oriundas dos simuladores de gelatina, que não contém tal artefato.

A estimação da posição da ponta da agulha tem, a partir de então, uma abordagem simples proposta, consistente na detecção do *máximo local de cinza da região segmentada*, representado pela região mais brilhante da imagem. Dada a abordagem fora do plano de aquisição de imagens e a reflexão mais forte da onda ultrassônica ocorrer na fronteira entre o meio e a agulha, teoricamente, o máximo local de cinza deve representar a posição da agulha na imagem.

### 3.4.1 Filtro de Kalman

Ao fim da estimação da posição da agulha, foi mantida a implementação de um *Filtro de Kalman*, como em (S. DOS REIS, 2019), a fim de prover uma nova estimativa para a identificação da ponta da agulha. Espera-se que as estimativas do filtro correspondam às do algoritmo, com variações suaves, cancelamento de ruídos e resistência a variações abruptas e erros pontuais. Para a implementação do filtro, as seguintes considerações foram feitas:

1. A estimação do filtro de Kalman, bem como a estimação feita pelos métodos do algoritmo, será comparada com o valor real medido pelo sensor NDI Aurora em (KIYOKU, 2015), sobre o qual serão detalhadas mais informações no Capítulo 4;
2. Os valores de covariância de ruído da medição devem atingir um "valor ótimo", tais que a matriz de ganho  $K_k$  não tenha valores tão elevados a ponto de comprometer a velocidade de sensibilidade e convergência do filtro, mas não tão baixos tais que as estimativas tenham grandes variações entre si.
3. A estimativa inicial do filtro é o centro da imagem, que, por sua vez, também é o centro da região de interesse. A partir desta atribuição inicial, acredita-se que o filtro passe a precisar de um menor tempo de ajuste para convergir aos valores obtidos pelos métodos de estimação já citados.

Desse modo, ao fim da implementação do Filtro de Kalman, tem-se uma nova proposta de algoritmo de rastreamento, cujo resumo pode ser visto na Figura 20. Em azul, estão destacadas todas as etapas nas quais houve alteração.

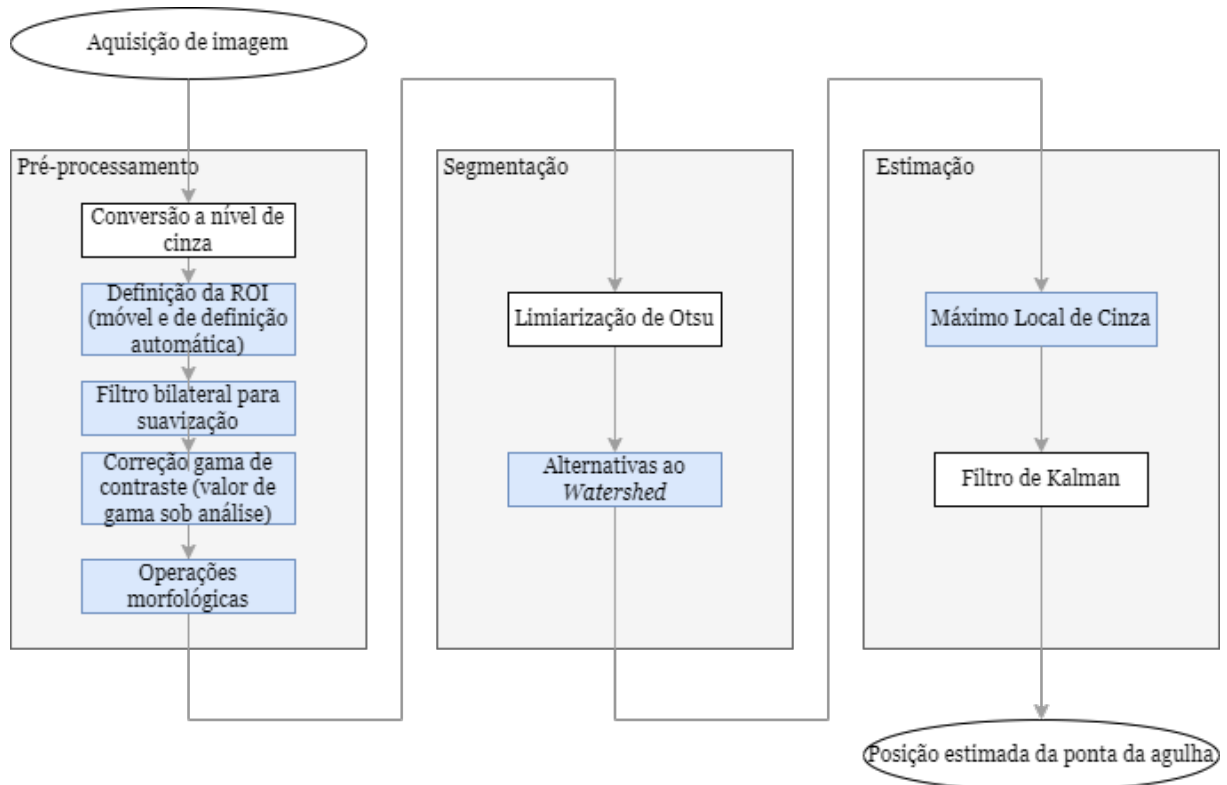


Figura 20 – Resumo do processo de estimação da ponta da agulha após propostas de alteração.

Fonte: Produzido pelo autor.

### 3.5 Custo Computacional

Dado que o objetivo do projeto é a redução do custo computacional do mesmo e, por consequência, sua aplicabilidade em tempo de execução, a partir do algoritmo proposto, considera-se cada operação como uma tarefa e realiza-se a medição dos tempos de execução das mesmas para que se avalie se elas são ou não compatíveis com uma aplicação online.

A taxa de exibição de quadros por segundo (FPS, ou *Frames Per Second*) das imagens adquiridas com o sistema de imageamento de ultrassom é de 30 FPS. Em uma aplicação ideal, 30 frames de imagem seriam processados em 1 segundo, o que implicaria em um deadline de 33,3ms para a estimação da posição da ponta da agulha em um frame. É um deadline apertado. Uma alternativa a isso é não processar todos os frames, claro que analisando o *trade-off* entre precisão e economia em custo computacional, podendo trabalhar, assim, com um deadline maior. De todo modo, fazendo o processamento de todos eles, o ideal é que os tempos de execução sejam os menores possíveis.

Em vista disso, a medição foi realizada utilizando o módulo *timeit*. Esta biblioteca gera resultados muito mais precisos que a biblioteca *time*, nativa de tantas linguagens de programação. A função *time()* retorna o tempo atual em segundos desde 1/1/1970. Isso considera lixo, outras atividades que o computador esteja fazendo, interrupções do usuário e

o valor retornado peca em precisão, além de a própria chamada da função *time()* já trazer consigo um *overhead*. Já a função *timeit* foi projetada em específico para medir o tempo de execução de funções e trechos de código.

O módulo define uma classe pública e três funções, das quais duas são aqui utilizadas. São elas:

1. *timeit.timeit("função", origem)*. Esta primeira função cria um *timer* com a instrução a ser medida e sua origem;
2. *timeit.repeat(repeat=5, number=1000000)*. Esta função faz com que se repita 5 vezes um ciclo de 1000000 execuções da função solicitada na função anterior. Tem-se como retorno o tempo médio de cada um dos ciclos de execução.

De posse dessas funções, as seguintes medições são feitas para as dimensões de região de interesse de  $200 \times 200$ ,  $150 \times 150$  e  $100 \times 100$  pixels:

1. Medição do tempo de definição da ROI para as três dimensões propostas;
2. Medição dos tempos de execução das operações realizadas no pré-processamento;
3. Medição dos tempos das três alternativas de segmentação aqui trabalhadas;
4. Medição do tempo de execução da estimação da ponta da agulha;
5. Cálculo do custo computacional do rastreamento.

## 4 Resultados

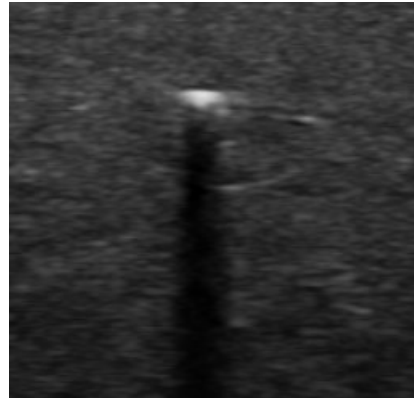
A análise das soluções propostas, tanto em trabalhos anteriores quanto no presente trabalho, se dá por meio de três vertentes:

1. *Análise qualitativa* do resultado das técnicas de processamento de imagem. Isto é, serão analisadas qualitativamente a suavização de ruídos, a preservação das bordas, a aplicação de operações morfológicas e como essas técnicas funcionam juntas, de modo a favorecer a segmentação da agulha e posterior estimação de sua posição;
2. *Eficácia de medição*. Devem ser comparadas as medições brutas, encontradas pelo algoritmo, as medições filtradas, obtidas por meio da aplicação do Filtro de Kalman, e a medição referência, feita utilizando o sensor NDI Aurora em (KIYOKU, 2015);
3. *Custo computacional do processo*. Independentemente de uma abordagem ser bem sucedida na estimação da posição da agulha, deve-se lembrar que almeja-se um sistema operante a tempo de execução. Desse modo, o tempo de execução das operações será medido e avaliado, a fim de validar se as técnicas empregadas podem ser utilizadas em uma aplicação online.

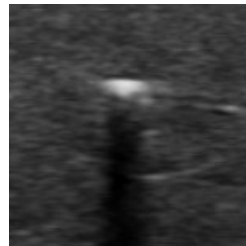
Importante ressaltar que, neste trabalho, são desconsideradas as operações realizadas em banho d'água. As considerações feitas na seção a seguir tem como base o banco de imagens desenvolvido em (KIYOKU, 2015) e dizem respeito às operações desenvolvidas com imagens referentes ao *phantom* opaco. Assume-se que, por ser mais ruidoso, o que for eficaz para o meio fibroso também será para o meio translúcido. O meio translúcido será levado em consideração apenas na análise da operação de abertura, que não é feita para o meio opaco.

### 4.1 Análise qualitativa das técnicas de processamento de imagens

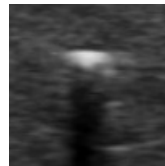
O primeiro passo para a análise qualitativa das técnicas de processamento de imagens é a comparação entre os três tamanhos de região de interesse. Eles podem ser vistos na Figura 21 a seguir.



(a) Região de interesse  $200 \times 200$ .



(b) Região de interesse  $150 \times 150$ .



(c) Região de interesse  $100 \times 100$ .

Figura 21 – Regiões de interesse de dimensões  $200 \times 200$ ,  $150 \times 150$  e  $100 \times 100$  pixels de imagem adquirida em *phantom* opaco .

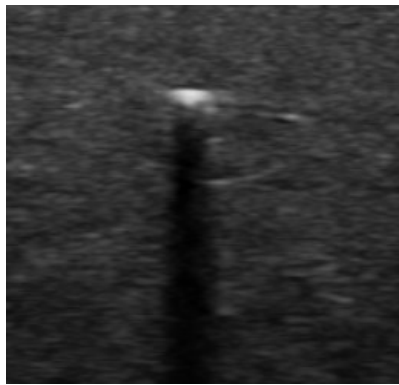
Fonte: Produzido pelo autor.

A análise será feita a seguir utilizando uma região de interesse de  $200 \times 200$  pixels, para uma melhor visualização do resultado das operações. No entanto, ressalta-se que as operações são igualmente bem sucedidas para os três tamanhos de ROI, do ponto de vista qualitativo.

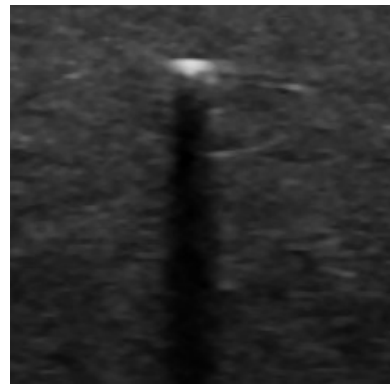
#### 4.1.1 Pré-processamento

##### 4.1.1.1 Filtros de suavização

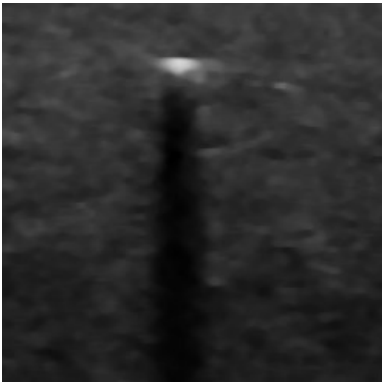
As imagens foram submetidas à aplicação de filtros de mediana de máscaras  $m = 3, 5, 7, 9$ . Além disso, implementou-se, também, um filtro bilateral, com valores de  $d = 5, 7, 10$ . Por sugestão do *OpenCV*, os valores de  $\sigma$  do filtro bilateral foram fixados em 10, para que o filtro não se torne lento. O resultado da filtragem para um frame de imagem proveniente do simulador opaco pode ser vista na Figura 22.



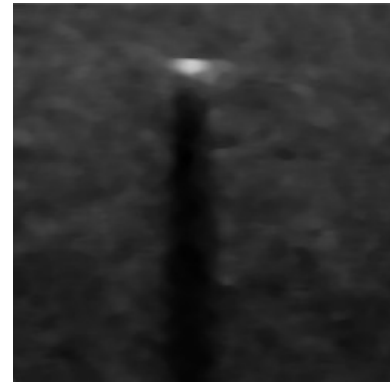
(a) Região de interesse em imagem de *phantom opaco*.



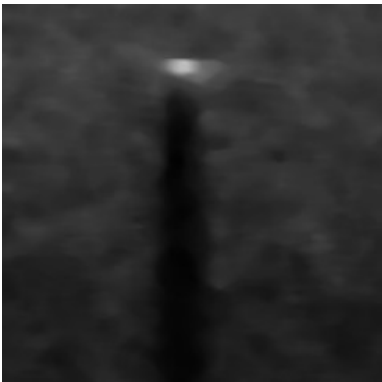
(b) Aplicação de filtro de mediana com máscara de dimensão 3.



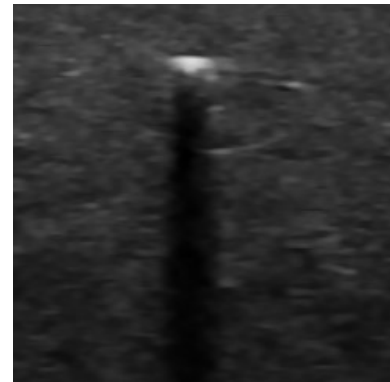
(c) Aplicação de filtro de mediana com máscara de dimensão 5.



(d) Aplicação de filtro de mediana com máscara de dimensão 7.



(e) Aplicação de filtro de mediana com máscara de dimensão 9.



(f) Aplicação de filtro bilateral com máscara  $d = 5$ .



(g) Aplicação de filtro bilateral com máscara  $d = 7$ .



(h) Aplicação de filtro bilateral com máscara  $d = 10$ .

Figura 22 – Aplicação de filtros de mediana e filtros bilaterais em região de interesse de imagem adquirida em *phantom opaco*.



Observa-se, tendo a ROI na Figura 23a como referência, que a aplicação de filtro de mediana tendo a máscara igual a 3 tem pouco efeito na suavização dos ruídos. Por outro lado, a máscara igual a 9 já compromete as bordas de modo muito significativo. A escolha da máscara para o filtro de mediana entre os valores de 5 e 7 já passaria a depender de como a correção gama resgataria o contorno das bordas da agulha.

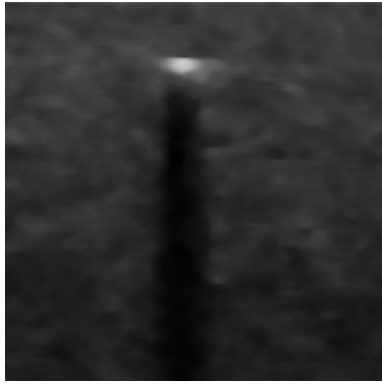
Analisando a aplicação de filtros bilaterais, observa-se que a aplicação da máscara  $d = 10$ , presente na Figura 22h, apresenta suavização semelhante ao filtro de mediana com máscara de dimensão 7, mostrado na Figura 22d, com maior preservação da área brilhante correspondente à posição da agulha. Opta-se, então, por seguir com o filtro de mediana com máscara igual a 7 e com o filtro bilateral com  $d = 10$ , associando a escolha final do filtro de suavização à interação com as demais operações e ao custo computacional dos mesmos, a ser medido posteriormente.

#### 4.1.1.2 Operações de realce

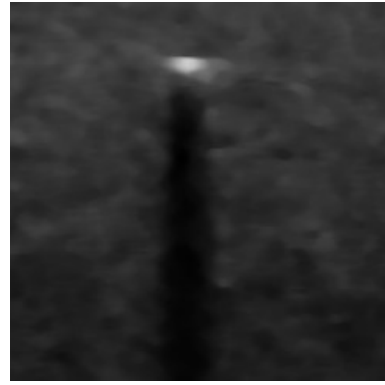
##### 4.1.1.2.1 Correção gama

Após a suavização de ruídos, analisou-se a correção gama para o realce das bordas. Para a imagem suavizada com o filtro de mediana de máscara igual a 7 e o filtro bilateral com  $d = 10$ , aplicou-se a correção gama para valores de  $\gamma$  iguais a 1.3, 1.5 e 1.7 e analisou-se os resultados de modo qualitativo. Pode-se ver os resultados nas figuras 23 e 24.

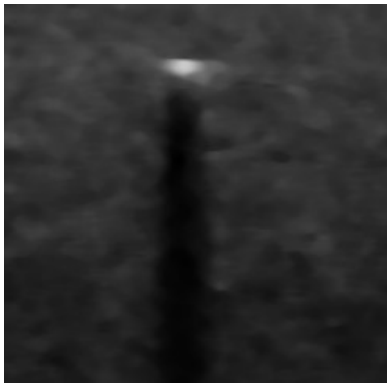
Analisando a aplicação da correção gama, opta-se pelo filtro bilateral por ver, nele, uma maior preservação das bordas com o mesmo nível de suavização de ruído, como já esperado dada a fundamentação teórica presente no Capítulo 2. No entanto, a escolha do valor de  $\gamma$  dependerá dos demais critérios de análise. De acordo com (S. DOS REIS, 2019), imagens mais claras, ou seja, com  $\gamma$  menor, acarretam uma segmentação mais difusa, porém com erro menor. Para valores de  $\gamma$  maiores, o contrário acontece.



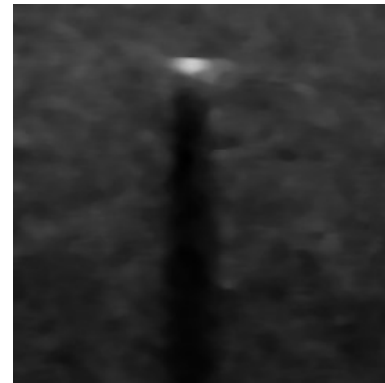
(a) Imagem de *phantom* opaco após aplicação de filtro de mediana de máscara 7.



(b) Correção gamma com  $\gamma = 1.3$ .



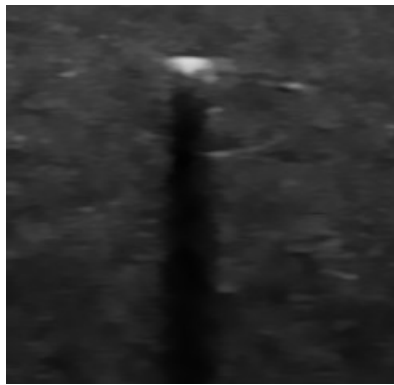
(c) Correção gamma com  $\gamma = 1.5$ .



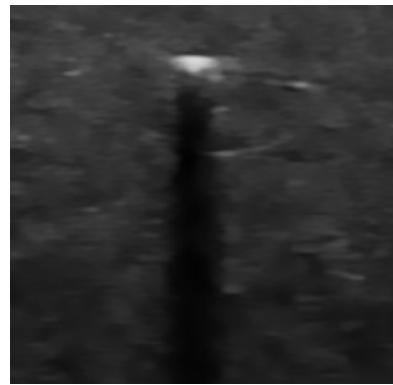
(d) Correção gamma com  $\gamma = 1.7$ .

Figura 23 – Aplicação da correção gama para valores de  $\gamma$  diferentes em imagem de *phantom* opaco submetida a filtro de mediana.

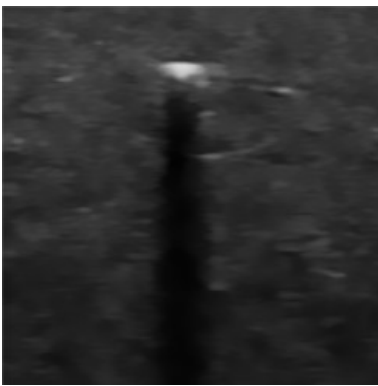
Fonte: Produzido pelo autor.



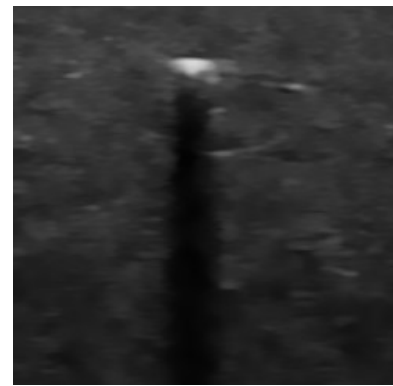
(a) Imagem de *phantom* opaco após aplicação de filtro bilateral com  $d = 10$ .



(b) Correção gamma com  $\gamma = 1.3$ .



(c) Correção gamma com  $\gamma = 1.5$ .



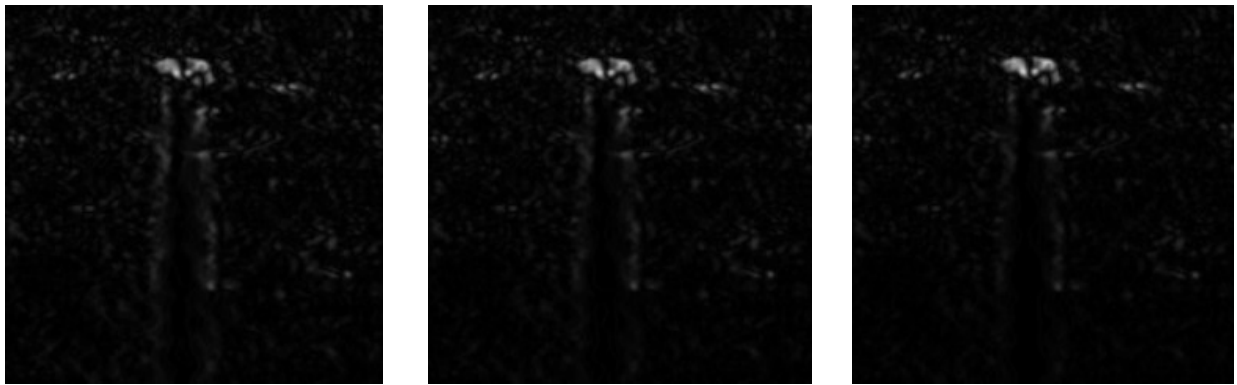
(d) Correção gamma com  $\gamma = 1.7$ .

Figura 24 – Aplicação da correção gama para valores de  $\gamma$  diferentes em imagem de *phantom* opaco submetida a filtro bilateral.

Fonte: Produzido pelo autor.

#### 4.1.1.2.2 Gradiente

O passo final para o realce das bordas é a aplicação do gradiente na imagem com contraste corrigido. Pode-se ver, na Figura 25, o resultado da aplicação do gradiente nas imagens resultantes da correção gamma após filtro bilateral. Observa-se que, na região da ponta da agulha, há uma pequena descontinuidade. Pode-se cogitar aplicar uma operação de fechamento, mas isso atrapalha a limiarização que leva à segmentação. Isso está ilustrado na Figura 26. Utilizando a Figura 25c como exemplo, mostra-se na Figura 26 o resultado de uma operação de fechamento que, antes de mais nada, "desfaz o trabalho" do gradiente em ressaltar as bordas. Em seguida, tem-se o resultado da limiarização de Otsu com e sem a operação de fechamento. Nota-se que, não sendo precedida de um fechamento, a limiarização é bem sucedida em separar a agulha do plano de fundo, o que não ocorre em caso contrário.



(a) Aplicação de gradiente após correção  $\gamma = 1.3$ .

(b) Aplicação de gradiente após correção  $\gamma = 1.5$ .

(c) Aplicação de gradiente após correção  $\gamma = 1.7$ .

Figura 25 – Aplicação de gradiente para realce de bordas após correção gama.

Fonte: Produzido pelo autor.



(a) Aplicação de fechamento após operação de gradiente.

(b) Resultado da operação de limiarização após fechamento.

(c) Resultado da operação de limiarização sem fechamento precedente.

Figura 26 – Execução da operação de limiarização, que precede a segmentação da imagem.

Fonte: Produzido pelo autor.

## 4.1.2 Segmentação

As duas primeiras abordagens de segmentação aqui desenvolvidas têm como ponto de partida a região de interesse imediatamente após a limiarização de Otsu, presente na Figura 26c. Já o watershed por código de cadeia tem por ponto de partida a operação de gradiente, pulando a etapa de limiarização.

### 4.1.2.1 Watershed Tradicional

O chamado *Watershed* tradicional, utilizado nas abordagens até aqui, é realizado em etapas. A partir da limiarização, ocorrem as seguintes etapas, a serem vistas na Figura 27.

1. Aplicação de abertura, para fins de remoção de pequenos ruídos brancos remanescentes (Figura 27a);

2. Determinação da área que se tem certeza que está no plano de fundo da imagem (Figura 27b);
3. Determinação da área que se tem certeza que está no plano de frente da imagem (Figura 27c);
4. Determinação da área de incerteza (Figura 27d);
5. Atribuição de marcadores para rotular as regiões. Regiões de certeza, seja no primeiro plano ou no plano de fundo, tem como rótulos um valor inteiro positivo. Áreas de incerteza tem rótulo igual a 0;
6. Aplicação do *Watershed* e atribuição do rótulo -1 às linhas de divisão e de uma cor qualquer, a ser definida pelo usuário, à posição dessas linhas na imagem original (Figura 27e).

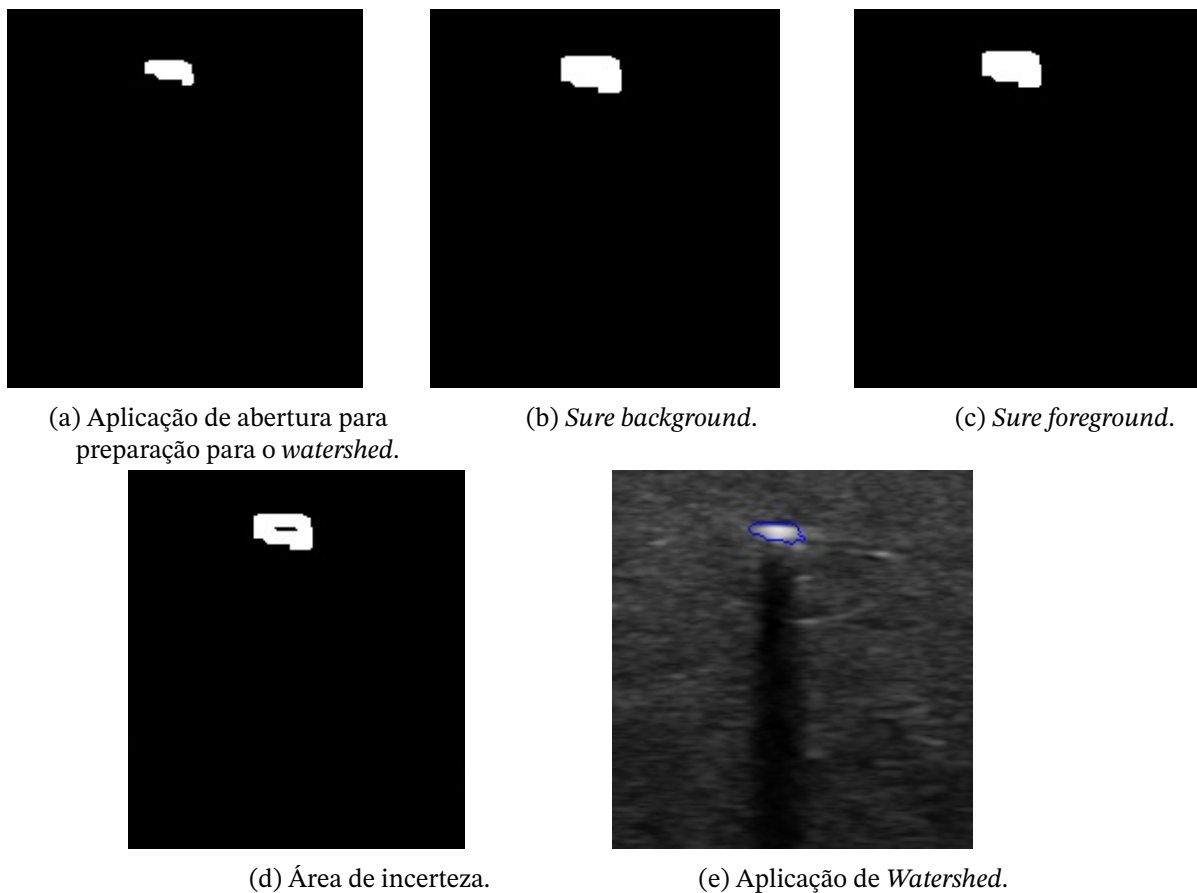


Figura 27 – Execução de *Watershed*.

Fonte: Produzido pelo autor.

#### 4.1.2.2 Watershed baseado em componentes conectados

A ideia do algoritmo proposto tem sua origem nos componentes conectados. O objetivo é rotular cada pixel com o valor representante da região à qual pertence. Escolhendo,

para cada pixel  $p$ , um vizinho como predecessor, tem-se como resultante um gráfico direcionado. No entanto, os pixels mínimos não têm uma vizinhança mais inclinada e, para eles, introduz-se outra relação de conectividade: todos os vizinhos de um pixel mínimo  $p$  com o mesmo nível de cinza que ele pertencem ao mesmo componente.

Consequentemente, estes pixels se unem de tal forma que o representante do mínimo regional é o pixel com o menor valor de endereço. Uma vez que este grafo é construído, seus componentes conectados são computados. Tem-se como entrada, portanto, além da imagem de entrada  $f$ , uma imagem  $l$ , que armazena, para cada pixel, seu rótulo. O algoritmo consiste em três varreduras, uma a menos que o Watershed em código de cadeia e mais eficiente que o Watershed tradicional.

Partindo da imagem 26c, o resultado desta segmentação pode ser visto na Figura 28. Pode-se ver que são detectados pelo algoritmo como objetos de segmentação a ponta da agulha e as bordas da imagem.

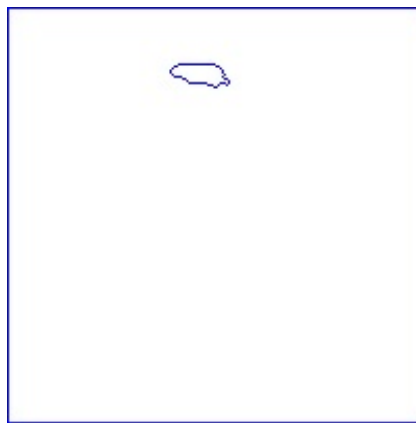
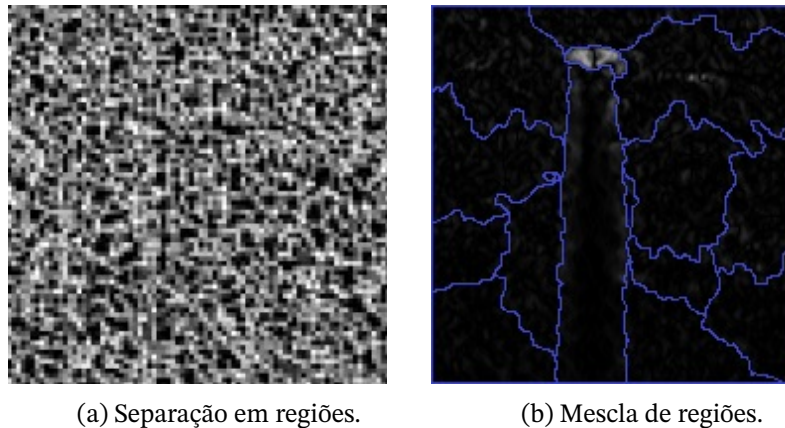


Figura 28 – Execução de Watershed via componentes conectados.

Fonte: Produzido pelo autor.

#### 4.1.2.3 Watershed baseado em código de cadeia

O algoritmo de segmentação baseado em código de cadeia tem como ponto de partida a operação de gradiente, na imagem, presente na Figura 25. A partir dela, o algoritmo primeiramente separa a imagem em regiões, depois mescla essas regiões baseado em sua similaridade. Desse modo, a segmentação ocorre em duas etapas, a serem vistas na figura 29.



(a) Separação em regiões.

(b) Mescla de regiões.

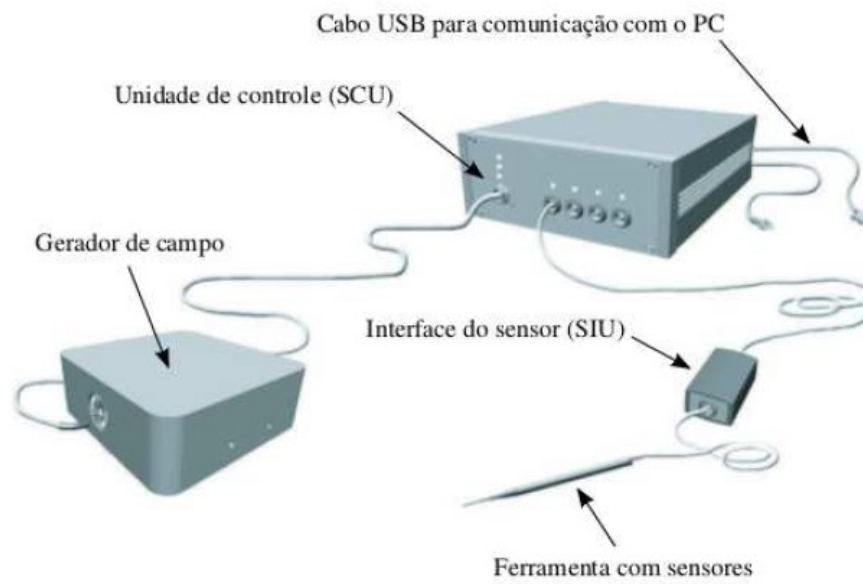
Figura 29 – Execução de *Watershed* baseado em código de cadeia.

Fonte: Produzido pelo autor.

De acordo com o algoritmo, primeiramente a imagem com a aplicação de gradiente foi separada em 1874 regiões. Posteriormente, com a operação de *merge*, o algoritmo segmentou com sucesso a ponta da agulha, mas também detectou o artefato de sombra acústica e separou o restante da imagem em 14 regiões. Com a estimação da posição via máximo local de cinza, pôde-se encontrar com sucesso a posição da ponta, no entanto, deve-se observar que o alto ruído da imagem interfere muito no processo e pode atrapalhar, levando a segmentações incorretas.

## 4.2 Eficácia de medição

A análise do sucesso das medições é feita tendo como referência a medição da posição da agulha por meio do sistema de rastreamento eletromagnético *NDI Aurora*. Trata-se de um sistema composto por um cateter e um sensor e um gerador de campo planar. O cateter e o sensor possuem posição e orientação em relação a um eixo de referência no gerador de campo. O sensor de 0.8mm foi posicionado dentro de uma agulha tubular de Nitinol de diâmetro externo de 1,14mm e interno de 0,96mm. Este cateter, por sua vez, foi posicionado no transdutor a 14mm da ponta da agulha. As imagens presentes na Figura 30 foram tiradas de (KIYOKU, 2015) e mostram como foi feita a preparação para a medição.



(a) Sistema de rastreamento eletromagnético NDI Aurora.

Fonte: (GERALDES, 2015)



(b) Cateter Aurora.

Fonte: (GERALDES, 2015)



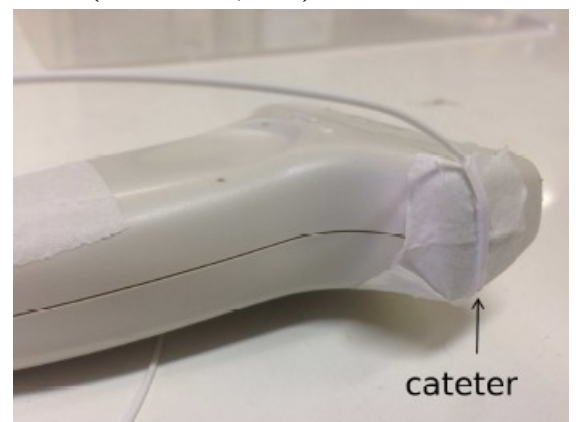
(c) Sensor Aurora.

Fonte: (GERALDES, 2015)



(d) Gerador de campo planar.

Fonte: (GERALDES, 2015)



(e) Cateter acoplado ao transdutor.

Fonte: (KIYOKU, 2015)

Figura 30 – Rastreamento Eletromagnético da ponta da agulha utilizando o sensor NDI Aurora.



De posse dessas informações, é possível tomar a posição determinada pelo sensor como posição real, de referência, e converter as coordenadas do plano da imagem em coordenadas reais. A Figura 31 mostra os resultados da medição simultânea do sensor eletromagnético, da transformada de Hough e do máximo local de cinza da posição da ponta da agulha em um simulador de gelatina translúcida, estando a posição medida pelo sensor em azul, pela transformada de Hough em verde e máximo local de cinza em vermelho.

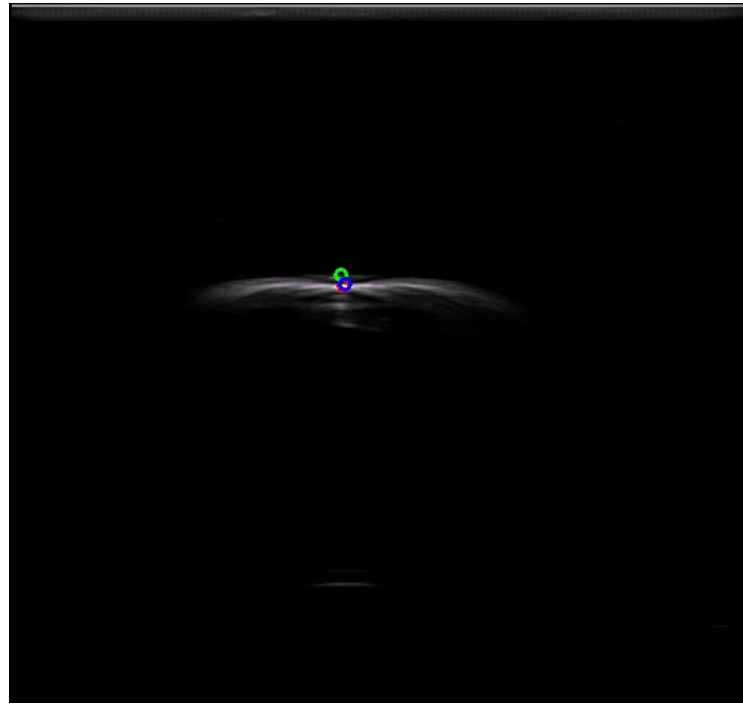


Figura 31 – Medição simultânea do sensor eletromagnético, da transformada de Hough e do máximo local de cinza da posição da ponta da agulha em um simulador de gelatina translúcida.

Fonte: (KIYOKU, 2015)

Diante deste cenário, junto ao banco de imagens existente neste trabalho, há também arquivos que guardam as posições detectadas pelo sensor durante a movimentação da agulha em *phantoms* em três tipos de ensaio: ensaio parado, no qual a agulha fica fixa numa posição e se avalia a sua localização encontrada, ensaio horizontal, correspondente a uma movimentação horizontal da agulha, e ensaio vertical, correspondente à movimentação vertical da agulha. Tendo uma medida base, gera-se um sinal de erro a partir das estimações feitas.

Partindo deste *setup*, são calculadas as seguintes medidas de erro para cada um dos tamanhos de ROI:

1. Erro de estimação de posição para *Watershed* tradicional (a ser chamado de *Watershed* 1) sem filtro de Kalman;
2. Erro de estimação de posição para *Watershed* tradicional (a ser chamado de *Watershed* 1) com adição do filtro de Kalman;

3. Erro de estimação de posição para *Watershed* baseado em componentes conectados (a ser chamado de *Watershed 2*) sem filtro de Kalman;
4. Erro de estimação de posição para *Watershed* baseado em componentes conectados (a ser chamado de *Watershed 2*) com adição do filtro de Kalman;
5. Erro de estimação de posição para *Watershed* baseado em código de cadeia (a ser chamado de *Watershed 3*) sem filtro de Kalman;
6. Erro de estimação de posição para *Watershed* baseado em código de cadeia (a ser chamado de *Watershed 3*) com adição do filtro de Kalman.

Os resultados da medição dos erros médio e máximo para cada um desses cenários se encontram nas Tabelas de 4 a 12.

Tabela 4 – Tabela de erros de medição do ensaio parado para ROI de  $200 \times 200$  pixels.

<b>Ensaio Parado</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	1.0	1.6
2. Watershed por componentes conectados	0.7	1.3
3. Watershed por código de cadeia	0.9	1.5
Filtro de Kalman + Watershed 1	0.9	1.1
Filtro de Kalman + Watershed 2	0.6	1.2
Filtro de Kalman + Watershed 3	0.7	1.0

Fonte: Produzido pelo autor.

Tabela 5 – Tabela de erros de medição do ensaio horizontal para ROI de  $200 \times 200$  pixels.

<b>Ensaio Horizontal</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	2.6	5.8
2. Watershed por componentes conectados	1.6	3.7
3. Watershed por código de cadeia	2.0	4.6
Filtro de Kalman + Watershed 1	1.8	5.6
Filtro de Kalman + Watershed 2	1.8	3.7
Filtro de Kalman + Watershed 3	1.8	4.6

Fonte: Produzido pelo autor.

Tabela 6 – Tabela de erros de medição do ensaio vertical para ROI de  $200 \times 200$  pixels.

<b>Ensaio Vertical</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	3.3	8.1
2. Watershed por componentes conectados	3.5	8.9
3. Watershed por código de cadeia	2.8	6.8
Filtro de Kalman + Watershed 1	2.3	5.7
Filtro de Kalman + Watershed 2	3.3	8.6
Filtro de Kalman + Watershed 3	2.8	6.6

Fonte: Produzido pelo autor.

Tabela 7 – Tabela de erros de medição do ensaio parado para ROI de  $150 \times 150$  pixels.

<b>Ensaio Parado</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	1.0	1.5
2. Watershed por componentes conectados	0.8	1.5
3. Watershed por código de cadeia	0.9	1.7
Filtro de Kalman + Watershed 1	0.9	1.0
Filtro de Kalman + Watershed 2	0.5	1.1
Filtro de Kalman + Watershed 3	0.7	1.0

Fonte: Produzido pelo autor.

Tabela 8 – Tabela de erros de medição do ensaio horizontal para ROI de  $150 \times 150$  pixels.

<b>Ensaio Horizontal</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	2.4	5.7
2. Watershed por componentes conectados	1.3	3.4
3. Watershed por código de cadeia	2.0	4.4
Filtro de Kalman + Watershed 1	1.8	5.3
Filtro de Kalman + Watershed 2	1.8	3.2
Filtro de Kalman + Watershed 3	1.8	3.7

Fonte: Produzido pelo autor.

Tabela 9 – Tabela de erros de medição do ensaio vertical para ROI de  $150 \times 150$  pixels.

<b>Ensaio Vertical</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	3.3	8.1
2. Watershed por componentes conectados	3.5	8.6
3. Watershed por código de cadeia	2.7	6.8
Filtro de Kalman + Watershed 1	2.1	5.3
Filtro de Kalman + Watershed 2	3.0	8.2
Filtro de Kalman + Watershed 3	2.6	6.5

Fonte: Produzido pelo autor.

Tabela 10 – Tabela de erros de medição do ensaio parado para ROI de  $100 \times 100$  pixels.

<b>Ensaio Parado</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	1.1	1.5
2. Watershed por componentes conectados	0.7	1.2
3. Watershed por código de cadeia	0.9	1.5
Filtro de Kalman + Watershed 1	0.9	1.0
Filtro de Kalman + Watershed 2	0.5	1.2
Filtro de Kalman + Watershed 3	0.7	1.0

Fonte: Produzido pelo autor.

Tabela 11 – Tabela de erros de medição do ensaio horizontal para ROI de  $100 \times 100$  pixels.

<b>Ensaio Horizontal</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	2.6	5.8
2. Watershed por componentes conectados	1.6	3.7
3. Watershed por código de cadeia	2.0	4.6
Filtro de Kalman + Watershed 1	1.8	5.6
Filtro de Kalman + Watershed 2	1.8	3.7
Filtro de Kalman + Watershed 3	1.8	4.6

Fonte: Produzido pelo autor.

Tabela 12 – Tabela de erros de medição do ensaio vertical para ROI de  $100 \times 100$  pixels.

<b>Ensaio Vertical</b>		
Método	Erro	
	Médio (mm)	Máximo (mm)
1. Watershed Tradicional	3.3	8.1
2. Watershed por componentes conectados	3.5	8.9
3. Watershed por código de cadeia	2.8	6.8
Filtro de Kalman + Watershed 1	2.3	5.7
Filtro de Kalman + Watershed 2	3.3	8.6
Filtro de Kalman + Watershed 3	2.8	6.6

Fonte: Produzido pelo autor.

Observa-se que, de modo geral, o Filtro de Kalman é eficiente na suavização da estimativa e, especialmente, na redução do valor do erro médio. Observa-se também que o método de Watershed baseado em componentes conectados apresenta menor erro se comparado aos demais. O erro se torna menor especialmente quando se trabalha com uma ROI de dimensões  $150 \times 150$  pixels. Quando se analisa o processamento com uma ROI de  $100 \times 100$ , observa-se que o erro chega a aumentar em alguns cenários. Atribui-se isso a conversão de coordenadas.

Além disso, deve-se observar que a dispersão no ensaio vertical é elevada. Acredita-se que isso se deve ao fato de a medição ocorrer em um plano de coordenadas 2D e o ensaio vertical movimentar a agulha justamente na coordenada que é perdida. Para o plano 2D, a agulha estaria parada, o que não é verdade. Isso pode ser uma fonte de erro e maiores dispersões, gerando uma maior disparidade entre os erros médio e máximo.

Diante dessas considerações, baseando-se apenas no erro, pode-se dizer que a melhor alternativa seria trabalhar com uma ROI de  $150 \times 150$  pixels e o Watershed baseado em componentes conectados. Para consolidá-los como os melhores métodos, deve-se avaliar o custo computacional do projeto, de modo a ver sua compatibilidade com uma aplicação online.

### 4.3 Custo Computacional

O requisito final para a validação de um método para a estimativa da posição da ponta da agulha é sua compatibilidade com uma aplicação em tempo de execução. Almejando este objetivo, ponto a ponto, mediu-se o tempo de execução de cada tarefa e considerações e propostas foram feitas de modo a melhorar o tempo de execução da aplicação.

### 4.3.1 Aquisição e transmissão de imagens

Antes desconsiderados, devem passar a figurar dentre as variáveis os tempos de aquisição e transmissão da imagem. Esta é a razão pela qual o presente trabalho precisou compreender o funcionamento do sistema de imageamento via Ultrassom utilizado para adquirir as imagens. Através da implementação da aplicação cliente do *OpenIGTLink*, o tempo de aquisição de um frame de imagem por um computador periférico foi estimado, sendo de  $14,66ms$ . A partir de então, este tempo não poderá mais ser desprezado.

### 4.3.2 Diminuição da ROI

Como já mencionado, estudou-se trabalhar com regiões de interesse menores. A Tabela 13 mostra o tempo de execução da definição da região de interesse com as dimensões já descritas.

Tabela 13 – Tempo de execução da definição de regiões de interesse.

Tamanho da ROI (pixels)	Tempo de execução (ms)
200 × 200	1,32
150 × 150	1,13
100 × 100	0,89

Fonte: Produzido pelo autor.

É intuitivo dizer que trabalhar com regiões de interesse menores gera uma economia de custo computacional. Está agora comprovado pela Tabela 13. Como a diferença entre os tempos é ínfima, a escolha é feita tendo como critério o erro de medição gerado e a economia que a redução da ROI tende a gerar nas demais operações.

### 4.3.3 Medição dos tempos de execução das tarefas

A análise dos tempos de execução é fundamental para o desenvolvimento deste projeto. As Tabelas de 14 a 16 mostram os resultados de 5 ciclos de 100000 medições utilizando a biblioteca *timeit* para os três tipos de segmentação trabalhados.

Tabela 14 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de  $200 \times 200$  pixels.

Tarefa	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Média (ms)
Filtro de mediana (m=7)	1,47	1,36	1,38	1,37	1,28	1,37
Filtro bilateral (d=10)	1,42	1,27	1,38	1,32	1,36	1,35
Correção Gama ( $\gamma = 1.5$ )	1,81	1,84	1,94	1,95	2,45	1,99
Gradiente	2,41	1,99	2,92	2,68	2,90	2,58
Limiarização	2,47	2,23	2,21	2,2	2,19	2,26
Watershed Tradicional	171,54	125,52	163,32	163,56	165,59	157,91
Watershed - componentes conectados	60,85	61,6	60,98	60,02	65,42	61,77
Watershed em código de cadeia	68,68	64,68	63,16	69,54	69,89	67,19
Estimação + Filtro de Kalman	62,79	75,32	65,78	64,25	64,03	66,43

Tabela 15 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de  $150 \times 150$  pixels.

Tarefa	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Média (ms)
Filtro de mediana (m=7)	1,29	1,32	1,32	1,34	1,28	1,31
Filtro bilateral (d=10)	1,32	1,38	1,3	1,29	1,31	1,32
Correção Gama ( $\gamma = 1.5$ )	1,62	1,71	1,74	1,69	1,77	1,7
Gradiente	2,13	2	2,21	2,07	2,12	2,11
Limiarização	2,23	2,19	2,22	2,14	2,16	2,18
Watershed Tradicional	104,48	105,25	103,3	104,86	105,69	104,72
Watershed - componentes conectados	42,53	43,37	42,88	42,85	44,06	43,14
Watershed em código de cadeia	47,42	47,95	46,99	47,83	48,21	47,68
Estimação + Filtro de Kalman	44,88	44,92	45,06	44,79	45,19	44,97

Tabela 16 – Tempos de execução médios das operações na segmentação via Watershed tradicional de phantoms de gelatina opaca para regiões de interesse de  $100 \times 100$  pixels.

Tarefa	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5	Média (ms)
Filtro de mediana (m=7)	1,15	1,09	1,07	1,12	1,1	1,11
Filtro bilateral (d=10)	1,17	1,14	1,2	1,11	1,08	1,10
Correção Gama ( $\gamma = 1.5$ )	1,43	1,49	1,46	1,51	1,48	1,47
Gradiente	1,26	1,25	1,43	1,24	1,27	1,29
Limiarização	1,25	1,26	1,23	1,23	1,23	1,24
Watershed tradicional	74,55	75,65	75,34	74,91	74,90	75,07
Watershed - componentes conectados	37,83	38,11	37,9	37,9	38,22	37,99
Watershed em código de cadeia	42,24	41,96	43,85	44,4	42,58	43
Estimação + Filtro de Kalman	30,15	28,97	29,83	30,09	29,41	29,69

Fonte: Produzido pelo autor.

A partir das Tabelas acima, algumas conclusões foram tiradas. Primeiramente, como era de se esperar, os tempos de execução diminuem com a redução da região de interesse. No entanto, o erro é maior com o menor tamanho de ROI disponível. Acredita-se que isso ocorre devido à conversão de coordenadas, mas também deve-se considerar que o ruído de uma imagem de ultrassom é demasiadamente elevado e isso prejudica a segmentação e o processo como um todo. Deve-se estudar alternativas para uma redução ainda maior de ruído. Por hora, nota-se que o filtro bilateral apresenta um melhor resultado na suavização que o filtro de mediana, com maior preservação das bordas. Dado que seus tempos de execução são similares, opta-se pelo filtro bilateral.

Em (HAN SUN JINGYU YANG, 2005), conclui-se que o *Watershed* baseado em código de cadeia é mais eficiente que o baseado em componentes conectados. Isso não ocorre aqui, tanto em relação ao erro quanto ao custo computacional. Acredita-se que o responsável por isso também é o ruído da imagem de ultrassom, dado que o *Watershed* baseado em código de cadeia primeiro separa a imagem em mais de mil regiões e depois as mescla com base em sua similaridade. É evidente que a presença do ruído interfere nesse processo e, portanto, no bom funcionamento do método de segmentação como um todo. Escolhe-se, portanto, o método de *Watershed* baseado em componentes conectados como método de segmentação. Compara-se, então, o tempo de execução anterior a este trabalho com o tempo de execução obtido com as alterações nas Tabelas 17 e 18.



Tabela 17 – Tabela comparativa do tempo de execução do algoritmo antes das modificações e após as mesmas para nova ROI de  $150 \times 150$  pixels.

Operação	Algoritmo antigo	Nova proposta
Aquisição	14,66	14,66
Definição da ROI	1,32	1,13
Suavização	1,37	1,32
Realce	1,99	1,7
Operação morfológica	2,58	2,11
Segmentação	157,91	43,14
Estimação + Filtro de Kalman	66,43	44,97
Tempo total	246,43	109,03

Fonte: Produzido pelo autor.

Tabela 18 – Tabela comparativa do tempo de execução do algoritmo antes das modificações e após as mesmas para nova ROI de  $100 \times 100$  pixels.

Operação	Algoritmo antigo	Nova proposta
Aquisição	14,66	14,66
Definição da ROI	1,32	0,89
Suavização	1,37	1,1
Realce	1,99	1,47
Operação morfológica	2,58	1,29
Segmentação	157,91	37,99
Estimação + Filtro de Kalman	66,43	29,69
Tempo total	246,43	87,09

Fonte: Produzido pelo autor.

Com base nessas escolhas, altera-se mais uma vez as Figuras 17 e 20, resultando do algoritmo de segmentação resumido na Figura 32.

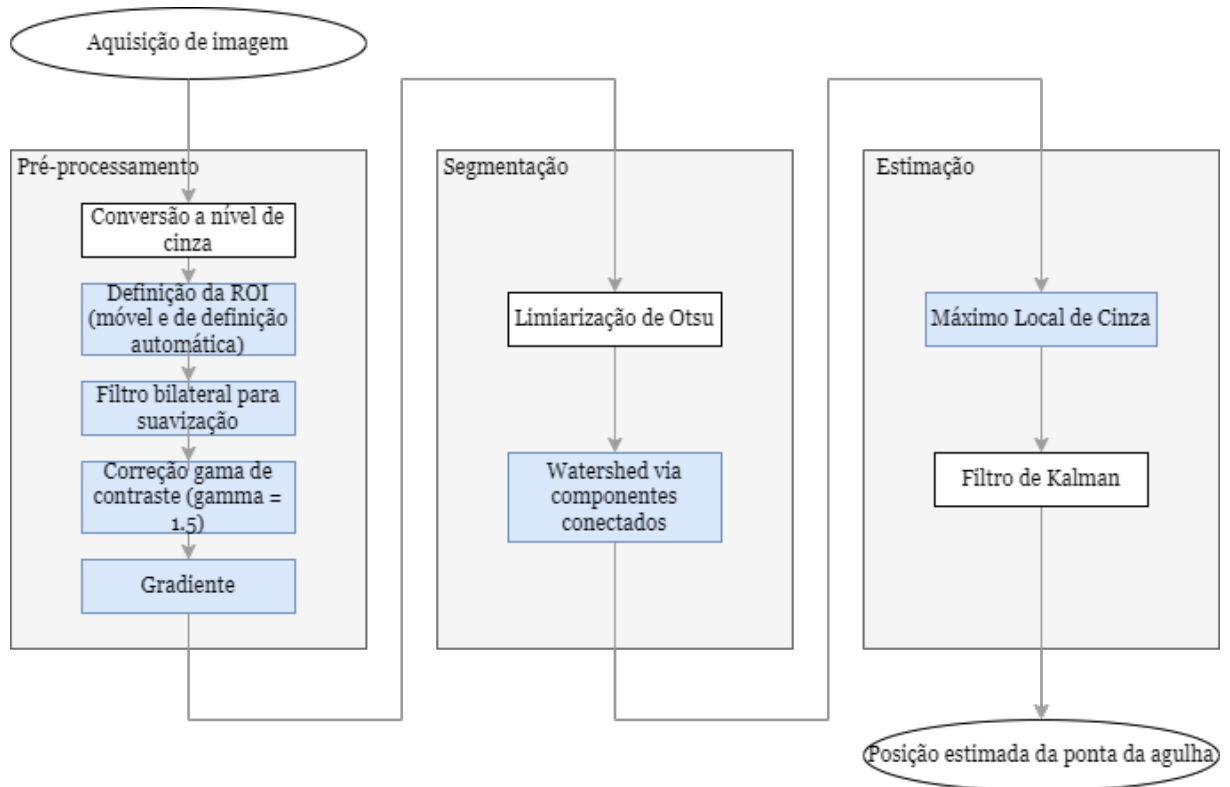


Figura 32 – Resumo do processo final de estimação da ponta da agulha.

Fonte: Produzido pelo autor.

Com este novo algoritmo, simulações de aquisição online foram feitas por meio de câmeras e do celular de um computador, considerando o tempo de aquisição de imagem e associando um tempo de transmissão ao processamento. Foram simulações bem sucedidas que indicam que as soluções aqui propostas devem se aplicar ao sistema presente no LARA.

## 5 Conclusões

O primeiro ponto a se ressaltar é a importância do tema aqui proposto. Como já citado, a adoção de procedimentos percutâneos apresenta inúmeras vantagens ao paciente, como menores riscos cirúrgicos, tempos de internação, recuperação e cicatrizes remanescentes. Aliada ao ultrassom, mantém-se um recurso extremamente vantajoso no meio médico, dado o baixo custo, facilidade de acesso já que se trata de uma tecnologia extremamente difundida e a ausência de riscos ao paciente. Nesse cenário, resalta-se a necessidade de precisão do sistema, pois lida-se com vidas.

Destrinchou-se, aqui, um sistema de imageamento via ultrassom direcionado a aplicações em robótica cirúrgica, visando a estimação da posição da ponta de uma agulha flexível durante sua movimentação em um simulador de tecido biológico. Foram analisadas as técnicas de pré-processamento utilizadas, alterações foram feitas e novos métodos de segmentação foram propostos.

No tocante ao pré processamento, passa-se a desprezar as imagens adquiridas com *phantoms* menos capazes de simular a dispersão de tecidos moles humanos e, por consequência, menos capazes também de simular ruídos. Analisando as imagens provenientes da movimentação em um simulador opaco, observa-se que o filtro bilateral apresenta maior suavização de ruídos com menor comprometimento das bordas, o que o torna uma alternativa melhor ao filtro de mediana, considerando, ainda, seus tempos de execução similares.

Quanto à segmentação, observa-se que o *Watershed* tradicional não é compatível com um processamento a tempo de execução. Esta hipótese já é considerada em (KIYOKU, 2015) e agora é confirmada. Duas alternativas são propostas à segmentação, ambas baseadas no algoritmo de *Watershed* que, apesar de lento, traz consigo uma boa segmentação da imagem. Numa delas, o *Watershed* é baseado em componentes conectados, utilizando uma analogia de chuva, assumindo que todos os componentes conectados que levam um gota ao mesmo vale formam uma bacia de captação associada a ele. O outro, fundamentado em código de cadeia, também se baseia em uma analogia de chuva, mas a simula para gerar componentes conectados usando código de cadeia ao invés do endereço de pixel. Ambos provaram ser eficientes na segmentação, mas deve-se considerar que o primeiro apresentou menor tempo de execução e menor erro de medição, sendo a melhor alternativa, portanto. Atesta-se, também, o efeito positivo da aplicação do Filtro de Kalman associado à estimação, considerando que, em todos os cenários, os valores de erro médio e máximo foram reduzidos, apesar de a estimativa de erro máximo no ensaio vertical ter sido considerada ainda elevada.

## 5.1 Perspectivas Futuras

Um ponto que se constata é que, ainda que a segmentação tenha ocorrido de modo satisfatório, o ruído em imagens geradas em *phantoms* de gelatina opaca ainda é alto. Ruídos desse tipo são do tipo *speckle* (ROSA, 2012), um ruído multiplicativo cuja suavização acaba por ser muito custosa. Apesar do filtro bilateral servir à segmentação, opções para melhorar a filtragem sem comprometer o custo computacional consistem em utilizar frames anteriores para filtrar frames seguintes, dado que o meio e o método de aquisição são constantes no processo e tem-se, portanto, um ruído padrão. Subtrações podem ser feitas de modo a minimizar esse tipo de ruído e tornar a segmentação mais rápida por reduzir o prejuízo do ruído ao processo.

Um ponto a se observar é o trabalho com o processamento de menos frames por segundo é bastante viável, mas que, ainda com ele, os tempos de execução das funções de segmentação e estimação ainda são elevados. Isso não é, necessariamente, um problema. A solução para isso, como trabalho futuro, seria a paralelização do processamento e a aplicação de requisitos de tempo real ao processamento das imagens de ultrassom. Dado que trata-se, aqui, de um processo cíclico, que se repete a cada frame, a estrutura de sistemas de tempo real necessária a garantir o deadline não é tão complexa. Um laço principal com tratamento de interrupções é uma boa sugestão.

Por problemas com o computador periférico e pelo fato de a maior parte deste trabalho ter sido feita à distância, dada a pandemia de *Covid-19*, tem-se como trabalho futuro importante a aplicação destes conceitos e implementações não apenas a simulações em computadores periféricos, mas no sistema de imageamento presente no laboratório. Simulações de rastreamento online foram feitas utilizando *webcam* e a câmera do celular, mas é fundamental que as implementações sejam aplicadas e ajustadas ao equipamento presente no LARA.

# Referências

- A BIENIEK, A. M. An efficient watershed algorithm based on connected components. **Pattern Recognition**, 1999. Citado na p. 46.
- ABOLHASSANI, N. Needle Insertion On Soft Tissues: A Survey. **Medical Engineering Physics**, v. 29, p. 413–431, 2007. Citado na p. 15.
- AGUIRRE, L. A. **Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3ª: Editora UFMG, 2007. Citado na p. 31.
- ANDRADE, W. T. Segmentação baseada em Textura e Watershed aplicada a imagens de Pólen. **Universidade Católica Dom Bosco**, 2011. Citado na p. 26.
- ARVELLOS, A. N. **Avaliação Tomográfica dos Parâmetros Anatômicos na Punção Renal Percutânea nos Decúbitos Ventral e Dorsal**. Mar. 2011. Diss. (Mestrado) – Universidade Federal de Juiz de Fora, Juiz de Fora. Citado na p. 13.
- BISHOP, G. W. An Introduction to the Kalman Filter. **University of North Carolina at Chapel Hill**, 2006. Citado nas pp. 31, 33.
- ETO, M.; NAITO, S. Robotic Surgery Assisted by the ZEUS System. **Department of Urology of the Graduate School of Medical Sciences from Kyushu University**, p. 39–48. Citado nas pp. 13, 14.
- GERALDES, A. A. Dispositivo Robótico Para Guiagem de Agulhas Flexíveis em Procedimentos Minimamente Invasivos. **Universidade de Brasília**, 2015. Citado nas pp. 15, 63.
- GONZALEZ, R. C. **Processamento Digital de Imagens**. 3. ed.: Editora Pearson, 2010. Citado nas pp. 17, 19, 21, 24–26, 45.
- HAN SUN JINGYU YANG, M. R. A fast watershed algorithm based on chain code and its application in image segmentation. **Pattern Recognition Letters**, 2005. Citado nas pp. 48, 71.
- HONG, J. An Ultrasound-driven needle-insertion robot for percutaneous cholecystostomy. **Phys. Med. Biol.**, p. 441-455, 2004. Citado na p. 14.
- KIYOKU, K. Rastreamento de Agulhas Flexíveis em Imagens de Ultrassom Para Uso Robótico de Inserção Automática. **Universidade de Brasília**, 2015. Citado nas pp. 15, 30, 31, 35, 37, 40, 43, 44, 50, 53, 62–64, 74.
- KREMKAU, F. W. Artifacts In Ultrasound Imaging. **J Ultrasound Med**, 1986. Citado na p. 29.
- LAING, F. C. The importance of ultrasonic side-lobe artifacts, 1982. Citado na p. 29.

- LASSO, A.; HEFFTER, T.; RANKIN, A.; PINTER, C.; UNGI, T.; FICHTINGER, G. PLUS: open-source toolkit for ultrasound-guided intervention systems. **IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING**, 2014. Citado na p. 38.
- LIMA, J. L. S. Reprodução e estudo de artefatos no ultrassom. **Revista Brasileira de Física Médica**, 2013. Citado na p. 31.
- MARESCAUX, J. Transatlantic Robotic Assisted Telesurgery. **Nature**, v. 413, 2001. Citado na p. 14.
- MATIAS, M. C. B. Robot-Assisted Steering of Flexible Needles for Percutaneous Procedures. **Universidade de Brasília**, 2012. Citado nas pp. 14, 15.
- MELLO, E. B. de. Restauração de Imagens Médicas Utilizando O Filtro de Kalman. **Universidade de São Paulo**, 1998. Citado na p. 32.
- MELLO, G. G. N. de; SOUZA PORTES MEIRELLES, G. de; OLIVEIRA, J. M. A. de; AJZEN, S.; D'IPPOLITO, G.; SZEJNFELD, J. Estudo Comparativo dos Procedimentos Percutâneos Orientados Por Métodos de Imagem No Tratamento das Coleções Hepáticas. **Radiologia Brasileira**, 2001. Citado na p. 14.
- NAROUZE, S. N. **Atlas of Ultrasound-Guided Procedures in Interventional Pain Management**. 2ª: Springer, 2011. Citado nas pp. 39, 40.
- NEEDLE Insertion On Soft Tissues: A Survey. **Medical Engineering Physics**, v. 29, p. 413–431, 2007. Citado na p. 15.
- ORTOPEDIA ESPECIALIZADA, N. de. **Cirurgias Minimamente Invasivas: Conceito e Aplicação na Ortopedia**. Ago. 2018. Disponível em: <<https://nucleodeortopedia.com.br/cirurgias-minimamente-invasivas-conceito-e-aplicacao-na-ortopedia/>>. Acesso em: 15 out. 2021. Citado na p. 13.
- PAPALEO, R. M. Ultrassonografia: princípios físicos e controle da qualidade. **Revista Brasileira de Física Médica**, 2019. Citado nas pp. 27, 28.
- PRATONDO, A. Integrating machine learning with region-based active contour models in medical image segmentation. **Elsevier**, 2016. Citado na p. 27.
- RANGEL, O.; TELLES, C. Tratamento da Dor Oncológica em Cuidados Paliativos. **Revista do Hospital Universitário Pedro Ernesto UERJ**, v. 1, n. 1, p. 5–6, 2012. Citado na p. 13.
- ROSA, R. G. S. Filtragem de Ruído Speckle em Imagens Clínicas de Ecografia. **Instituto Politécnico de Bragança**, 2012. Citado nas pp. 42, 43, 75.
- S. DOS REIS, V. B. e. Estudo de técnicas para rastreamento de agulhas flexíveis em imagens de ultrassom. **Universidade de Brasília**, 2019. Citado nas pp. 15, 50, 56.

- TOKUDA, J.; FISCHER, G. S.; PAPADEMETRIS, X.; YANIV, Z.; IBANEZ, L.; CHENG, P.; LIU, H.; BLEVINS, J.; ARATA, J.; GOLBY, A. J.; KAPUR, T.; PIEPER, S.; BURDETTE, E. C.; FICHTINGER, G.; TEMPANY, C. M.; HATA, N. OpenIGTLink: an open network protocol for image-guided therapy environment. **Int J Med Robot**, 2009. Citado na p. 37.
- ULTRASONIX, M. C. **SonixTouch Ultrasound System Extended User Manual**. 2013a. Citado nas pp. 35, 36.
- ULTRASONIX, M. C. **Transducers Product Especification Sheet**. 2013b. Citado na p. 36.
- VIEIRA, S. L. Desenvolvimento de um phantom para treinamento de biópsia de mama guiada por ultrassom. **Universidade de São Paulo**, 2005. Citado na p. 39.