

Thiago Nagaoka

Medição De Nível De Rios Baseada Em Análise De Vídeo

Brasil

2019

Thiago Nagaoka

Medição De Nível De Rios Baseada Em Análise De Vídeo

Trabalho de Conclusão do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - Campus Curitiba, apresentado como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Universidade Tecnológica Federal do Paraná

Departamento Acadêmico de Eletrônica

Departamento Acadêmico de Informática

Engenharia de Computação

Orientador: Prof. Dr. Bogdan Tomoyuki Nassu

Brasil

2019

Thiago Nagaoka

Medição De Nível De Rios Baseada Em Análise De Vídeo/ Thiago Nagaoka. –
Brasil, 2019-

57p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Bogdan Tomoyuki Nassu

Monografia (Graduação) – Universidade Tecnológica Federal do Paraná

Departamento Acadêmico de Eletrônica

Departamento Acadêmico de Informática

Engenharia de Computação, 2019.

1. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

Thiago Nagaoka

Medição De Nível De Rios Baseada Em Análise De Vídeo

Trabalho de Conclusão do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - Campus Curitiba, apresentado como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Trabalho aprovado. Brasil:

Prof. Dr. Bogdan Tomoyuki Nassu
Orientador

Professor
Convidado 1

Professor
Convidado 2

Brasil
2019

Agradecimentos

Ofereço a minha mais sincera gratidão:

a Marcelo Adriano Alves dos Santos, Nelson de Lima Ribeiro da Defesa Civil, e Esc. PC Rodrigo Taborda Ribas da Secretaria de Segurança Pública pela disposição a esclarecer as dúvidas sobre medição do nível de rios;

ao orientador, Prof. Dr. Bogdan Tomoyuki Nassu pela paciência, estímulo, correções e sugestões que foram essenciais para o desenvolvimento e a conclusão desse trabalho;

aos meus pais, Maria Emígdia da Cruz Rocha Nagaoka e Takeshi Nagaoka pela paciência, estímulo e apoio psicológico que foram essenciais para o desenvolvimento e conclusão desse trabalho.

Resumo

Alagamentos, enchentes e enxurradas são alguns desastres que uma grande cidade, como Curitiba, enfrenta. As autoridades de Curitiba são notificadas sobre algum desastre através dos Núcleos Comunitários de Defesa Civil, mas essa notificação é realizada apenas após o acontecimento do desastre. Um sistema automatizado de medição de nível de rios pode ser integrado a uma *smart city* para notificar as autoridades e a população sobre algum desastre que pode ocorrer em uma região da cidade. Neste trabalho é proposta uma nova abordagem para medição do nível de corpos d'água utilizando análise de vídeo. Os trabalhos anteriormente realizados sobre a automatização da medição de nível de corpos d'água utilizam algoritmos clássicos de processamento digital de imagens. Estes algoritmos clássicos são sensíveis a variações de iluminação local e perspectiva da câmera. O sistema proposto é composto por uma régua que possui faixas claras e escuras se alternando, e por uma boia. Foram testados dois métodos de medição, um método de verificação dos padrões claros e escuros e um método iterativo baseado na largura das faixas. O método da verificação do padrão atingiu um erro máximo de 2,5 centímetros. Já no método iterativo baseado nas larguras obteve erro máximo de 7,5 centímetros.

Palavras-chave: Medição de nível de rios, medição de nível de corpos d'água, processamento digital de imagens, *smart city*.

Abstract

Floodings, inundations and torrents are some environmental disasters that a large city, such as Curitiba, struggles with. Curitiba's authorities are notified of a disaster by the *Núcleos Comunitários de Defesa Civil*, but this notification is done only after a catastrophe happened. An automated system for water level measurement can be integrated to a smart city solution so it can notify the city's authorities and its population about some disaster that can happen suddenly in a city's region. In this work, we present an approach to measure water levels using video analysis. Previous work on automated water level measurement use classical digital image processing algorithms, which are sensitive to local illumination and camera perspective. The proposed system is composed by a ruler that has bright and dark stripes and by a floating base. We tested two measurement methods, one that verifies the bright and dark pattern, and another based on the width of the stripes. The pattern verification method reached a maximum error of 2,5 centimeters. Method based on the stripe width ended up at a maximum error of 7,5 centimeters.

Keywords: Water level measurement of rivers, digital image processing, smart city.

Lista de ilustrações

Figura 1 – Resultado esperado (Autoria própria).	13
Figura 2 – Arquitetura da solução proposta (Autoria própria).	14
Figura 3 – Processo de identificação da linha d’água: (a) Escala de cinza (b) Equalização de histograma (c) Binarização (d) Morfologia (e) Linha d’água (Lin et al., 2013).	16
Figura 4 – Processo para reconhecimento de caracteres: (a) Filtro da média (b) Filtro <i>bottom hat</i> (c) Binarização (d) Remove objetos pequenos e os não-caracteres (Lin et al., 2013).	17
Figura 5 – Componente <i>Hue - H</i> (Xu et al., 2018).	18
Figura 6 – Morfologia (Xu et al., 2018).	18
Figura 7 – Localização pelos componentes conectados (Xu et al., 2018).	19
Figura 8 – Precisão da detecção (Xu et al., 2018).	19
Figura 9 – Segregação dos componentes (Xu et al., 2018).	19
Figura 10 – Acumulação e identificação do pico (Sakaino, 2016).	20
Figura 11 – Arquitetura do sistema desenvolvido em (Hübner, 2013).	21
Figura 12 – Primeira etapa do sistema desenvolvido em (Hübner, 2013).	21
Figura 13 – Segunda etapa do sistema desenvolvido em (Hübner, 2013).	22
Figura 14 – Quarta etapa do sistema desenvolvido em (Hübner, 2013).	22
Figura 15 – Imagens do modelo e do quadro com os pontos de interesse representados em azul (Autoria própria).	25
Figura 16 – Exemplo de resultado do casamento de descritores dos pontos de interesse da imagem do modelo e do teste (Autoria própria).	26
Figura 17 – Planos com os pontos destacados em verde e o resultado da transformação planar desses pontos (Autoria própria).	27
Figura 18 – Imagens do padrão e da régua completa usadas no sistema proposto (Autoria própria).	28
Figura 19 – Imagens do modelo em <i>RGB</i> , escala de cinza e os pontos de interesse extraídos do modelo (Autoria própria).	29
Figura 20 – Imagens do quadro em <i>RGB</i> , escala de cinza e os pontos de interesse extraídos do quadro (Autoria própria).	30
Figura 21 – Imagens do quadro e do modelo e região do modelo com os pontos de interesse que tiveram correspondência entre si destacados com a cor verde (Autoria própria).	31
Figura 22 – Resultados da transformação plana e a remoção da região do modelo (Autoria própria).	32
Figura 23 – Ilustração dos elementos das equações 4.3 e 4.2 (Autoria própria). . .	34

Figura 24 – Gráfico da relação entre a medição real e a medição na imagem com dados providos do <i>software</i> projetado para associar pixels e centímetros (Autoria própria).	35
Figura 25 – Imagem contendo a régua completa convertida em escala de cinzas e rotacionada a 270° para melhor visualização (Autoria própria).	35
Figura 26 – Região selecionada destacada em branco da imagem da régua (Autoria própria).	35
Figura 27 – Região selecionada da figura 26 (Autoria própria).	36
Figura 28 – Gráfico do sinal extraído da imagem da régua horizontalmente posicionada (Autoria própria).	36
Figura 29 – Gráfico do sinal suavizado para redução de ruídos (Autoria própria). . .	36
Figura 30 – Gráfico da derivada do sinal suavizado (Autoria própria).	37
Figura 31 – Gráfico da figura 29 que ilustra a verificação dos padrões claros e escuros (Autoria própria).	40
Figura 32 – Ilustração da medição do nível d'água dado o valor convertido do índice (Autoria própria).	40
Figura 33 – Iterações do algoritmo 5 com os pontos da lista em azul e a reta da regressão linear em vermelho (Autoria própria).	44
Figura 34 – Imagens dos símbolos que formam a imagem do modelo presente na figura 34c (Autoria própria).	46
Figura 35 – Imagens que ilustram as medidas dos acessórios que a régua apresenta e a sua versão final (Autoria própria).	47
Figura 36 – Imagem exemplo do experimento realizado. (Autoria própria).	48
Figura 37 – Iterações de regressões lineares da pior medida dos blocos claros da primeira amostra (Autoria própria).	50
Figura 38 – Primeira iteração de regressões lineares da pior medida dos blocos escuros da primeira amostra: 0 <i>outliers</i> (Autoria própria).	51
Figura 39 – Locais da régua que o algoritmo computou ser os locais onde o nível d'água se encontra da primeira amostra (Autoria própria).	51
Figura 40 – Iterações de regressões lineares da pior medida dos blocos claros da segunda amostra (Autoria própria).	52
Figura 41 – Iterações de regressões lineares da pior medida dos blocos escuros da segunda amostra (Autoria própria).	52
Figura 42 – Locais da régua que o algoritmo computou ser os locais onde o nível d'água se encontra da segunda amostra (Autoria própria).	53

Lista de tabelas

Tabela 1 – Tabela de constantes do experimento no corpo d’água de 25 centímetros (Autoria própria)	48
Tabela 2 – Tabela das medições obtidas do primeiro experimento, valores em centímetros(Autoria própria).	48
Tabela 3 – Tabela dos erros obtidos do primeiro experimento, valores em centímetros(Autoria própria).	49

Sumário

1	INTRODUÇÃO	12
1.1	Organização do trabalho	14
2	TRABALHOS RELACIONADOS	16
2.1	Applications of Image Recognition for Real-Time Water Level and Surface Velocity	16
2.2	Water Level Estimation Based on Image of Staff Gauge in Smart City	18
2.3	Camera-Vision-Based Water Level Estimation	19
2.4	Medição automática de nível de água baseado em imagens: Caso do rio Itajaí-Açú	20
3	DEFINIÇÕES GERAIS	24
3.1	Imagen e imagem digital	24
3.2	<i>Scale-Invariant Feature Transform - SIFT</i>	24
3.3	Casamento de descritores	25
3.4	Matriz de homografia	26
3.5	Random Sample Consensus - <i>RANSAC</i>	26
4	MEDIÇÃO DO NÍVEL D'ÁGUA	28
4.1	Descrição da régua	28
4.2	Detecção da régua	29
4.3	Análise do erro da distância real e da distância da imagem da régua extraída da detecção	31
4.4	Medição do nível	33
4.4.1	Obtenção dos dados	33
4.4.2	Método da verificação do padrão	38
4.4.3	Método iterativo baseado na largura das faixas	40
5	EXPERIMENTOS E RESULTADOS	45
5.1	Descrição da régua	45
5.2	Descrição do protótipo do sistema proposto	46
5.3	Descrição do experimento	47
5.4	Resultados obtidos	47
6	CONCLUSÃO	54

1 Introdução

Cerca de 82% dos desastres naturais no Paraná em 2017 foram causados por incidências de chuvas intensas, segundo registros no Sistema de Defesa Civil. O alagamento e as enxurradas, segundo a Coordenadoria Estadual de Proteção e Defesa Civil, estão entre os 10 desastres de maior prevalência no Paraná em 2017. A cidade de Curitiba registrou o maior número de desastres naturais do estado, cerca de 13 registros. Estes desastres causam vários danos humanos, sendo que em números os principais são: 2775 pessoas desalojadas, 442 desabrigadas, 69 feridas, e 20 mortes. No pilar econômico os desastres também causam prejuízos. Por exemplo, o prejuízo no ano de 2017 foi de R\$ 158.955.910,37. Deste valor, R\$ 60.996.123,17 foram no setor público, enquanto R\$ 97.959.787,20 foi no setor privado ([Coordenadoria Estadual de Proteção e Defesa Civil, 2018](#)).

Uma forma das autoridades do estado receberem informações sobre desastres ambientais é através dos Núcleos Comunitários de Defesa Civil. Estes núcleos integram as comunidades nas ações de segurança social e preservação ambiental em áreas de risco. Entre os objetivos desta organização está a prevenção de desastres e a preparação das comunidades para o prestamento de assistência em desastres e acidentes. Os núcleos comunitários são formados por um grupo de pessoas organizado em uma comunidade localizada em uma área de risco, seja ela um distrito, bairro, rua, edifício, entre outros, que se envolve com as atividades de defesa civil como voluntário. Entre as atividades está a educação preventiva, organização e execução de campanhas, manter contato com a Coordenadoria Municipal de Defesa Civil, e assistência a esta na execução de ações de defesa civil ([Coordenadoria Estadual de Proteção e Defesa Civil, 2019](#)).

O problema da obtenção de informações sobre enchentes pode se beneficiar do conceito de *smart city*. *Smart cities* são cidades que combinam as estruturas físicas e tecnológicas para se tornarem eficientes e interconectadas ([Chourabi et al., 2012](#)). Este conceito é usado, por exemplo, na França. O *OSIRIS* é um sistema usado em várias cidades da França, o qual gera relatórios alertando sobre congestionamentos causados pelo transbordamento de rios, quais fábricas são afetadas, e gera plano de ações que devem ser tomadas. Neste plano de ações é considerado o impacto local da inundação. Os dados usados pelo sistema francês são obtidos da base de dados do Instituto Geográfico Nacional Francês ([Morel; Hissel, 2019](#)).

O objetivo principal deste trabalho é apresentar um novo sistema automatizado para medição do nível de corpos d'água (rios e lagos) baseado na análise de vídeo. Este sistema pode ser integrado a um projeto de *smart city*. O sistema é composto por uma câmera, uma régua de medição especialmente projetada, uma boia e um computador

onde as imagens são processadas. Foi desenvolvido um protótipo para prova de conceito do sistema. Neste sistema são analisados três segundos de um vídeo e, a partir dessa análise, são apresentadas medidas do nível d'água. A figura 1 mostra um exemplo da saída pretendida para o problema.



Figura 1 – Resultado esperado (Autoria própria).

O sistema desenvolvido segue a arquitetura ilustrada na figura 2. A régua é detectada utilizando como referência uma imagem de modelo, localizada no topo da régua. Para isso, foi usada uma técnica baseada no casamento de descritores SIFT (*Scale-Invariant Feature Transform*) (Lowe, 2004). Após o casamento de descritores, é feita uma retificação de perspectiva planar, usando uma matriz de homografia computada pelo algoritmo RANSAC (Fischler; Bolles, 1981). Após esta retificação planar, as propriedades da régua são analisadas através de algoritmos desenvolvidos especialmente para o sistema proposto. Uma das análises é baseada na regularidade do padrão, ou seja, existe uma seqüência de faixas brancas e pretas, que se alternam. A profundidade do rio é identificada com base no ponto onde esta regularidade é cessada na régua. Outra análise é baseada na largura das faixas do padrão, que é constante; e mesmo sob efeito de perspectiva, varia de forma previsível. Neste caso, a profundidade do rio é identificada quando são encontradas faixas com larguras que desviam consideravelmente do esperado. Convertendo-se as medidas para centímetros, tem-se a medida do nível d'água. A boia é empregada para que a transparência e o reflexo gerados pela água não interfiram na localização da interrupção do padrão. A transparência ou reflexo geram prolongamento falso na parte submersa da régua. O sistema automatizado produziu, em média, 3,125 centímetros de diferença da medida real em experimento realizado com um protótipo em escala reduzida.

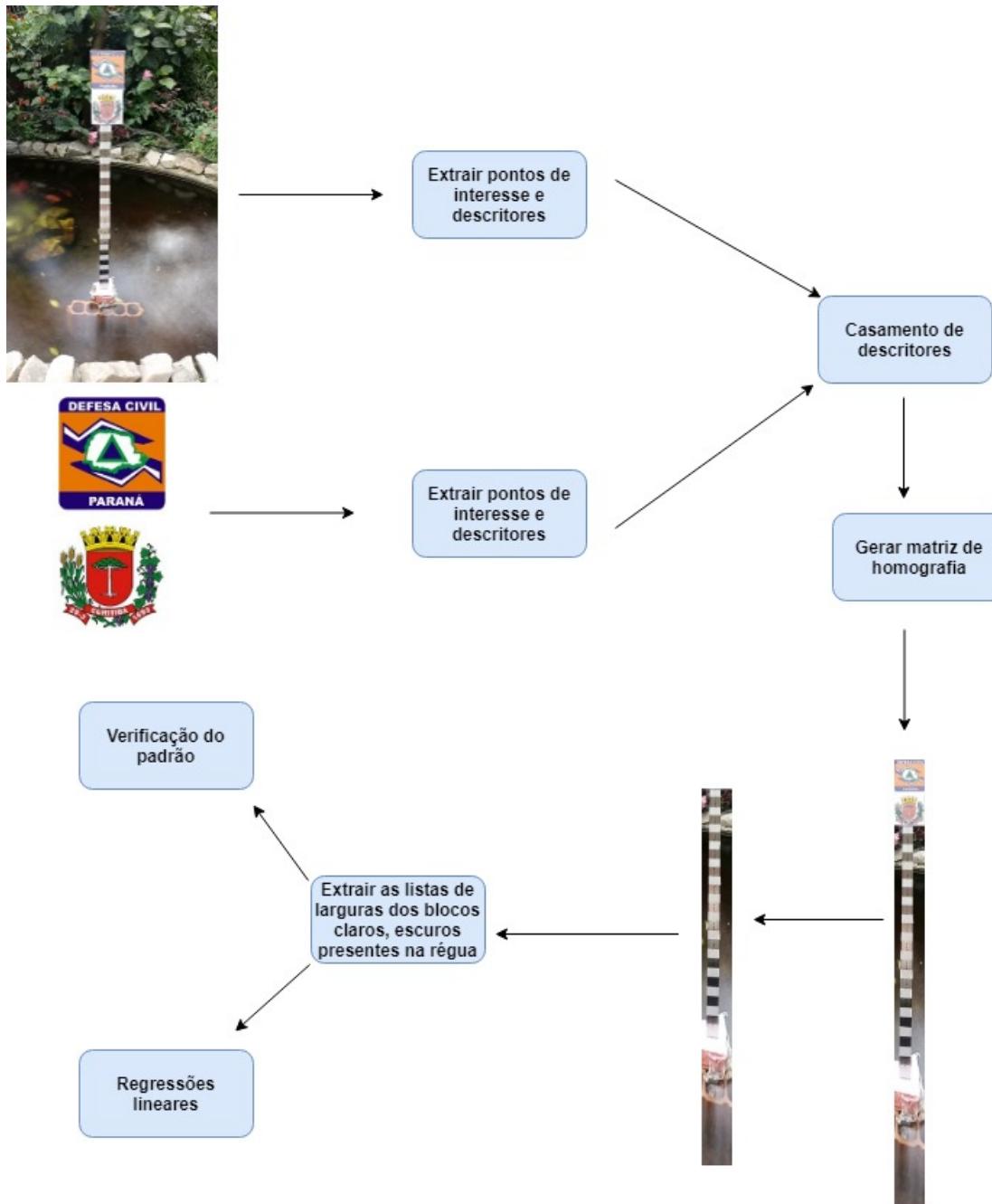


Figura 2 – Arquitetura da solução proposta (Autoria própria).

1.1 Organização do trabalho

A estrutura seguida neste trabalho é esta: no capítulo 2, trabalhos relacionados ao tema de medição de nível de rios serão descritos e comparados com o sistema proposto. No capítulo 3 será definida a nomenclatura de imagem que será usada até o final do trabalho, e serão descritos todos os algoritmos clássicos usados no trabalho. O capítulo 4 irá expor os procedimentos empregados para a medição do nível d'água, desde a detecção da régua, até o cálculo do nível do corpo d'água. Também será abordado o projeto da régua. O capítulo 5 irá descrever o experimento do sistema automatizado e mostrará os resultados

adquiridos da experimentação. O capítulo 6 apresentará as conclusões do trabalho.

2 Trabalhos relacionados

Neste capítulo serão apresentados trabalhos relacionados à medição automatizada de nível d'água. Na seção 2.1 será mostrado o sistema desenvolvido por (Lin et al., 2013). A seção 2.2 apresentará o sistema projetado por (Xu et al., 2018). A seção 2.3 apresenta um trabalho desenvolvido por (Sakaino, 2016). A seção 2.4 apresenta o trabalho desenvolvido por (Hübner, 2013).

2.1 Applications of Image Recognition for Real-Time Water Level and Surface Velocity

O sistema desenvolvido por (Lin et al., 2013) apresentou uma técnica de medição d'água onde é detectada a linha d'água e é feito o reconhecimento de caracteres.

Para que a detecção da linha d'água seja feita foram utilizadas técnicas clássicas de *processamento digital de imagens*, como mostra a figura 3. Primeiramente é feita a conversão para a escala de cinza e depois uma equalização do histograma. Em seguida é feita a binarização da imagem usando o método de Otsu e posteriormente é feita a utilização de técnicas de morfologia. Com isso, utilizando a detecção de bordas, é utilizada a borda inferior como a representação da linha d'água.

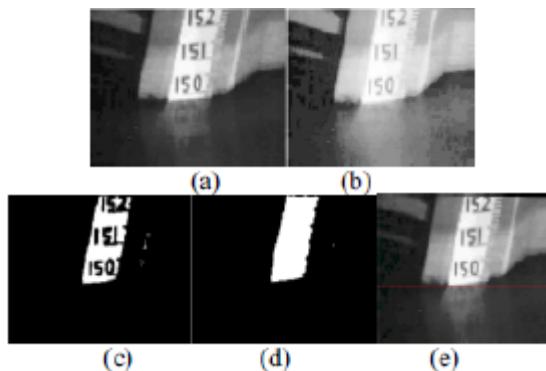


Figura 3 – Processo de identificação da linha d'água: (a) Escala de cinza (b) Equalização de histograma (c) Binarização (d) Morfologia (e) Linha d'água (Lin et al., 2013).

Após a detecção da linha d'água é feito o reconhecimento de caracteres, onde são usadas novamente as técnicas clássicas de *processamento digital de imagens*. O processo inicial está registrado na figura 4. Primeiramente é aplicado o filtro da média e o filtro *bottom-hat*. Em seguida é usada a binarização e a remoção de pequenos objetos e de não-caracteres da imagem. Com isso, para o reconhecimento de caracteres é usada a

técnica de *template matching*. Para efetuar a leitura do nível d'água é usado o método de interpolação linear.

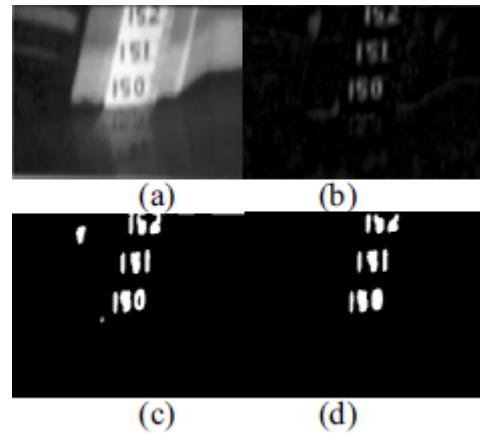


Figura 4 – Processo para reconhecimento de caracteres: (a) Filtro da média (b) Filtro *bottom hat* (c) Binarização (d) Remove objetos pequenos e os não-caracteres (Lin et al., 2013).

Apesar da medição estar de acordo com a medição real, a binarização é muito sensível a variações locais de iluminação, pois ela só é eficiente para imagens bimodais, onde há uma separação entre frente e fundo. A detecção da régua pode ser comprometida quando em torno da régua existirem vários objetos, com tonalidade mais clara, que não sejam a régua. A eficiência das técnicas de morfologia depende do tamanho em pixels do elemento estruturante, e além disso, a variação local da iluminação afeta muito o resultado da técnica usada. A metodologia utilizada especula que a água não seja transparente e que não possua reflexo. O reconhecimento de caracteres pode não ser eficiente quando existirem caracteres escritos fora da régua. É necessária uma câmera exclusivamente fixa apontada para a régua e é necessário que esta seja de alta resolução para identificar os caracteres.

No sistema proposto é utilizada uma boia para contornar os problemas do reflexo e da transparência na água. São utilizados algoritmos que contornam o problema da iluminação e da escala do objeto para a detecção de objetos. Não são utilizados os caracteres da régua, é utilizada a regularidade de padrão. A abordagem do sistema proposto não necessita que a câmera seja fixa, pois os algoritmos utilizados identificam a régua seja qual for a região da imagem. Além do mais, a identificação da irregularidade do padrão não é deteriorada quando uma imagem é de baixa resolução.

2.2 Water Level Estimation Based on Image of Staff Gauge in Smart City

O sistema de medição de nível de corpo d'água apresentado em (Xu et al., 2018) também demonstra duas etapas para medição, sendo a primeira a detecção da régua, e em seguida o reconhecimento de caracteres.

Para a detecção da régua são usadas também as técnicas clássicas de *processamento digital de imagens*. Primeiramente converte-se o espaço de cores de *RGB* para *HSV* e em seguida é selecionado o componente *H*, que codifica o matiz da cor, pois esse componente separa efetivamente o fundo e os objetos, como ilustrado na figura 5. Com o componente *H* é aplicada a binarização utilizando o método de Otsu e os métodos de morfologia: erosão e dilatação. Os dois métodos de morfologia foram aplicados para que a régua fosse destacada, como mostra a figura 6. A região da régua é localizada usando a extração dos componentes conectados, onde é usada o maior componente, resultado mostrado na figura 7. Para melhor visualização dos caracteres e eliminar reflexo d'água, foi usado um outro mapeamento, chamado de mapeamento-C e nele é aplicada a remoção dos componentes menores pela extração dos componentes conectados. O resultado desses passos está presente na figura 8.

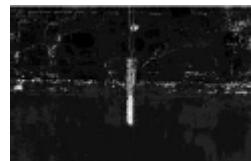


Figura 5 – Componente *Hue* - *H* (Xu et al., 2018).

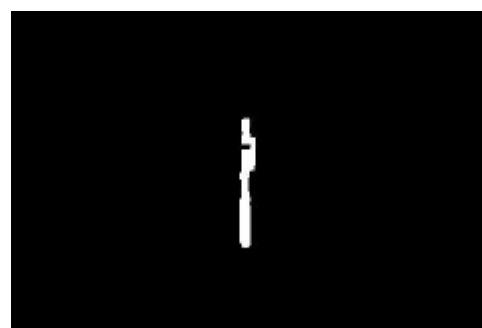


Figura 6 – Morfologia (Xu et al., 2018).

Para o reconhecimento de caracteres é usado um *CNN*, uma rede neural convolucional. Antes de sua utilização é feita uma segregação dos componentes da parte esquerda da figura 8 (d), o resultado dessa segregação está presente na figura 9.

Apesar da medição ser precisa com desvio padrão relativo de 3.09%, e a câmera não precisar ser fixa, este método ainda possui sensibilidade à intensidade local da iluminação e



Figura 7 – Localização pelos componentes conectados (Xu et al., 2018).

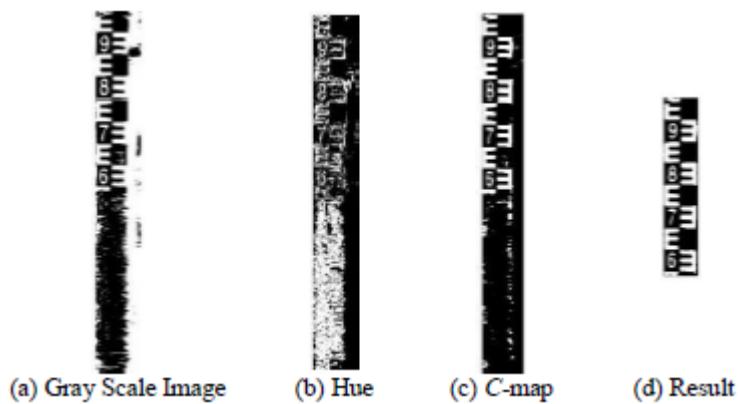


Figura 8 – Precisão da detecção (Xu et al., 2018).

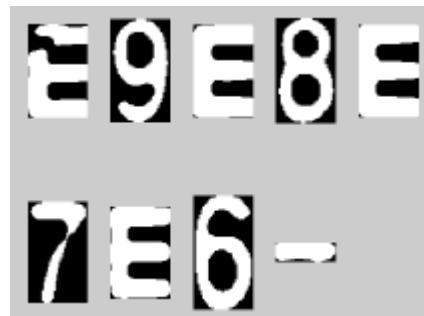


Figura 9 – Segregação dos componentes (Xu et al., 2018).

ao tamanho do elemento estruturante das técnicas de morfologia, o que afeta a eficiência na leitura. Se existirem outros elementos maiores que a régua na imagem, o sistema reconhece que esse outro objeto é a régua, efetuando leituras equivocadas. A câmera necessita ser de alta resolução para melhor visualização dos caracteres.

2.3 Camera-Vision-Based Water Level Estimation

O sistema (Sakaino, 2016) utiliza uma técnica clássica de *processamento digital de imagens*, histograma. O sistema é baseado em dois passos baseados em histograma, pois

a técnica assume que a superfície d'água e a margem do rio se encontram em diferentes picos nos histogramas.

Primeiramente é selecionada pelo usuário uma região onde se encontra uma parte da margem e do rio. Nessa região é aplicada a equalização de histograma para equalizar os picos locais. Depois é feita uma acumulação feita horizontalmente para que seja feita a análise no histograma. O local onde se encontra o maior pico é o local onde se encontra o nível d'água, como é visto na figura 10. A distância física entre a parte baixa e alta da margem do rio é usada para relacionar o local do pico com a medida real.

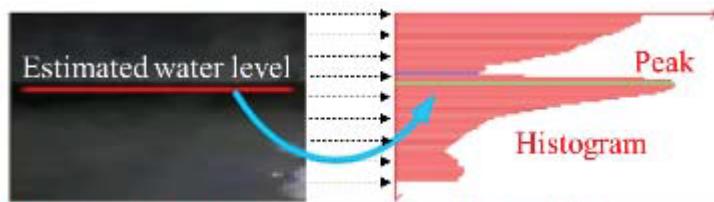


Figura 10 – Acumulação e identificação do pico ([Sakaino, 2016](#)).

Apesar da acurácia ser alta, ela depende das características da câmera, como o fator de *zoom-in*, resolução da imagem, taxa de amostragem. O sistema continua sendo sensível a iluminação local, o que provoca leituras equivocadas quanto ao nível d'água. Ela depende da seleção da região em que a margem do rio se encontra, ou seja, a detecção da linha d'água depende da indicação, feita pelo usuário, da região em que a margem do rio e a água se encontram. Se algum objeto percorrer a região que o usuário selecionou, a detecção do nível d'água pode falhar.

2.4 Medição automática de nível de água baseado em imagens: Caso do rio Itajaí-Açú

O sistema descrito em ([Hübner, 2013](#)) foi desenvolvido usando técnicas clássicas de processamento digital de imagens, como por exemplo morfologia matemática e filtragem. As etapas desenvolvidas pela autora estão ilustradas na figura 11.

A primeira etapa é chamada de simplificação e filtragem, pois nela ocorre primeiramente o redimensionamento para 800 x 600 pixels, como ocorre na figura 12 (a). Esta imagem redimensionada é convertida para escala de cinzas, ilustrada na figura 12 (b). Após a conversão, é computado o método de fechamento de morfologia matemática, que unifica as marcações escuras e os números da régua, e também elimina buracos. O resultado do fechamento está presente na figura 12 (c). Após isso, o método de abertura é aplicado para eliminar a régua. A ilustração da abertura está presente na figura 12 (d). A seguir, a subtração dos resultados do fechamento e da abertura é computada, destacando a régua, como ilustrado na figura 12 (e). Por fim, uma filtragem por área 5000 é realizada,

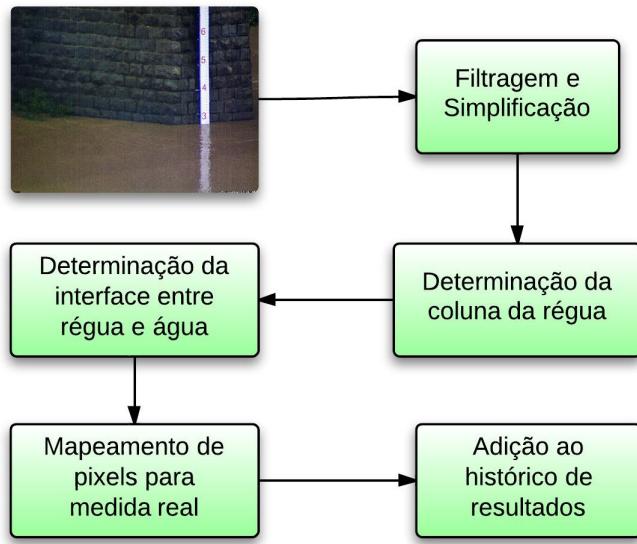


Figura 11 – Arquitetura do sistema desenvolvido em (Hübner, 2013).

eliminando os componentes de tamanho superior ao limitado, resultando somente na régua, como pode ser visto na figura 12 (f).

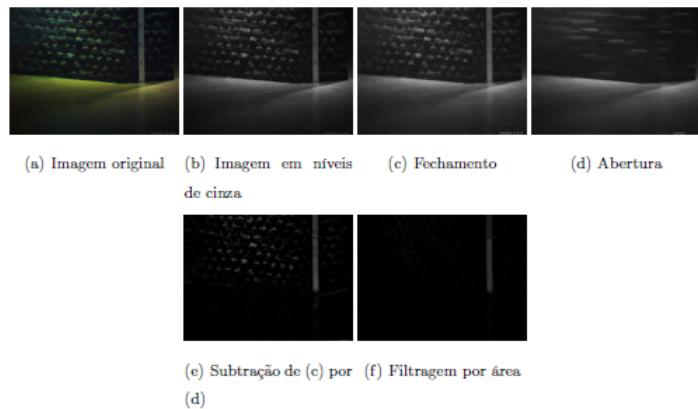


Figura 12 – Primeira etapa do sistema desenvolvido em (Hübner, 2013).

A segunda etapa é a busca da coluna da régua. Ela é realizada primeiramente através da soma das intensidades de cada coluna, para detectar as colunas mais claras, como ilustrado na figura 13 (b). Depois são eliminadas as colunas com intensidades menores que 50, para suavizar o sinal, como mostra a figura 13 (c). As figuras 13 (d) e 13 (e) ilustram o realce das posições onde a régua se localiza. Na figura 13 (f) é usado o máximo regional, para localizar a régua. Na figura 13 (g) é ilustrado o resultado da média das posições que correspondem à coluna central. A figura 13 (h) ilustra a localização da coluna na imagem.

A terceira etapa é a localização da interface entre a régua e a água. Como as intensidades da região da régua são superiores à região do rio, a localização da interface entre a régua e a água se dá com a posição da maior queda dos valores de intensidade.

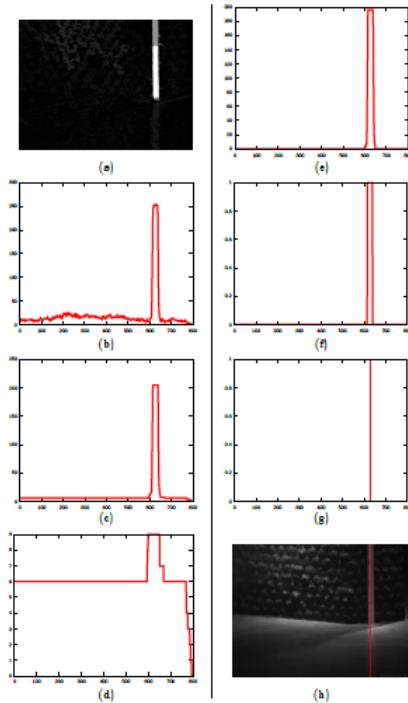


Figura 13 – Segunda etapa do sistema desenvolvido em (Hübner, 2013).

A quarta etapa é o mapeamento de pixels para a medida real. A figura 14 mostra a equação utilizada pela autora para fazer a conversão da posição obtida na terceira etapa para metros, onde a variável *pos* corresponde à posição da maior queda extraída na terceira etapa.

$$H \leftarrow 600$$

$$n \leftarrow (6, 80 - 0, 18) \frac{H - pos}{H} + 0.18$$

Figura 14 – Quarta etapa do sistema desenvolvido em (Hübner, 2013).

Apesar do erro de precisão ter sido de 5 centímetros para medições próximas ao limite, os erros de precisão foram maiores que 15 centímetros em outras medições. Isso ocorreu, pois a autora encontrou problemas de iluminação que provocam reflexos da régua na água, esse reflexo dificulta a localização da posição de maior queda da terceira etapa. Além disso, o algoritmo proposto por (Hübner, 2013) exige que a câmera seja fixa, como também a régua possuir intensidade bem clara. Também é preciso saber a largura em pixels da régua, pois é necessária para o elemento estruturante na aplicação da técnica de morfologia matemática *abertura*, precisando ser reconfigurada caso haja mudança na posição da régua.

O sistema proposto neste trabalho fornece uma solução que contorna o problema do reflexo na água, inserindo uma boia no sistema. A boia auxilia no bloqueio da parte da régua que se encontra inserido na água. Além disso, o sistema utiliza algoritmos que contornam o problema de iluminação na detecção de objetos. E por último, mas não menos

importante, o sistema desenvolvido não exige que a câmera seja fixa, ou seja, se a régua se encontrar em qualquer posição da imagem, em um ângulo qualquer, ela será detectada.

3 Definições gerais

Neste capítulo serão apresentados os principais conceitos de processamento de imagens para uma melhor compreensão dos capítulos seguintes. Também serão apresentados algoritmos clássicos usados neste trabalho.

3.1 Imagem e imagem digital

Uma imagem é uma função bidimensional que recebe dois parâmetros, x e y . As duas entradas dessa função são coordenadas espaciais de um plano. A saída da função é a intensidade ou o nível de cinza da imagem naquele determinado ponto. O ponto a que a função bidimensional se refere é o ponto (x, y) , esse ponto é chamado de *pixel* (Gonzalez; Woods, 2006).

A imagem digital é uma matriz 2-D que possui $M \times N$ de dimensão, onde M é o número de linhas e N é o número de colunas. O valor de cada elemento da matriz se chama intensidade ou nível de cinza (Gonzalez; Woods, 2006). A definição matemática de imagem usada nesse trabalho é a definida em 3.1.

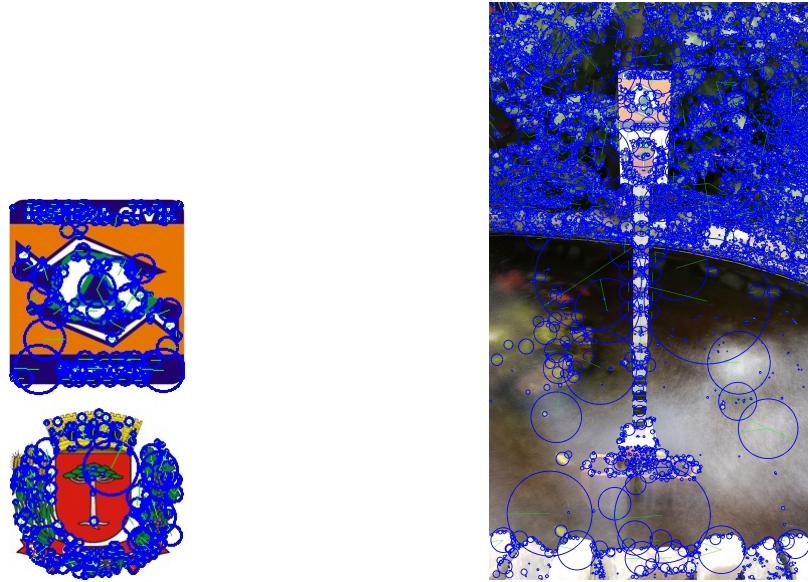
$$\mathbf{f}(x, y) \quad f : \mathbb{N}_0^2 \rightarrow \mathbb{N} \quad \text{onde} \quad 0 \leq \mathbf{f}(x, y) \leq 255, \quad 0 \leq x \leq M - 1, \quad 0 \leq y \leq N - 1 \quad (3.1)$$

3.2 *Scale-Invariant Feature Transform - SIFT*

Scale-Invariant Feature Transform é um algoritmo desenvolvido por (Lowe, 2004) que extrai descritores e pontos de interesse de uma imagem. Esses descritores e pontos de interesse são invariantes a escala e rotação da imagem, e parcialmente invariantes a iluminação e a perspectiva da câmera.

A entrada do algoritmo é uma imagem e a saída é uma lista de pontos de interesse. Cada ponto de interesse possui sua localidade, escala, orientação e um descritor. Cada descritor é formado por um vetor de 128 elementos, que representam a orientação e magnitude dos gradientes da região do ponto de interesse que aquele vetor representa (Lowe, 2004).

Com os pontos de interesse e descritores extraídos de imagens diferentes, estes podem ser comparados entre si. Com a propriedade da invariância, se um objeto estiver presente em várias imagens é esperado que existam descritores similares entre si. Isso faz com que o algoritmo *SIFT* detecte objetos em imagens com variações de iluminação, escala, rotação e, até certo ponto, de perspectiva.



(a) Imagem do modelo com os pontos de interesse representado em azul.
 (b) Imagem do quadro com os pontos de interesse representado em azul.

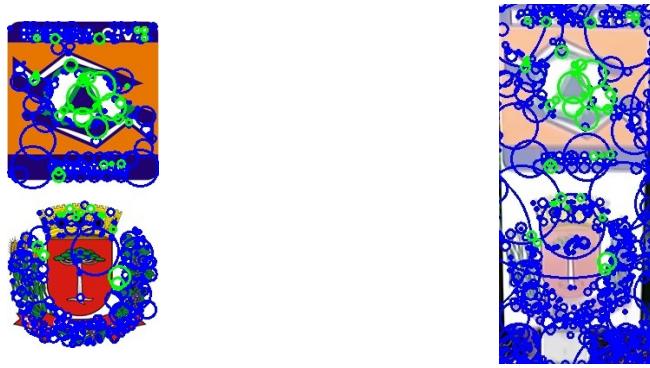
Figura 15 – Imagens do modelo e do quadro com os pontos de interesse representados em azul (Autoria própria).

As figuras 15a e 15b mostram os pontos de interesse extraídos a partir da aplicação do algoritmo *SIFT*.

3.3 Casamento de descritores

O algoritmo utilizado para o casamento de descritores é a força bruta. Cada descritor do primeiro conjunto é casado com todos os descritores do segundo conjunto, e a distância Euclidiana entre os descritores é estimada para todas as combinações dos elementos dos dois conjuntos. As distâncias dos casamentos de cada descritor são ordenadas, os dois melhores casamentos são retornados. Após isso, é calculada a razão entre a distância do primeiro melhor casamento e a distância do segundo melhor. Se o resultado dessa razão for pequeno, então o casamento é inambíguo, pois a distância do primeiro melhor é bem menor que a distância do segundo. Se o resultado da divisão for um valor alto, então o casamento é ambíguo, ou seja, as duas distâncias são parecidas, podendo o casamento ser descartado. Ele possui como entrada os descritores das duas imagens e como saída os dois melhores casamentos. ([OpenCV-Python Tutorials, 2019](#)).

Um exemplo do resultado da aplicação desse algoritmo está nas figuras 16a e 16b, onde os pontos de interesse similares às duas imagens obtiveram casamento.



(a) Pontos de interesse do modelo casa-
dos destacados com a cor verde. (b) Pontos de interesse do teste casados
destacados com a cor verde.

Figura 16 – Exemplo de resultado do casamento de descritores dos pontos de interesse da imagem do modelo e do teste (Autoria própria).

3.4 Matriz de homografia

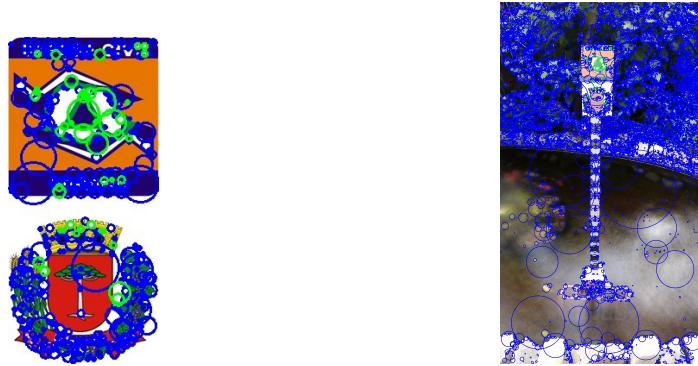
A matriz de homografia é uma matriz que mapeia a projeção de um plano para outro, ou seja, essa matriz projeta os pontos que estão em um plano para outro plano. Essa projeção é calculada através de multiplicação de matrizes usando coordenadas homogêneas (Bradski; Kaehler, 2008).

Uma vantagem de se utilizar a matriz de homografia para fazer a projeção é que os pontos não perdem a colinearidade, ou seja, se alguns pontos em um plano estão sobre uma linha, esses mesmo pontos estarão sobre a mesma linha no outro plano. A desvantagem que existe a esse cálculo de projeção é a atenuação do paralelismo.

Um exemplo da aplicação da matriz de homografia é apresentado nas figuras 17a, 17b e 17c. As figuras 17a e 17b mostram dois planos, cujos pontos utilizados para o mapeamento estão representados na cor verde. A figura 17c mostra o resultado final desse mapeamento.

3.5 Random Sample Consensus - *RANSAC*

O *RANSAC* é um algoritmo que possui como entrada duas listas de pontos que estão em dois planos, que são o plano origem e o plano destino, e como saída o algoritmo retorna uma matriz de homografia. Esse algoritmo seleciona diversas vezes alguns pontos que estão no plano de origem e computa várias matrizes de homografia. Por consenso, a matriz de homografia que melhor transforma os pontos do plano de origem para o plano de destino é selecionada (Fischler; Bolles, 1981).



(a) Imagem do modelo, exemplo de um plano, com os pontos destacados com a cor verde utilizados no cálculo com a matriz de homografia.

(b) Imagem do quadro, exemplo de um plano, com os pontos destacados com a cor verde utilizados no cálculo com a matriz de homografia.



(c) Resultado da operação com a matriz de homografia.

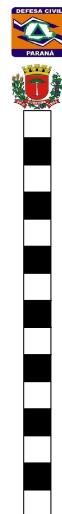
Figura 17 – Planos com os pontos destacados em verde e o resultado da transformação planar desses pontos (Autoria própria).

4 Medição do nível d'água

Neste capítulo será apresentado o algoritmo para a medição do nível de corpo d'água. Na seção 4.1 serão detalhadas as especificações de como a régua e a boia devem ser. Na seção 4.2 serão apresentados os procedimentos para a detecção da régua. Na seção 4.3 é apresentada uma análise da proporcionalidade do erro da distância real do topo da régua até um ponto qualquer dela com a mesma distância na imagem da régua. Também é apresentada uma relação entre a distância física e a distância na imagem da régua. Na seção 4.4 são apresentados os procedimentos para a medição do nível d'água.

4.1 Descrição da régua

A régua deve apresentar padrões claros e escuros, cada um alternando entre si e com altura constante. Deve-se utilizar também uma imagem simples com elementos com bom nível de contraste como modelo, como presente na figura 18a. Essa imagem deve ser fixa sobre uma superfície plana rígida. A rigidez deve ser o suficiente para que a imagem não oscile devido ao vento. Essa imagem colada numa superfície deve ser fixada na régua de modo que ambos estejam posicionados seqüencialmente, ou seja, a imagem posicionada no topo da régua. A figura 18b mostra uma possível representação da régua, onde a imagem está fixa no topo da régua, e os padrões claros e escuros bem definidos.



(a) Imagem que foi colocada no topo da régua.
(b) Imagem ilustrativa da régua usada no sistema proposto.

Figura 18 – Imagens do padrão e da régua completa usadas no sistema proposto (Autoria própria).

4.2 Detecção da régua

Primeiramente a imagem do modelo em *RGB*, como da figura 19a, deve ser convertida em escala de cinza, como mostra a figura 19b. Após isso é aplicado o algoritmo *SIFT* na imagem em escala de cinza do modelo, resultando na extração dos pontos de interesse e descritores, como explicado na seção 3.2. Ele é aplicado apenas uma vez na imagem do modelo antes da operação do sistema. Somente os pontos de interesse extraídos precisam estar armazenados, e estes podem ser visualizados em azul na figura 19c.



(a) Imagem em cores *RGB* do mo-

(b) Imagem em escala de cinza do
deulo.



(c) Imagem do modelo com os pontos de interesse em azul extraídos pelo algoritmo *SIFT*.

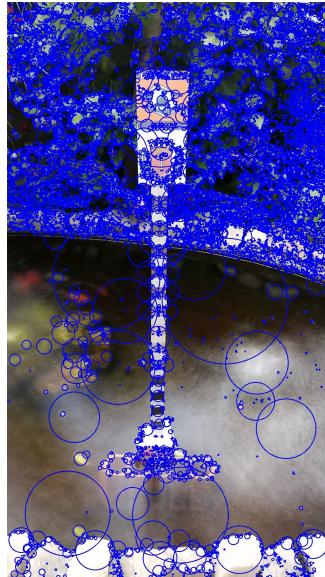
Figura 19 – Imagens do modelo em *RGB*, escala de cinza e os pontos de interesse extraídos do modelo (Autoria própria).

Em seguida são capturados M segundos de quadros de uma câmera e computada a média desses quadros capturados, como pode ser visto na figura 20a. Este cálculo da média dos quadros é computado para remoção de ruídos. Este quadro resultado da média dos M segundos é convertido em uma imagem em escala de cinza, como pode ser visto na figura 20b. Nesta imagem em escala de cinza é aplicada o algoritmo *SIFT*. Os pontos de interesse extraídos estão destacados em azul na figura 20c.

Posteriormente é executado o casamento de descritores por força bruta das imagens do modelo e do quadro. Os pontos de interesse cujos descritores obtiveram casamento



(a) Imagem em cores *RGB* do quadro. (b) Imagem em escala de cinza do quadro.

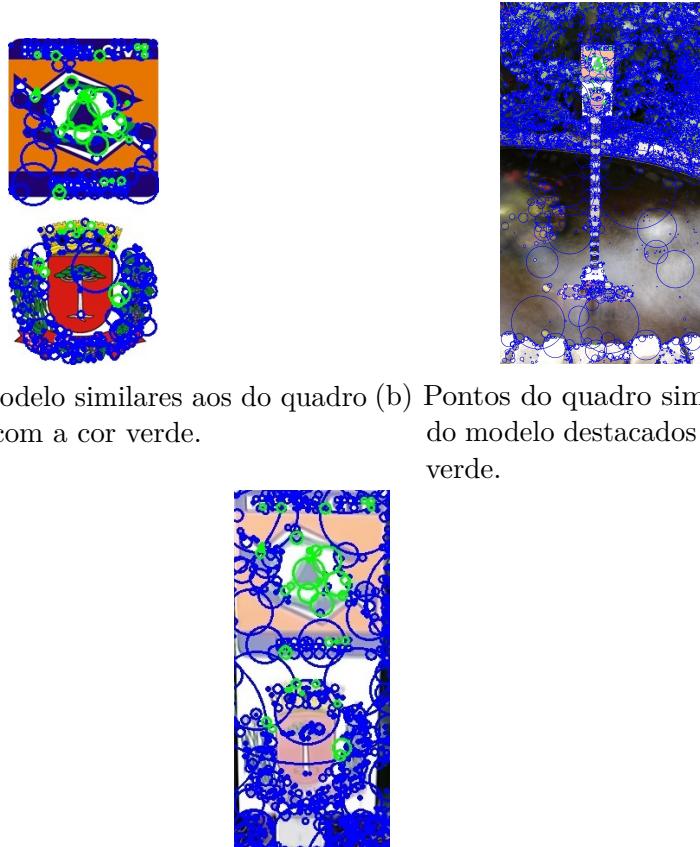


(c) Imagem do quadro com os pontos de interesse em azul extraídos pelo algoritmo *SIFT*.

Figura 20 – Imagens do quadro em *RGB*, escala de cinza e os pontos de interesse extraídos do quadro (Autoria própria).

estão destacados em verde nas figuras 21a, 21b e 21c. A figura 21c é uma figura da região em que os pontos de interesse do modelo são similares aos pontos de interesse no quadro.

Com os pontos de interesse similares é computada a matriz de homografia através do algoritmo *RANSAC*, explanado na seção 3.5, que efetua uma transformação entre planos, da imagem do modelo para a imagem do quadro. Após isso é computada a matriz inversa da homografia, pois se deseja efetuar a transformação planar da imagem do quadro para a imagem do modelo. Uma transformação planar usando a matriz inversa de homografia é então aplicada no quadro, cujo resultado está presente na figura 22a. Essa aplicação



- (a) Pontos do modelo similares aos do quadro destacados com a cor verde.
- (b) Pontos do quadro similares ao do modelo destacados com a cor verde.
- (c) Região da figura 21b onde os pontos tiveram correspondência com os pontos do modelo.

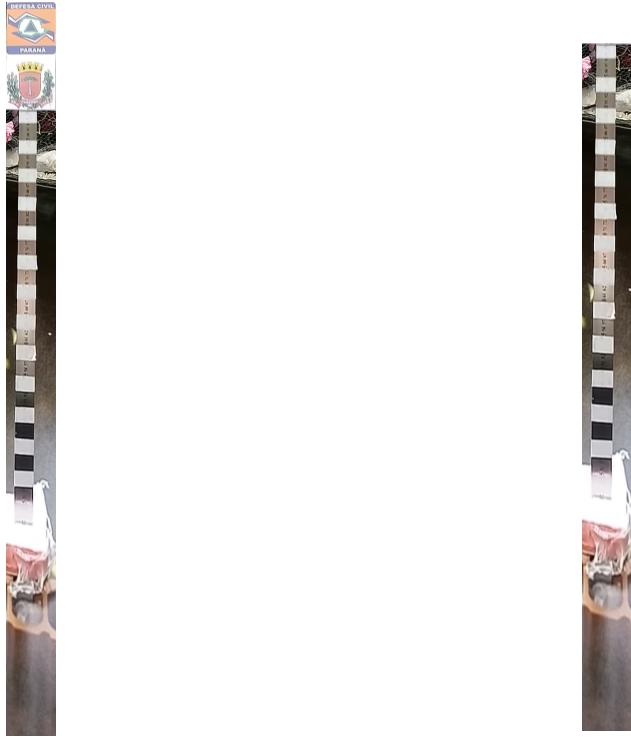
Figura 21 – Imagens do quadro e do modelo e região do modelo com os pontos de interesse que tiveram correspondência entre si destacados com a cor verde (Autoria própria).

é efetuada pois se deseja transformar a imagem do quadro em uma imagem da régua completa. E, por último, para a medição do nível d'água, a parte que a imagem do modelo se encontra na figura 22a não é relevante mais, então pode-se removê-la. O resultado dessa remoção está ilustrado na figura 22b.

4.3 Análise do erro da distância real e da distância da imagem da régua extraída da detecção

Foram realizados testes iniciais para analisar se a distância física do topo da régua até um ponto qualquer dela é diretamente proporcional à mesma distância na imagem da régua. Esta análise é realizada para que seja verificado o efeito dos erros que o cálculo da transformação dos pontos de um plano para outro causa na perspectiva dos blocos claros e escuros. Ou seja, se o efeito do erro para os pontos que estão afastados do topo da régua é grande.

A distância na imagem é medida através de um fator multiplicativo r pela equação



(a) Transformação planar usando a matriz inversa de homografia.
 (b) Remoção da região do modelo da figura 22a.

Figura 22 – Resultados da transformação plana e a remoção da região do modelo (Autoria própria).

4.1. Na equação 4.1, a variável tam_p representa a altura em pixels da imagem do modelo. A variável tam_cm representa a altura em centímetros da imagem do modelo. Para os testes foi utilizada uma régua em escala reduzida com 100 centímetros de comprimento da base até o topo. A altura da imagem do modelo usado nos testes é 379 pixels. A altura real do modelo é 17,8 centímetros. Substituindo estes valores na equação 4.1 resulta em 21,29 pixels/centímetros.

$$r = \frac{tam_p}{tam_cm} \quad (4.1)$$

$$z = 100 - y \quad (4.2)$$

Para que os testes fossem realizados, foi desenvolvido um pequeno programa de testes que extrai a régua da imagem. Este programa demanda ao usuário que indique onde cada medida real se localiza na imagem. Cada ponto demandado da imagem é convertido em centímetros dividindo seu valor por r da equação 4.1. Este valor em centímetros corresponde à porção y ilustrada na figura 23, que corresponde a um valor com medição tendo como referência o topo da régua. Aplica-se então a equação 4.2 obtendo um valor cuja medição tem como referência a base da régua, que condiz com a variável z da figura 23.

Com todos os pontos convertidos em centímetros e suas medidas reais, pode-se computar um gráfico que demonstre a relação entre a medição na imagem e a medição real. Este gráfico é mostrado na figura 24.

No gráfico da figura 24 observa-se que o limite máximo nas coordenadas x e y é de 100 centímetros, pois a imagem do modelo está posicionada na medição de 100 centímetros da régua. Com isso, observa-se que as medições maiores que 80 centímetros possuem erro pequeno cada vez que se aproxima do topo da régua. Este fenômeno acontece, pois a transformação planar calculada a partir do modelo no topo da régua possui imprecisões, com um erro que se acumula conforme se afasta da placa. Neste gráfico percebe-se também que o erro tem um comportamento linear, cuja correção pode ser realizada a partir de uma função linear, como da equação 4.3. Na equação 4.3 a variável z se refere à medição calculada. A variável w corresponde à medição real.

Com essa análise da equação 4.3 do erro, conclui que os seus parâmetros são sensíveis às mudanças de posição de cada padrão com relação à câmera, ou seja, se porventura a câmera for deslocada para perto ou para longe dos padrões, esse processo de testes terá de ser repetido. Essa equação é utilizada somente na localização da base da régua. Ela não é usada na etapa da medição pois a largura e a intensidade dos pixels de cada padrão são computadas independentemente se um padrão está maior ou menor que o adjacente.

$$w = 0,858 \times z + 13,07 \quad (4.3)$$

4.4 Medição do nível

Nesta seção serão descritos os dois métodos utilizados para a medição do nível de corpo d'água. Os dois métodos propostos para a medição do nível se baseiam em propriedades do padrão de faixas presentes na régua. Na subseção 4.4.1 será descrito como estas propriedades são computadas a partir de uma imagem da régua como da figura 22b. Na subseção 4.4.2 é descrito o método que verifica a continuidade do padrão, localiza a sua ruptura, que é onde se encontra o nível d'água e calcula o nível em centímetros a partir dessa localização. Na subseção 4.4.3 é descrito um método iterativo baseado na largura das faixas claras e escuras. Considerando a largura das faixas como fixa, localiza a faixa cuja largura é muito divergente das outras faixas, que é onde se encontra o nível d'água. Com essa localização é calculado o nível em centímetros.

4.4.1 Obtenção dos dados

Primeiramente a imagem que contém a régua completa é convertida em escala de cinzas, como ilustra a figura 25. A imagem da régua da figura 25 está representada

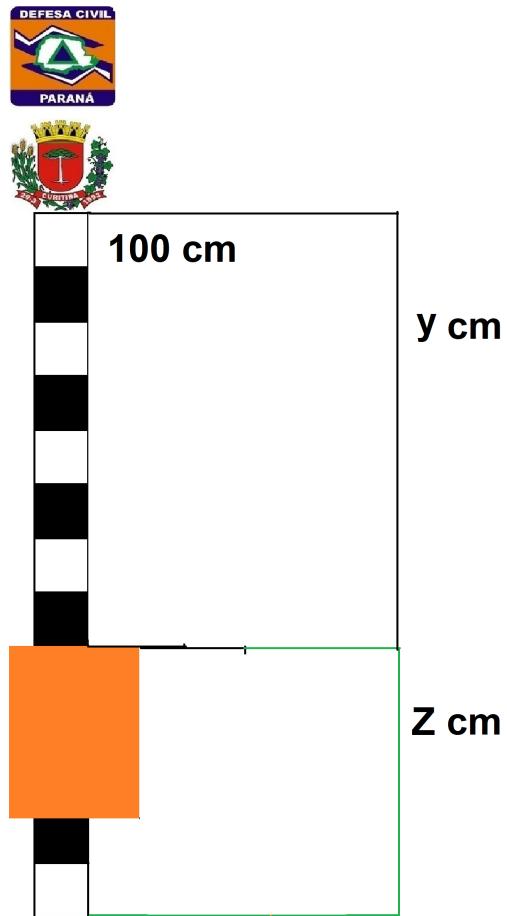


Figura 23 – Ilustração dos elementos das equações 4.3 e 4.2 (Autoria própria).

horizontalmente para melhor visualização do que será explicado posteriormente.

Em seguida é selecionada uma região que cobre horizontalmente toda a imagem da régua, centralizada verticalmente. Essa região representa 29% da altura de toda a imagem da régua, pois a altura da imagem da régua da figura 25 é de 175 pixels e a altura da região selecionada é de 51 pixels. Essa região está ilustrada na figura 26 como sendo a área do interior das duas linhas retas destacadas em branco. Como pode ser observado na figura 26, a região selecionada está no centro da imagem, mas não cobre exatamente o centro da régua. Isto ocorre por causa de imprecisões na estimativa descrita na seção 4.3 das medições da régua. A imagem da seleção efetivada está apresentada na figura 27.

Após a seleção, é computada a média das colunas da imagem da régua, se ela estiver posicionada horizontalmente na cena. Se ela estiver posicionada verticalmente, o cálculo da média deve ser das linhas da imagem da régua. Essa média é calculada para atenuar problemas causados pelas imprecisões na retificação da imagem da régua. A média atenua também os reflexos e a sujeira na superfície da mesma. As médias geram um sinal de uma variável, que está ilustrado na figura 28. Na figura 28 são percebidas as alternâncias entre os blocos claros e escuros e a ruptura desta intercalação.

É computada a suavização deste sinal para a redução de ruídos, usando o filtro

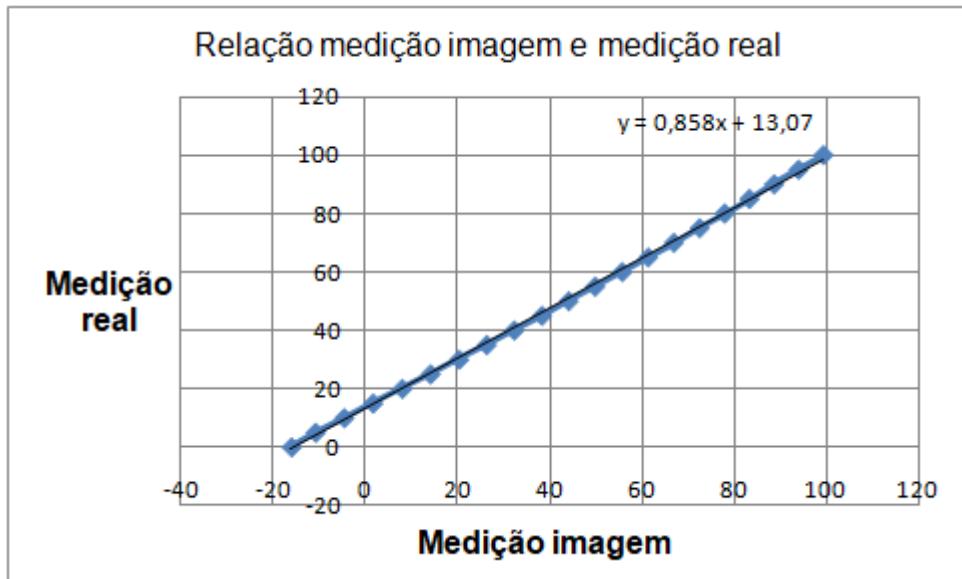


Figura 24 – Gráfico da relação entre a medição real e a medição na imagem com dados providos do *software* projetado para associar pixels e centímetros (Autoria própria).



Figura 25 – Imagem contendo a régua completa convertida em escala de cinzas e rotacionada a 270° para melhor visualização (Autoria própria).

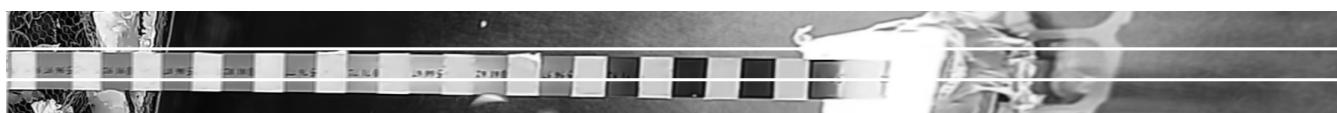


Figura 26 – Região selecionada destacada em branco da imagem da régua (Autoria própria).

da mediana com janela de 51 unidades de largura. O gráfico da suavização do sinal está representado na figura 29. Na figura 29 se percebe que houve uma atenuação nos ruídos. Este decremento nos ruídos não prejudica a existência da alternância dos blocos claros e escuros e o seu rompimento.

Após a suavização do sinal, é computada a derivada para a identificação dos locais onde ocorrem as mudanças de intensidade de níveis de cinza, onde são localizados os blocos claros e escuros. A figura 30 ilustra um exemplo de gráfico da derivada do sinal suavizado. Observando-se as figuras 29 e 30 se percebe que a derivada é positiva quando a mudança da tonalidade escura para a clara é drástica. Quando a mudança da tonalidade é oposta, percebe-se que a derivada é negativa.

Com a derivada computada pode-se calcular a largura de cada bloco em pixels. Para que seja obtida a largura em pixels de cada bloco, se deve localizar primeiramente



Figura 27 – Região selecionada da figura 26 (Autoria própria).

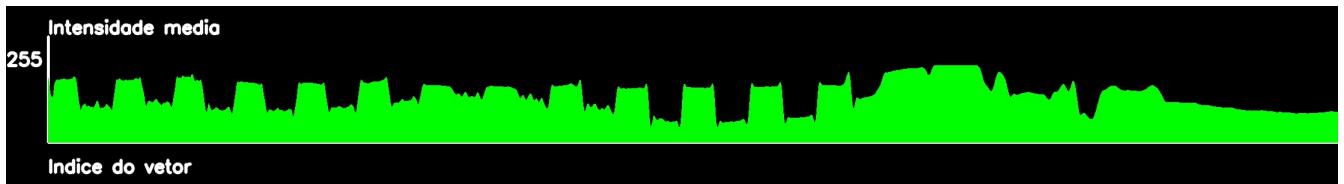


Figura 28 – Gráfico do sinal extraído da imagem da régua horizontalmente posicionada (Autoria própria).

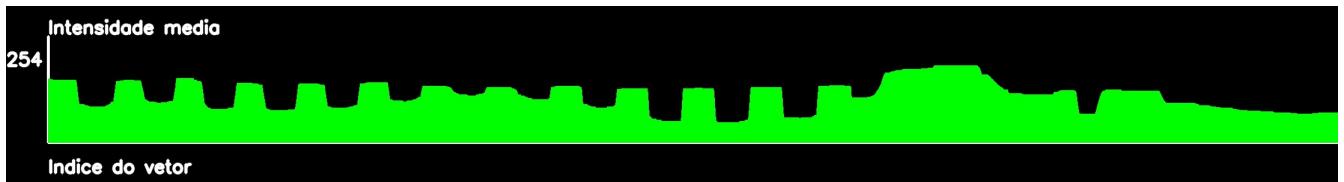


Figura 29 – Gráfico do sinal suavizado para redução de ruídos (Autoria própria).

a posição de cada bloco, que foram obtidos através do algoritmo 1. Na primeira linha do algoritmo está presente a posição atual em que o algoritmo irá começar a percorrer o vetor dx , esta variável é inicializada em 0 quando o primeiro bloco está localizado na primeira posição do vetor. Se a localização do primeiro bloco está na última posição do vetor, então esta variável deve ser preenchida com o valor da última posição do vetor. A variável que controla a indicação do sinal do valor da derivada se encontra na segunda linha do algoritmo, se o primeiro bloco é claro, então o valor dessa variável é *verdadeiro*, pois é desejado encontrar a localização da transição do bloco claro para o bloco escuro. Se o primeiro bloco é escuro, então o valor é *falso*, pois a próxima transição é do bloco escuro para o bloco claro. A terceira linha inicializa uma lista de posição com a posição do primeiro bloco. Com essas atribuições iniciais estabelecidas, o algoritmo percorre todo o vetor da derivada de bloco em bloco através do limite máximo de cada bloco, representado pela constante `MAX_BLOCK_WIDTH`. Este percurso é feito pelo *loop* presente na linha 4. `MIN_BLOCK_WIDTH` e `MAX_BLOCK_WIDTH` são constantes que representam o intervalo mínimo e máximo que será percorrido dentro um bloco claro ou escuro pelo *loop* presente na linha 7. A constante N é o valor da largura da imagem da figura 27 quando a régua está posicionada horizontalmente, ou da altura da imagem quando a régua está posicionada verticalmente. No corpo deste *loop* da linha 7 é procurada a posição da maior derivada de acordo com o sinal deste valor, pois é nessa posição que ocorre a transição de escuro para claro ou de claro para escuro. Quando a maior derivada é localizada a sua posição é então inserida na lista final das posições e a variável de controle do sinal do valor da derivada inverte o valor. Se o valor dessa variável era *verdadeiro*, então a maior derivada encontrada tinha sinal negativo, então o valor a ser atribuído agora é *falso* para que a maior derivada a ser encontrada em seguida seja positivo. Se o valor era *falso*, então

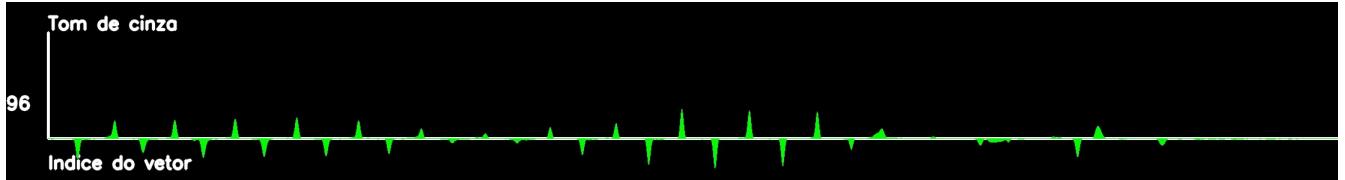


Figura 30 – Gráfico da derivada do sinal suavizado (Autoria própria).

o valor da maior derivada encontrada tinha sinal positivo e a próxima maior derivada tem sinal negativo, então a variável de controle é atribuída com valor *verdadeiro*.

Algoritmo 1: Algoritmo de localização das posições dos blocos.

Input : Vetor da derivada do sinal: dx
Output : Lista das posições de cada bloco: $lista_posicoes$

```

1 pos_atual  $\leftarrow 0$ 
2 negativo  $\leftarrow Verdadeiro$ 
3 lista_posicoes  $\leftarrow [pos\_atual]$ 
4 while pos_atual + MAX_BLOCK_WIDTH  $< N$  do
5   valor_maior  $\leftarrow |dx[pos\_atual + MIN\_BLOCK\_WIDTH]|$ 
6   pos_maior  $\leftarrow pos\_atual + MIN\_BLOCK\_WIDTH$ 
7   for i  $\leftarrow pos\_maior + 1$  to pos_atual + MAX_BLOCK_WIDTH do
8     if ((negativo e  $dx[i] < 0$ ) ou (não negativo e  $dx[i] > 0$ )) e
9       ( $|dx[i]| > valor\_maior$ ) then
10         valor_maior  $\leftarrow |dx[i]|$ 
11         pos_maior  $\leftarrow i$ 
12     end
13   end
14   lista_posicoes  $\leftarrow$  insere valor de pos_maior na lista
15   pos_atual  $\leftarrow pos\_maior$ 
16   negativo  $\leftarrow$  não negativo
17 end

```

Após a localização da posição de cada bloco, são calculadas as larguras em pixels utilizando o algoritmo 2. Percebe-se que o vetor $lista_posicoes$ é percorrido e o valor corrente é subtraído do valor anterior. O resultado é então inserido no vetor $lista_larguras$. Na linha 2 se nota que o vetor $lista_posicoes$ é percorrida a partir da posição 1, pois na linha 3 o valor corrente é subtraído do valor anterior e também pelo primeiro bloco estar localizado na posição zero. Se o primeiro bloco estivesse situado na última posição, então o vetor $lista_posicoes$ seria percorrido a partir da última posição até a posição 1 e na linha 3 o valor da posição anterior seria subtraído do valor da posição corrente.

Em seguida, a lista da largura de cada bloco é dividida em duas sublistas: uma lista guarda as larguras dos blocos claros, já a outra lista armazena as larguras dos blocos escuros. Após isso, é calculada a intensidade média de cada bloco utilizando o algoritmo 3. Neste algoritmo, na linha 3 é feita a média aritmética dos intervalos de cada bloco, que

Algoritmo 2: Algoritmo que calcula a largura de cada bloco.

Input : Lista das posições de cada bloco: lista_posicoes

Output : Lista da largura de cada bloco: lista_larguras

```

1 lista_larguras ← Começa como uma lista vazia
2 for i ← 1 to TAMANHO_LISTA_POSICOES do
3   lista_larguras ← insere (lista_posicoes[i] - lista_posicoes[i - 1]) na
      lista_larguras
4 end

```

são indicados pelas posições listadas em *lista_posicoes*. O valor dessa média aritmética é a intensidade média daquele bloco.

Algoritmo 3: Algoritmo que calcula a altura média de cada bloco.

Input : Vetor do sinal suavizado: vetor_suavizado e lista das posições de cada bloco: lista_posicoes

Output : Lista da altura de cada bloco: lista_alturas

```

1 lista_alturas ← Começa como uma lista vazia
2 for i ← 1 to TAMANHO_LISTA_POSICOES do
3   lista_alturas ← insere a média de vetor_suavizado[lista_posicoes[i - 1] ...
      lista_posicoes[i]] na lista_alturas
4 end

```

4.4.2 Método da verificação do padrão

Este método de medição a partir da verificação do padrão de claros e escuros é realizado a partir da lista de alturas dos blocos. Para isso, foi projetado o algoritmo 4. Neste algoritmo foi definido DIFERENCA_MINIMA_ENTRE_BLOCOS como sendo a menor diferença de altura possível entre dois blocos de cores diferentes. A realização dos procedimentos do algoritmo começa com a atribuição à variável *desce* o valor *verdadeiro*, pois a passagem do primeiro bloco para o segundo é de descida, ou seja, de claro para escuro. A lista *mantem_padrao* tem como primeiro elemento o valor *verdadeiro*, pois se assume que o primeiro bloco segue a série de padrões a ser analisada. Em seguida o vetor *lista_alturas* é percorrido a partir do segundo elemento, como é visto na linha 3. Nesta passagem pelo vetor *lista_alturas* é calculada a diferença da altura do bloco corrente com a altura do bloco anterior. Na linha 5 é testada se essa diferença é muito pequena. Se sim, se assume que os dois blocos possuem a mesma altura. Posteriormente, na linha 8, é verificado se a passagem do bloco anterior para o bloco corrente está de acordo com o valor da variável de controle *desce*. Ou seja, se a passagem do bloco anterior para o bloco corrente for de descida e o valor da variável de controle *desce* for *verdadeiro*, então insere *verdadeiro* no vetor *mantem_padrao*. Se a passagem entre os blocos for de subida e o valor de controle for *falso*, insere *verdadeiro* no vetor *mantem_padrao*. Se a passagem entre os

blocos não corresponder com o valor da variável de controle, que é o caso da linha 11, insere *falso* no vetor *mantem_padrao*. Após a verificação a variável de controle é negada para que os outros padrões sejam verificados, como consta na linha 14, ou seja, se o valor era *verdadeiro*, ele passa a ser *falso* e vice-versa.

Algoritmo 4: Algoritmo que verifica o padrão de claros e escuros dos blocos.

Input : Lista da altura de cada bloco: *lista_alturas*
Output : Lista da verificação de cada bloco: *mantem_padrao*

```

1 desce  $\leftarrow$  Verdadeiro
2 mantem_padrao  $\leftarrow$  Primeiro valor inserido é Verdadeiro
3 for i  $\leftarrow$  1 to TAMANHO_LISTA_ALTURAS do
4   | diferenca  $\leftarrow$  lista_alturas[i] – lista_alturas[i – 1]
5   | if | diferenca |  $<$  DIFERENCA_MINIMA_ENTRE_BLOCOS then
6   |   | diferenca  $\leftarrow$  0
7   | end
8   | if ( desce e diferenca  $<$  0 ) ou ( não desce e diferenca  $>$  0 ) then
9   |   | mantem_padrao  $\leftarrow$  insere Verdadeiro na lista mantem_padrao
10  | end
11  | else
12  |   | mantem_padrao  $\leftarrow$  insere Falso na lista mantem_padrao
13  | end
14  | desce  $\leftarrow$  não desce
15 end

```

O resultado do algoritmo 4 pode ser verificado no gráfico da figura 31. Neste gráfico os blocos que seguem os padrões estão marcados em vermelho. Já os blocos marcados em amarelo indicam a localização na qual o padrão foi interrompido. Os valores presentes no gráfico representam a largura dos blocos claros e escuros. No gráfico se percebe que a ruptura do padrão aconteceu pelo fato de que era previsto que a passagem dos blocos fosse de descida, e não aconteceu. A passagem que ocorreu foi de subida, e é neste bloco em que o padrão é interrompido.

$$y = PRIM_PADR - DIST_PADR_PADR \times indice - ALT_BOIA \quad (4.4)$$

A lista *mantem_padrao* pode ser percorrida para localizar a primeira interrupção da continuidade do padrão. Esta localização da interrupção é então convertida em centímetros, utilizando a equação 4.4, obtendo assim a medição do nível d'água. Na equação 4.4 a constante *PRIM_PADR* é a medição na régua onde se encontra o primeiro padrão, que é na medição de 100 centímetros. A constante *DIST_PADR_PADR* se refere à distância do topo de cada padrão ao topo do padrão adjacente, que neste caso é de 2,5 centímetros. A variável *indice* é o valor do índice da primeira descontinuidade do padrão. Até essa parte da equação, foi medida a porção *x* da régua representada na figura 32. Para se obter a

medição do nível d'água, deve-se subtrair a porção h da figura 32, que representa a altura da boia. A constante ALT_BOIA da equação 4.4 representa a altura da boia, que neste caso é de 5 centímetros. Aplicando na equação 4.4 o índice da primeira interrupção, que é 27, o valor de y é de 27,5 centímetros. Assim, a medição do nível d'água desse exemplo é de 27,5 centímetros.

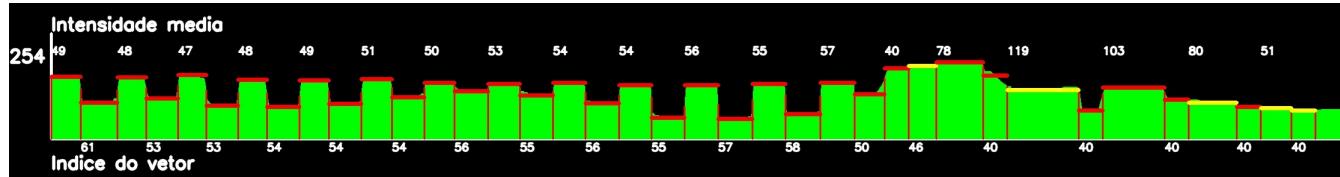


Figura 31 – Gráfico da figura 29 que ilustra a verificação dos padrões claros e escuros (Autoria própria).

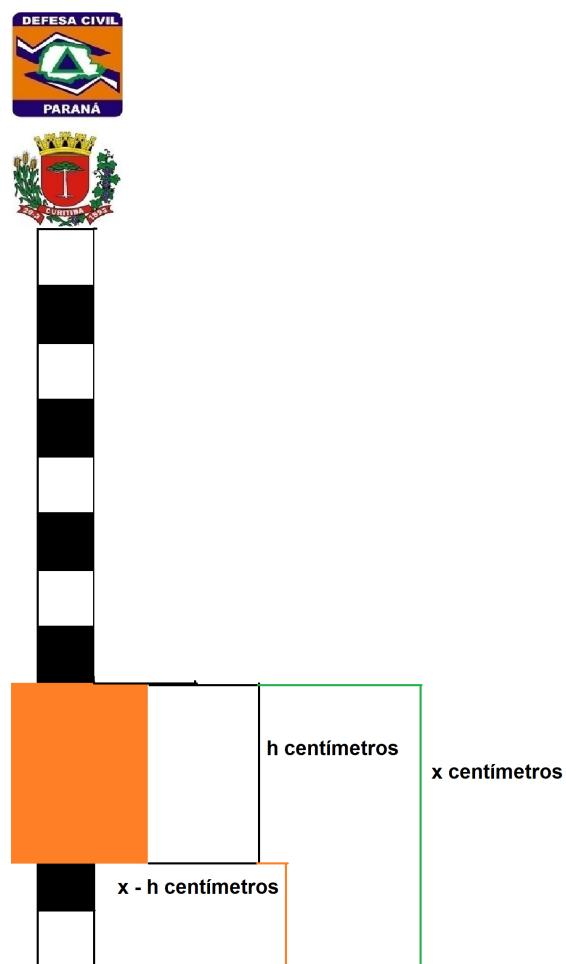


Figura 32 – Ilustração da medição do nível d'água dado o valor convertido do índice (Autoria própria).

4.4.3 Método iterativo baseado na largura das faixas

Este método de medição é realizado a partir das três listas de larguras de faixas computadas na subseção 4.4.1. Para isso, foi projetado o algoritmo 5 para identi-

ficar o padrão cuja largura é muito discrepante dos outros padrões. A constante *MINIMO_PONTOS_OUTLIERS* restringe o número de *outliers* de uma determinada lista de pontos. Os *outliers* são os padrões cujas larguras variam de uma forma muito diferente do restante da régua. A constante *THRESH* delimita a distância mínima de um ponto da lista até uma reta para não ser considerada *outlier*. Na linha 1 é realizada uma cópia da lista de entrada, para que não ocorram modificações na lista original. Na linha 2 é computada uma regressão linear nesta lista copiada. Os resultados desta regressão linear são os coeficientes da equação da reta. Na linha 3 é calculada a distância de cada ponto da lista à reta computada anteriormente. Na linha 4 é computado o número de *outliers*, que contabiliza cada distância que excedeu o valor da constante *THRESH*. Na linha 5 é verificado se o último elemento da lista é um *outlier*. Na linha 6 é verificado se existem muitos *outliers* na lista, e também se o número de elementos da lista é o mínimo possível para prosseguir. No interior do *loop* testado na linha 6 é retirado o último elemento da cópia da lista de entrada. Com a retirada do último elemento, a regressão linear é então recalculada e as outras variáveis que foram calculadas nas linhas 3, 4 e 5 são repetidas. Este processo é realizado até remanescer uma quantidade de elementos na lista cuja regressão linear produza o mínimo de *outliers*. Na linha 13 do algoritmo é computado o índice da lista onde a continuidade do padrão é interrompida.

$$y = PRIM_PADR_CLAR - DIST_PADR_CLAR \times indice - ALT_BOIA \quad (4.5)$$

$$y = PRIM_PADR_ESC - DIST_PADR_ESC \times indice - ALT_BOIA \quad (4.6)$$

Este algoritmo é aplicado na lista *lista_larguras* e também nas duas sublistas, *lista_claras* e *lista_escuras*. O algoritmo retorna o índice da largura discrepante de cada lista. Com o valor do índice, pode-se calcular a medição do nível em centímetros. Para o retorno do algoritmo aplicado na lista *lista_larguras* é usada a mesma equação 4.4 da seção 4.4.2, usando os mesmos valores das constantes. Para a lista *lista_claras* é usada a equação 4.5, onde a constante *PRIMEIRO_PADRAO_CLARO* é a medição na régua onde se encontra o primeiro padrão claro, que é na medição de 100 centímetros. A constante *DISTANCIA_PADRAO_CLARO* se refere à distância do topo de cada padrão claro ao topo do padrão claro adjacente, que neste caso é de 5 centímetros. Para a lista *lista_escuras* é usada a equação 4.6, onde a constante *PRIMEIRO_PADRAO_ESCURO* é a medição na régua onde se encontra o primeiro padrão escuro, que é na medição de 97,5 centímetros. A constante *DISTANCIA_PADRAO_ESCURO* se refere à distância do topo de cada padrão escuro ao topo do padrão escuro adjacente, que neste caso é de 5 centímetros.

Após a medição do nível d'água através das três listas, calcula-se também a média das medições obtidas da lista de todos os padrões, da lista dos padrões claros e da lista dos padrões escuros. Também é calculada a mediana da medição das três listas.

Algoritmo 5: Algoritmo que localiza a interrupção do padrão através de regressões lineares.

Input : Lista de pontos: *lista_pontos*
Output : Índice do ponto aonde a boia se encontra na régua: *indice*

```

1 lista  $\leftarrow$  cópia de lista_pontos
2  $[m, c] \leftarrow$  computa a regressão linear de lista
3 lista_distancia  $\leftarrow$  computa a distância de cada ponto de lista à reta da regressão linear
4 numero_outliers  $\leftarrow$  computa o número de elementos da lista lista_distancia que excedem o valor THRESH
5 ultimo_e_outlier  $\leftarrow$  verifica se o último elemento da lista lista_distancia excede o valor THRESH
6 while  $((\text{numero\_outliers} > \text{MINIMO\_PONTOS\_OUTLIERS} \text{ ou } \text{ultimo\_e\_outlier}) \text{ e } (\text{tamanho}(\text{lista}) \geq \text{MINIMO\_PONTOS\_LISTA}))$  do
7   lista  $\leftarrow$  Elimina o último elemento da lista lista
8    $[m, c] \leftarrow$  computa a regressão linear de lista
9   lista_distancia  $\leftarrow$  computa a distância de cada ponto de lista com a reta da regressão linear
10  numero_outliers  $\leftarrow$  computa o número de elementos da lista lista_distancia que excedem o valor THRESH
11  ultimo_e_outlier  $\leftarrow$  verifica se o último elemento da lista lista_distancia excede o valor THRESH
12 end
13 indice  $\leftarrow$  tamanho(lista) - 1

```

Como exemplo da aplicação do algoritmo 5, será usada a sublista de larguras de blocos claros. A lista das larguras é a seguinte: 49, 48, 47, 48, 49, 51, 50, 53, 54, 54, 56, 55, 57, 40, 78, 119, 103, 80, 51. Na figura 33a os pontos que estão na cor azul representam os pontos da lista de entrada do algoritmo. A reta ilustrada em vermelho foi computada através de uma regressão linear. Neste caso é verificado se o número de *outliers* é maior que um número mínimo de *outliers* que é admissível. Neste exemplo foi estimado que para ser um *outlier*, um ponto deve ter uma distância à reta maior que 8 pixels. Foi estimado também que o número mínimo de *outliers* aceitável é 1. O número de *outliers* computado que a reta produz com relação aos pontos da figura 33a foi 4. O último elemento da lista é um *outlier* também, e o número de elementos na lista é também o suficiente para prosseguir com o algoritmo. Este número de elementos suficiente estimado é de 3 elementos. Como o número de *outliers* é maior que o número de *outliers* aceitável, e o número de elementos da lista é o suficiente para prosseguir, então o último elemento da lista é eliminado e a regressão linear é recalculada. Esse processo é repetido para as retas das figuras 33b, 33c, 33d, e é percebido que a cada regressão linear, a reta se

justapõe aos pontos remanescentes. Na figura 33e se percebe que a reta se ajusta aos pontos remanescentes, e é a partir dessa regressão que o número de *outliers* volta a decrescer. O número de *outliers* nesta etapa é 2, como esse número é maior que o número mínimo aceitável, que é 1, então prossegue com o algoritmo eliminando o último elemento da lista. Após a eliminação, a reta da figura 33f é computada com a regressão linear. O número de *outliers* calculado a partir da reta da figura 33f é 1. Como o número de *outliers* não é maior que o mínimo aceitável, que é também 1, então o último elemento é verificado, que é um *outlier*. Então, o último elemento da lista é eliminado e a regressão linear desta nova lista é então calculada. Com a regressão linear, uma nova reta é computada, resultando na figura 33g. A partir desta nova reta, o número de *outliers* é extraído, que é zero *outliers*. Como o número de *outliers* não é maior que o número mínimo aceitável, que é 1, o último elemento é verificado, concluindo que ele não é *outlier*. Então, o índice em que o padrão é interrompido é calculado a partir do tamanho da nova lista, onde foram eliminados todos os *outliers*. O índice da interrupção é dado pelo decremento do valor do tamanho da nova lista. Neste exemplo, o índice onde houve a interrupção do padrão foi 12. A partir do valor do índice, pode-se converter para centímetros utilizando a equação 4.5. Com a constante *PRIMEIRO_PADRAO_CLARO* tendo o valor 100, a constante *DISTANCIA_PADRAO_CLARO* com o valor 5 e a constante *ALTURA_BOIA* tendo o valor 5. Substituindo a variável *indice* da equação 4.5 por 12, é obtido o valor 35. Assim, a medição do nível d'água a partir desse método neste exemplo é de 35 centímetros.

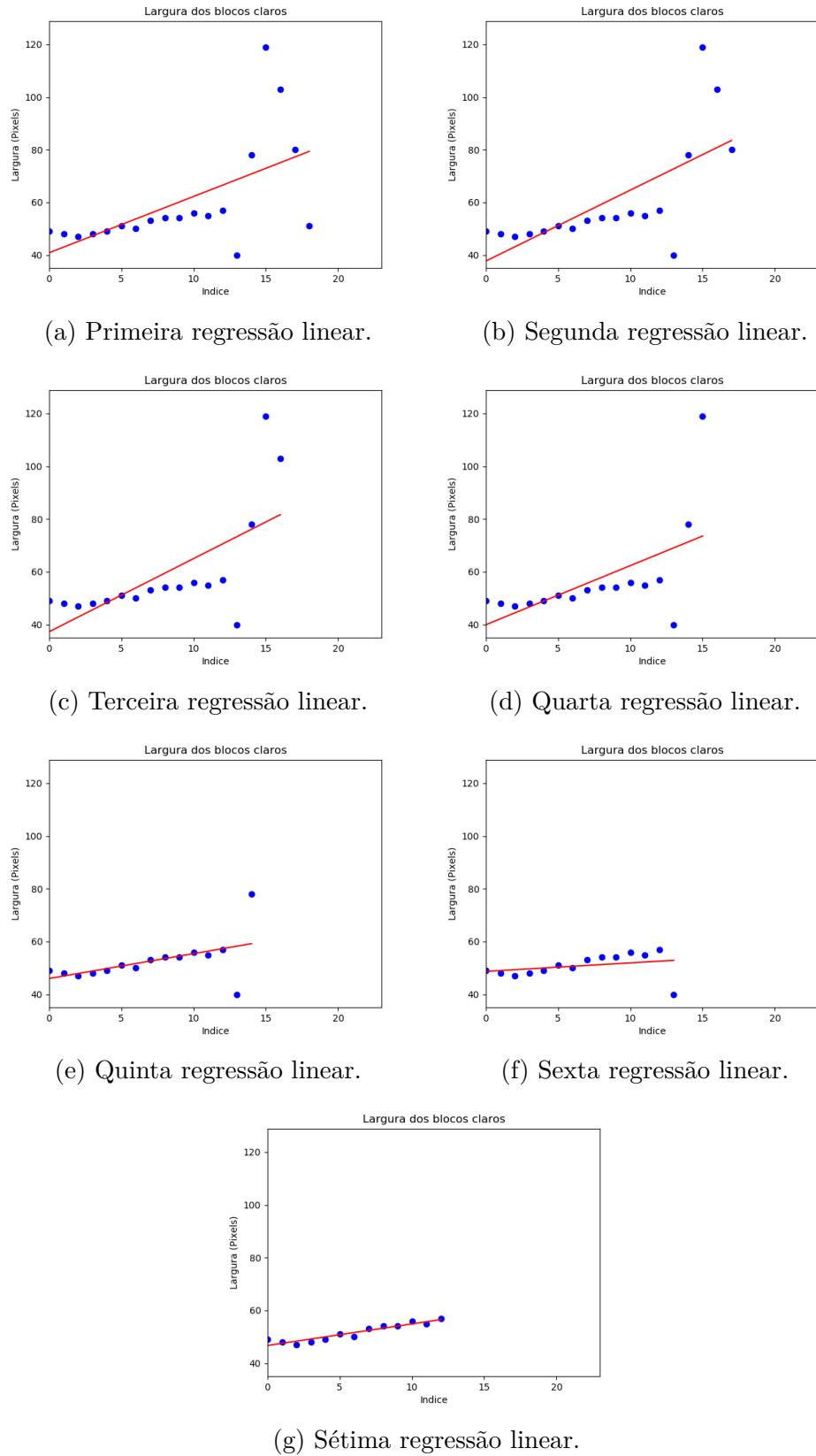


Figura 33 – Iterações do algoritmo 5 com os pontos da lista em azul e a reta da regressão linear em vermelho (Autoria própria).

5 Experimentos e Resultados

Após a explanação dos algoritmos utilizados no capítulo 4, serão detalhados no presente capítulo os experimentos realizados. Além dos experimentos, serão detalhados os resultados obtidos. Na seção 5.1 serão descritas a régua projetada e a imagem do modelo. Na seção 5.2 é descrito como o protótipo foi desenvolvido. Na seção 5.3 são expostos os experimentos da aplicação do protótipo. A seção 5.4 discute os resultados obtidos dos experimentos descritos na seção 5.3.

5.1 Descrição da régua

Para o experimento que será descrito na seção 5.3 foi construída uma régua em escala reduzida, de 100 centímetros, composta por uma imagem do modelo que fica no topo da régua, uma série de padrões do topo à base da mesma e uma boia.

A imagem do modelo é composta pelo símbolo da *Coordenadoria Estadual da Defesa Civil*, extraído do site da *Defesa Civil do Paraná*¹. O símbolo está ilustrado na figura 34a. Outro símbolo que compõe o modelo é da *Prefeitura Municipal de Curitiba*, extraída do site da prefeitura². A figura 34b mostra a imagem desse símbolo do município. Para se obter a imagem do modelo, a imagem do símbolo da *Defesa Civil do Paraná* é posicionada no topo da imagem do símbolo da *Prefeitura Municipal de Curitiba*, como mostra a ilustração da figura 34c. No caso do experimento deste trabalho, a altura da imagem do modelo ficou de 17,8 centímetros. Em seguida, a imagem do modelo foi fixada em uma superfície rígida de papelão.

A régua utilizada possui coloração cinza, com as letras das métricas em preto e material metálico. Ela possui altura de 100 centímetros (1 metro). Com a imagem do modelo já pregada na superfície rígida, cola-se a imagem na parte superior dianteira da régua. O extremo inferior da imagem do modelo encontra-se exatamente na medição de 100 centímetros (1 metro). A parte superior da régua deve se encontrar no centro da parte inferior da imagem do modelo. Estas primeiras medições estão dispostas na figura 35a. Como podem ser observadas na figura 35b, as faixas claras foram compostas por fitas adesivas de cor branca. Estas fitas, que possuem largura de 2,5 centímetros, foram fixadas na régua exatamente nas métricas: 100, 95, 90, 85, 80, 75, 70, 65, 60, 55, 50, 45, 40, 35, 30, 25, 20, 15, 10, 5, 0. Estas métricas estão na unidade de medida em centímetros. As faixas escuras foram compostas pelo espaço entre as duas fitas, espaços que também correspondem a 2,5 centímetros. A menor largura de faixa aceitável é determinada pela

¹ Site da *Defesa Civil do Paraná*: <www.defesacivil.pr.gov.br>

² Site da *Prefeitura Municipal de Curitiba*: <www.curitiba.pr.gov.br/>

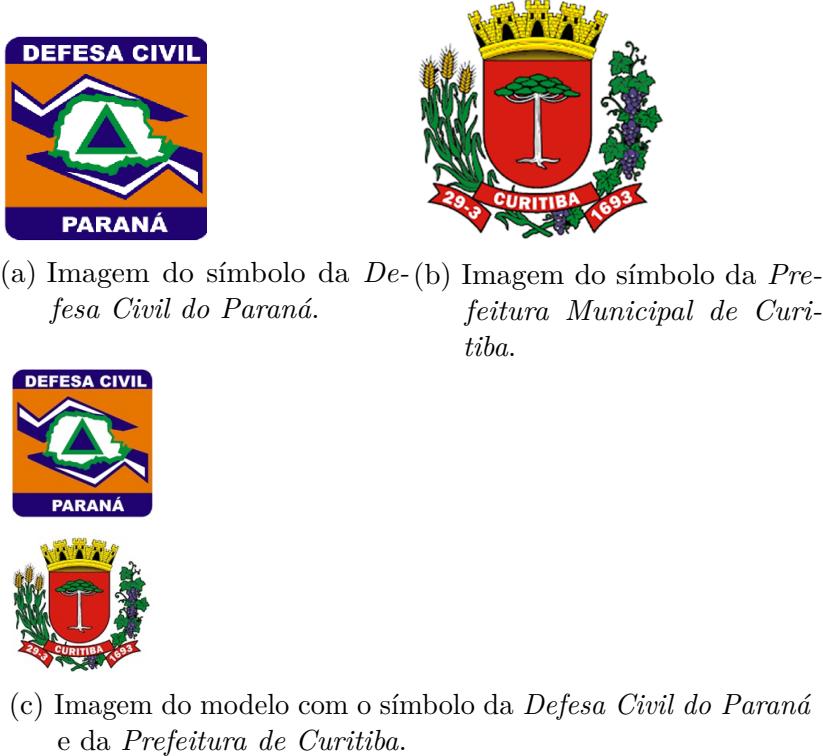


Figura 34 – Imagens dos símbolos que formam a imagem do modelo presente na figura 34c (Autoria própria).

largura da mesma nas imagens da câmera. Quando a câmera possui uma melhor resolução e qualidade, ela consegue capturar as faixas de menor largura. Isso implica em maior precisão.

Também foi projetada uma boia conforme a medição demonstrada na figura 35a, ela foi concebida com material de isopor, cromatizada em laranja, com altura de 5 centímetros e revestida com plástico, pois o líquido utilizado para colorá-la é sensível a água. A concepção final da boia está ilustrada na figura 35c.

5.2 Descrição do protótipo do sistema proposto

O protótipo do sistema foi desenvolvido na linguagem *Python*, versão 2.7.14, pelo fato da sua sintaxe ser elegante e da sua aplicabilidade em várias áreas do conhecimento (Python Software Foundation, 2019). A biblioteca usada para processamento digital de imagens foi o *OpenCV*, que é uma biblioteca de código aberto de visão computacional, projetada para aplicações de tempo real e eficiência computacional (Bradski; Kaehler, 2008). Ela pode ser usada para programas escritos em *Python*. A versão utilizada nesse protótipo foi 3.0.0.

A câmera usada para os experimentos é o *smartphone LG K10*. Ela possui 1920x1080 de resolução de vídeo *full HD*.



Figura 35 – Imagens que ilustram as medidas dos acessórios que a régua apresenta e a sua versão final (Autoria própria).

5.3 Descrição do experimento

Para o teste do protótipo do sistema proposto foi concebido um experimento em um corpo d’água com uma medição de 25 centímetros. Um exemplo deste experimento está ilustrado na figura 36. Uma câmera capturou aproximadamente 10 segundos em cinco vídeos da régua imersa, ou seja, cada um desses cinco vídeos contém 10 segundos dessa régua imersa. Para que as amostras alimentassem o protótipo, foi realizada uma calibração com as constantes portando os valores da tabela 1.

5.4 Resultados obtidos

Com as amostras do experimento sendo computadas no protótipo, foram obtidas as medições da tabela 2. Calculando os erros em centímetros, foi obtida a tabela 3.

Observando a tabela 3, foi notado que as maiores discrepâncias aconteceram quando foram computadas regressões lineares na lista de largura de blocos claros e escuros da primeira e segunda amostra. Para a primeira amostra foram obtidos erros de 10 e 22,5 centímetros nas regressões lineares na lista dos blocos claros e escuros respectivamente.



Figura 36 – Imagem exemplo do experimento realizado. (Autoria própria).

Constante	Valor
HOR_WINDOW_WIDTH	7
VERT_WINDOW_WIDTH	25
MIN_BLOCK_WIDTH	40
MAX_BLOCK_WIDTH	120
N_STEPS_DX	5
DIFERENCA_MINIMA_ENTRE_BLOCOS	10
MINIMO_PONTOS_OUTLIERS	1
MINIMO_PONTOS_LISTA	3
THRESH	8
ALTURA_PLACA	17,8
ALTURA_REGUA	100
ALTURA_BOIA	5

Tabela 1 – Tabela de constantes do experimento no corpo d’água de 25 centímetros (Autoria própria).

Amostra	Visual	Padrão	Claras	Escuras	Claras Escuras	Média	Mediana
1	25	27,5	35	2,5	32,5	23,33	32,5
2	25	27,5	35	7,5	27,5	23,33	27,5
3	25	27,5	35	27,5	27,5	30	27,5
4	25	27,5	30	27,5	27,5	28,33	27,5
5	25	27,5	35	27,5	32,5	31,67	32,5

Tabela 2 – Tabela das medições obtidas do primeiro experimento, valores em centímetros(Autoria própria).

Para a segunda amostra foram obtidos erros de 10 e 17,5 centímetros, respectivamente.

Analizando o erro da regressão linear da lista de blocos claros da primeira amostra,

Amostra	Padrão	Claras	Escuras	Claras Escuras	Média	Mediana
1	2,5	10	22,5	7,5	1,67	7,5
2	2,5	10	17,5	2,5	1,67	2,5
3	2,5	10	2,5	2,5	5	2,5
4	2,5	5	2,5	2,5	3,33	2,5
5	2,5	10	2,5	7,5	6,67	7,5

Tabela 3 – Tabela dos erros obtidos do primeiro experimento, valores em centímetros(Autoria própria).

observa-se que da primeira iteração até a quinta o número de *outliers* era maior que 1. Essa quantidade de *outliers* maior que 1 pode ser notada nas figuras 37a e 37b, cada uma representando a primeira e a quinta iteração, respectivamente. Na figura 37a a reta computada a partir da regressão não está ajustada aos pontos, fazendo com que o número de *outliers* seja maior que 1. Já na figura 37b é observada que a reta formada começa a se ajustar aos pontos, atenuando o número de *outliers*. Mas na iteração seguinte, na sexta, a lista de pontos possui apenas um *outlier*, mas o último elemento da lista é também um *outlier*, como pode ser observado na figura 37c. Na última iteração não existem *outliers*, finalizando a medição para a lista de largura de blocos claros, situação ilustrada na figura 37d, constatando que a reta se ajustou à maioria dos pontos. Com isso, o índice em que o algoritmo identificou o nível que a boia se encontra é 12. Assim, convertendo o índice para centímetros através da equação 4.5, utilizando o valor 100 para a constante PRIM_PADR_CLAR, o valor 5 para DIST_PADR_CLAR, o valor 5 para ALT_BOIA e o valor 12 para a variável indice, é alcançado o valor 35, assim a medição calculada é de 35 centímetros. Esse valor encontrado tem uma diferença de 10 centímetros do valor real, que é de 25 centímetros, isso ocorre pois, de acordo com o algoritmo implementado da iteração de regressões lineares, a lista que se formou na sétima iteração não formou uma quantidade de *outliers* maior que 1, ou seja, não existe um ponto na lista que tem distância a reta maior que 8. Esse local não é onde está o nível do corpo d'água, como ilustra a figura 39, sendo que o índice 12 está representado como um círculo vermelho. O índice deveria ser 14, que está representado como um círculo verde na figura 39, pois convertendo para centímetros com a equação 4.5, seria 25 centímetros. Além disso, observando a figura 39, percebe-se que os blocos dos índices 13 e 14 possuem larguras bem diferentes dos outros blocos, sendo *outliers*, por isso essas larguras são descartadas pelo algoritmo 5. Essa discrepância da largura do bloco do índice 13 e 14 é justificada pela iluminação, pois a boia foi revestida com material plástico.

Na lista de larguras dos blocos escuros, no primeiro cálculo de número de *outliers* ele encontra zero *outliers*, como pode ser visto na figura 38. Neste caso não existe ponto cuja distância até a reta seja maior que o THRESH, que no caso é 8. A maior distância de um ponto até uma reta, no exemplo da lista de largura de blocos escuros é 6,5. No caso da lista

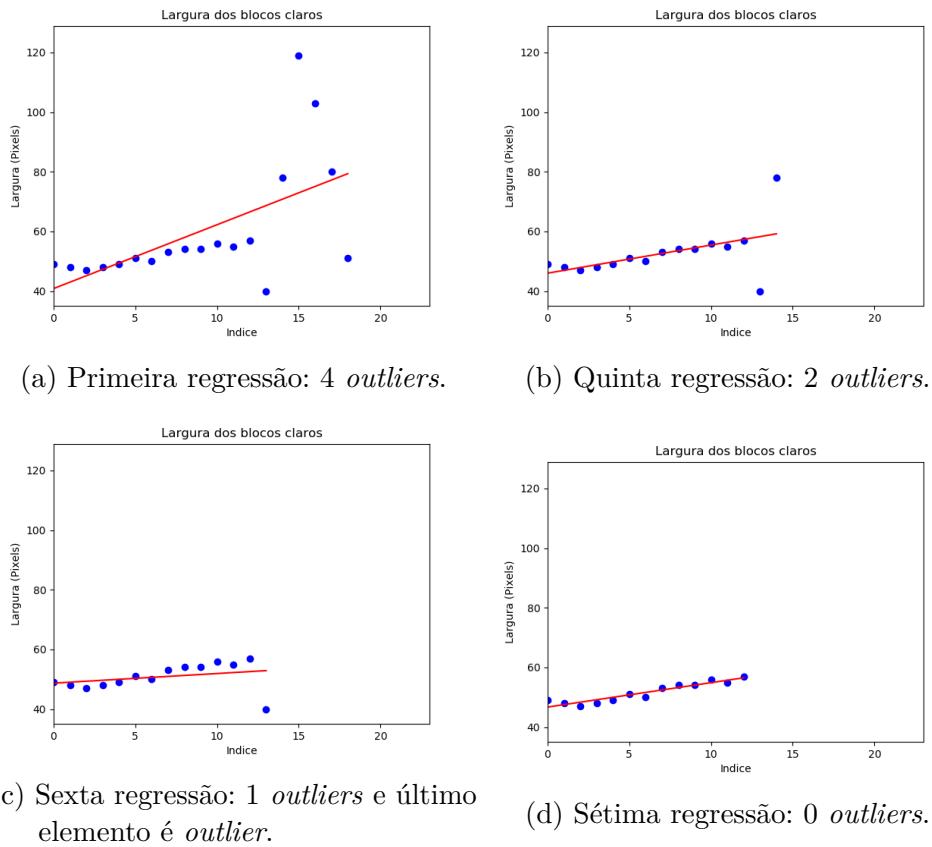


Figura 37 – Iterações de regressões lineares da pior medida dos blocos claros da primeira amostra (Autoria própria).

de largura de blocos escuros, o índice retornado foi 18, que pode ser usado para converter em centímetros utilizando a equação 4.6. Utilizando a constante *PRIM_PADR_ESC* com valor 97,5, a constante *DIST_PADR_ESC* com valor 5, a constante *ALT_BOIA* com valor 5 e a variável *indice* com valor 18, é obtido o valor 2,5, ou seja, a medição calculada é 2,5 centímetros. Essa medição calculada tem um erro de diferença da medição real de 22,5 centímetros. O índice retornado não é o local correto em que o nível se encontra, pois para o algoritmo o nível se encontra submerso a água, como pode ser visto na figura 39, onde esse índice está representado com a cor amarela.

Observando o erro da segunda amostra, as regressões lineares na lista de blocos claros da primeira a sétima regressão obtiveram número de *outliers* maior que 1. Essas regressões estão representadas nas figuras 40a, que ilustra a primeira iteração, e na figura 40b, que ilustra a sétima iteração. Observa-se também que a reta das regressões lineares se ajusta aos pontos da lista a cada iteração, atenuando o número de *outliers*. Na oitava regressão, ilustrada na figura 40c, o número de *outliers* foi igual a 1, mas o último elemento da lista do algoritmo 5 também é *outlier*, então é realizada mais uma regressão linear, que está representada na figura 40d. Nesta iteração, o número de *outliers* é zero, finalizando o algoritmo, constatando que a reta da última regressão linear computada se ajustou à

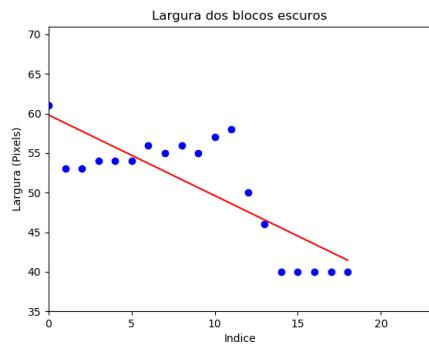


Figura 38 – Primeira iteração de regressões lineares da pior medida dos blocos escuros da primeira amostra: 0 *outliers* (Autoria própria).

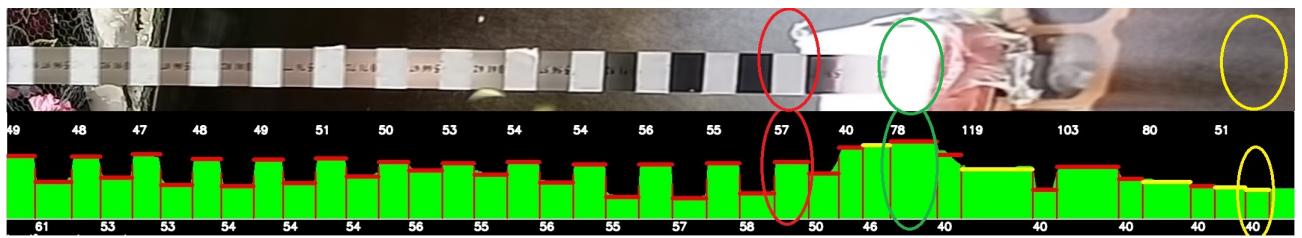


Figura 39 – Locais da régua que o algoritmo computou ser os locais onde o nível d'água se encontra da primeira amostra (Autoria própria).

maioria dos pontos da lista. Como na primeira amostra, o índice que o algoritmo retorna é 12, substituindo na equação 4.5, com os mesmos valores para as constantes e o valor 12 para a variável *indice*, a medição calculada é 35 centímetros. Essa medição possui diferença de 10 centímetros da medição real. Nesse caso, o índice deveria ser 14, como mostra a figura 42, onde o índice 12 está representado como um círculo vermelho e o índice 14 como círculo verde. Pode ser observada na figura 42 que a largura dos blocos 13 e 14 é mais discrepante com relação aos blocos dos índices anteriores, fazendo com que elas sejam *outliers*, sendo eliminadas da lista do algoritmo 5.

Para as regressões lineares na lista de blocos escuros, na primeira regressão o número de *outliers* é maior que 1, possuindo 7 *outliers*, como pode ser visto na figura 41a. Na segunda regressão o número de *outliers* é atenuada para 1, como pode ser observado na figura 41b. Nesta nova regressão, a maior distância de um ponto até a reta é 11,21, ou seja, esse ponto é um *outlier*, como ele é o último elemento da lista do algoritmo 5, ele é descartado e uma nova regressão linear é computada, essa nova regressão está ilustrada na figura 41b. Como pode ser observado, nenhum ponto possui distância até a reta maior que 8, assim, o algoritmo retorna o índice 17. Substituindo o índice 17 na equação 4.6, com os mesmos valores da primeira amostra para as constantes e o valor 17 para a variável *indice*, a medição calculada é 7,5 centímetros. Essa medição calculada tem um erro de diferença da medição real de 17,5 centímetros. Como pode ser visto na figura 42, esse índice representa uma parte que está submersa da régua, que está representado com a cor

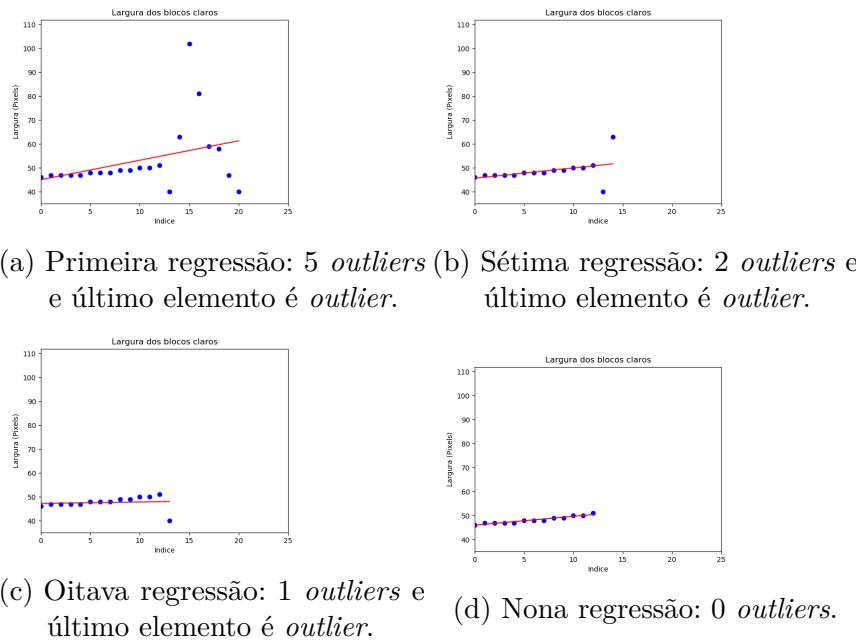


Figura 40 – Iterações de regressões lineares da pior medida dos blocos claros da segunda amostra (Autoria própria).

amarela, não podendo ser o local do nível d'água.

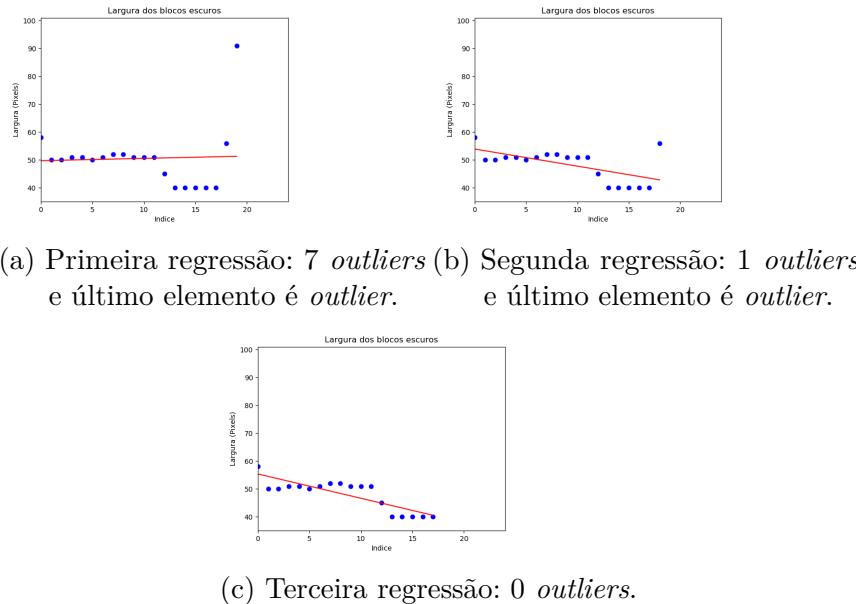


Figura 41 – Iterações de regressões lineares da pior medida dos blocos escuros da segunda amostra (Autoria própria).

Com isso, a medição que causou a menor diferença em comparação à medição real foi a que utilizou o método de verificação dos padrões, que obteve diferença máxima de 2,5 centímetros com relação à medição real. Isso se deve ao fato da iluminação local interferir pouco. A iluminação interfere pouco nesse método pois, é analisada a continuidade dos padrões, que é independente à largura dos blocos. A iluminação pode ser um empecilho para

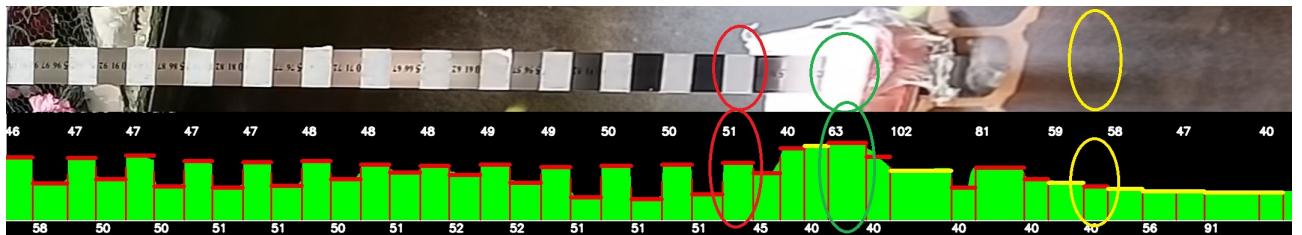


Figura 42 – Locais da régua que o algoritmo computou ser os locais onde o nível d'água se encontra da segunda amostra (Autoria própria).

esse método quando a análise acontecer nos padrões próximos à boia pois, o material que reveste ela é de plástico, que reflete a iluminação. Essa reflexo pode alterar a continuidade dos padrões, o que não ocorreu no experimento. Para o método das iterações baseadas na largura dos blocos, a iluminação pode ser um problema, pois ela altera a largura dos blocos claros e escuros, tornando-os *outliers*. Como pode ser observado nas figuras 39 e 42, no índice 26 é percebido que esse reflexo existe, alterou a largura dos blocos dos índices 26, 27, tornando-os *outliers*. Além disso, o cálculo do número de *outliers* depende do formato da reta da regressão linear, que muda quando a câmera se afasta ou se aproxima dos blocos claros e escuros e da iluminação local. Se o formato da reta muda, o valor da constante *THRESH* também deve ser alterada pois, a distância dos pontos à reta será diferente. O método de medição através das iterações baseadas nas larguras obteve uma diferença máxima entre a calculada e a real de 22,5 centímetros. Quando se calcula a média das medições através do método das iterações, ou seja, a média da medição utilizando a lista da largura dos blocos claros, a medição utilizando a lista dos blocos escuros e a medição utilizando a lista com todos os blocos, a diferença máxima é atenuada para 6,67. Pelo cálculo da mediana da medição utilizando as três listas anteriores, a diferença máxima é diminuída para 7,5 centímetros.

6 Conclusão

Neste trabalho foi abordado o problema da medição do nível de rios. Observou-se que as enxurradas e o alagamento são alguns dos desastres que mais ocorrem no Paraná e, além de causarem danos humanos, causam prejuízos. Além disso, uma forma que as autoridades possuem de receberem informações a respeito de desastres ambientais é através dos Núcleos Comunitários de Defesa Civil. O conceito de *smart city* pode ser explorado para incorporar o sistema de medição do nível de corpos d'água, pois o mesmo pode notificar as autoridades e a população de enchentes, alagamentos e enxurradas que podem ocorrer em uma determinada região da cidade. Com o levantamento bibliográfico, percebeu-se que os trabalhos recentes com a medição do nível de rios aplicam os algoritmos clássicos de processamento digital de imagens, que são sensíveis à iluminação local e perspectiva da câmera. O sistema automatizado proposto nesse trabalho abrange dois métodos de medição, o método da verificação do padrão e o método das iterações baseadas nas larguras, que buscam atenuar a interferência da iluminação e da perspectiva da câmera.

Os resultados obtidos mostram que o método da verificação do padrão foi o que atingiu um erro máximo menor que o outro apresentado, que foi de 2,5 centímetros. Além disso, percebeu-se que a interferência da iluminação é pequena, permitindo que o método funcione em ambientes com iluminação variável.

O método das iterações baseadas nas larguras alcançou erros variados, sendo que o erro máximo atingido foi de 22,5 centímetros. Apesar dessa variação no erro, as médias e as medianas das medições utilizando as três listas atenuaram o erro para 6,67 e 7,5 centímetros, respectivamente. Esse método considera que a largura dos blocos é constante, apenas variando nos blocos nos quais o nível se encontra.

Apesar disso, a iluminação ainda é um empecilho, pois ela pode provocar a descontinuidade do padrão, como também alterar a largura dos blocos claros e escuros. Isso ocorre devido ao material utilizado para revestir a boia, que é de plástico, refletir. Uma forma de contornar esse problema é modificando o material usado para revestir a boia usando materiais menos reflexivos.

Para futuros trabalhos, é sugerida a automatização do procedimento descrito na seção 4.3 do capítulo 4 pois, se desejar posicionar a câmera em outro local, esse procedimento deve ser realizado novamente, gerando uma nova equação relacionando a medição real e a medição da imagem da régua. A etapa da detecção da régua depende dessa relação para que a base da régua seja estimada. Também é aconselhada a automatização da calibração dos valores das constantes da tabela 1 pois, com a mudança da posição da câmera, a reta oriunda da regressão linear é alterada também, modificando os valores

das constantes. Também é sugerida a experimentação do sistema proposto em um corpo d'água real, para que sejam realizados testes em rios, lagos ou esgotos.

Referências

- Bradski, D. G. R.; Kaehler, A. *Learning OpenCV, 1st Edition*. First. [S.l.]: O'Reilly Media, Inc., 2008. ISBN 9780596516130. Citado 2 vezes nas páginas [26](#) e [46](#).
- Chourabi, H. et al. Understanding smart cities: An integrative framework. In: *2012 45th Hawaii International Conference on System Sciences*. [S.l.: s.n.], 2012. p. 2289–2297. ISSN 1530-1605. Citado na página [12](#).
- Coordenadoria Estadual de Proteção e Defesa Civil. *Anuário da Defesa Civil do Paraná Ações Desenvolvidas em 2017*. <http://www.cepel.pr.gov.br/arquivos/File/Anuario2017CEPED.pdf>, 2018. Acessado em: 26/03/2019. Citado na página [12](#).
- Coordenadoria Estadual de Proteção e Defesa Civil. *NUDEC*. 2019. Disponível em: <<http://www.defesacivil.pr.gov.br/Pagina/NUDEC>>. Citado na página [12](#).
- Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Citado 2 vezes nas páginas [13](#) e [26](#).
- Gonzalez, R. C.; Woods, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X. Citado na página [24](#).
- Hübner, K. L. *Medição automática de nível de água baseado em imagens : Caso do rio Itajaí-Açú*. 2013. Monografia (Bacharel em Ciência da Computação), UDESC (Universidade do Estado de Santa Catarina), Joinville, Brasil. Citado 5 vezes nas páginas [7](#), [16](#), [20](#), [21](#) e [22](#).
- Lin, F. et al. Applications of image recognition for real-time water level and surface velocity. In: *2013 IEEE International Symposium on Multimedia*. [S.l.: s.n.], 2013. p. 259–262. Citado 3 vezes nas páginas [7](#), [16](#) e [17](#).
- Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 60, p. 91–, 11 2004. Citado 2 vezes nas páginas [13](#) e [24](#).
- Morel, G.; Hissel, F. The impact of flooding on transport networks: optimization of action plans in a decision support tool. In: *River Basin Management 2019*. [S.l.: s.n.], 2019. Citado na página [12](#).
- OpenCV-Python Tutorials. *The Python Tutorial*. 2019. <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html>. Acessado: 2019-10-11. Citado na página [25](#).
- Python Software Foundation. *The Python Tutorial*. 2019. <<https://docs.python.org/3/tutorial/index.html>>. Acessado: 2019-08-14. Citado na página [46](#).
- Sakaino, H. Camera-vision-based water level estimation. *IEEE Sensors Journal*, v. 16, n. 21, p. 7564–7565, Nov 2016. ISSN 1530-437X. Citado 4 vezes nas páginas [7](#), [16](#), [19](#) e [20](#).

Xu, Z. et al. Water level estimation based on image of staff gauge in smart city. In: *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. [S.l.: s.n.], 2018. p. 1341–1345. Citado 4 vezes nas páginas [7](#), [16](#), [18](#) e [19](#).