



**Universidade de Brasília
Faculdade de Tecnologia**

**Estudo e Desenvolvimento de um Protótipo
de um Scanner de Projétil
Portátil para Criminalística**

Filipe Aziz Batista

PROJETO FINAL DE CURSO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília
2023

**Universidade de Brasília
Faculdade de Tecnologia**

**Estudo e Desenvolvimento de um Protótipo
de um Scanner de Projétil
Portátil para Criminalística**

Filipe Aziz Batista

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Orientador: Prof. Dr. Walter de Britto Vidal Filho

Coorientador: Prof. Dr. Fernando Merege

Brasília

2023

A995e Aziz Batista, Filipe.
Estudo e Desenvolvimento de um Protótipo de um Scanner de Projétil Portátil para Criminalística / Filipe Aziz Batista; orientador Walter de Britto Vidal Filho; coorientador Fernando Merege. -- Brasília, 2023.
183 p.

Projeto Final de Curso (Engenharia de Controle e Automação)
-- Universidade de Brasília, 2023.

1. Balística Forense. 2. Confronto Balístico. 3. Microestruturas. 4. Processamento de Imagens e Sinais. I. de Britto Vidal Filho, Walter, orient. II. Merege, Fernando, coorient. III. Título

**Universidade de Brasília
Faculdade de Tecnologia**

**Estudo e Desenvolvimento de um Protótipo
de um Scanner de Projétil
Portátil para Criminalística**

Filipe Aziz Batista

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Trabalho aprovado. Brasília, 25 de Julho de 2023:

Prof. Dr. Walter de Britto Vidal Filho,
UnB/FT/ENM
Orientador

Prof. Dr. Guilherme Caribe de Carvalho,
UnB/FT/ENM
Examinador interno

**Prof. Dr. José Maurício Santos Torres da
Motta, UnB/FT/ENM**
Examinador interno

Dr. Fernando Merege, Externo
Coorientador

Brasília
2023

Este trabalho é dedicado aos meus pais que sempre estiveram ao meu lado, desde pequeno me incentivando a seguir meus sonhos, nunca medindo esforços para que eu pudesse segui-los. Vocês fizeram de tudo para que eu pudesse estar aqui hoje e sei que continuam se esforçando para que eu continue a traçar meu caminho. Sempre foram referências para mim, como pessoas, como pais, como casal e como profissionais. Dedico à minha irmã, que sempre foi minha maior fã, minha maior impulsionadora, minha maior torcedora, que nunca me deixou de lado, mesmo nos momentos de pouquíssimas interações, sempre foi e continua sendo uma luz na minha vida. A vocês, que estiveram ao meu lado durante os altos e baixos, dedico este trabalho como um reflexo do esforço e da dedicação que compartilhamos.

Agradecimentos

Agradeço primeiramente à minha família, em especial aos meus pais, à minha irmã, à minha avó Ivone e minhas tias Solange e Rosângela que estão e sempre estiveram ao meu lado a cada dia da minha jornada na graduação, me apoiando, me incentivando e jamais me pressionando, me dando muito amor e carinho em todos os momentos, principalmente nos mais difíceis.

Aos meus pais, agradeço por confiarem nas minhas escolhas e deixarem com que eu fosse capaz de amadurecer e aprender com meus erros, me apoiando a cada passo e me orientando durante essa caminhada.

À minha irmã, agradeço pelas conversas e momentos de muita risada e descontração, por ser minha inspiração durante todo esse processo, inspiração de força, de resiliência e de dedicação, por me mostrar que há momentos de extrema dedicação e momentos de descanso, sendo necessário muito balanceamento entre tais momentos, e por toda a atenção e aconselhamento de sempre.

À minha avó Ivone, e minhas tias Solange e Rosângela, que sempre me acompanharam, me aconselharam, me transmitiram confiança, amor, carinho e muita força. Vocês, mesmo que em muitos momentos de longe, foram um dos meus pilares durante todo esse processo.

Agradeço aos meus avós, Antônio e Maria do Socorro, que não se encontram mais entre nós, mas que desde sempre me apoiaram e se orgulharam das minhas conquistas. Aos meus tios, tias, primos e primas que sempre se preocuparam e me apoiaram.

Agradeço a todos os meus amigos que estão ao meu lado desde a escola, Guilherme, Fernanda, Rafael, Maria, Elisa, Giulliano, João e André, que mesmo com certo afastamento após cada um seguir seu caminho, continuaram a me incentivar e nunca me deixaram de lado, sempre me alegrando e me dando ânimo e confiança para continuar.

Aos meus amigos da graduação, que me receberam e me acolheram após minha entrada tardia na turma e até hoje me apoiam e estão comigo em todas as situações. Em especial à Natalia que foi a primeira amizade a me acolher e que sempre foi gentil quando eu merecia e rígida quando necessário, sendo de grande influência para meu amadurecimento dentro da universidade. À Lívia pelos ótimos momentos vividos, pelo apoio em muitos momentos e por muitos aprendizados obtidos nessa convivência de muito tempo. Ao Arthur que foi a primeira pessoa que conheci na turma e que foi responsável por me enturmar, um amigo especial que me apresentou aos meus demais amigos, por estar comigo desde então, em todas as etapas da graduação, nas mais baixas e mais altas, foi minha dupla de

todos os momentos, me marcou grandemente em diversas situações, não me deixou desistir e me impulsionou a ser melhor do que achei que poderia ser. Ao Marcus, outro grande amigo, completando o trio de sempre, foi um amigo relativamente tardio, amizade que começou após alguns anos de curso, mas que foi e é especial, que esteve ao meu lado nos piores momentos e me viu na minha pior fase, me ajudando a melhorar e sendo de grande importância durante todo o percurso. Agradeço enormemente a todos vocês por fazerem parte da minha vida e me inspirarem sempre, espero poder viver muito mais experiências com todos vocês.

Agradeço, ainda, à Isabelle, que embora esteja há menos tempo no meu cotidiano, fez e faz a maior diferença, me apoiando, ma ajudando, me incentivando, me impulsionando, me alegrando, me dando todo carinho e amor durante todo o processo desde que entrou em minha vida. Sem você eu não estaria onde estou, muitas de minhas conquistas tiveram sua influência e espero que essa influência continue em minha vida.

Venho agradecer, também, ao meu orientador, professor Walter de Britto, que me deu a oportunidade de trabalhar com este projeto, me auxiliando em todo o processo, aceitando todos os obstáculos que passei no caminho e não deixando de cobrar e buscar resultados, sempre se importando com a minha evolução.

Às professoras Carla Koike e Dianne Viana e ao professor Jones Yudi que me aceitaram como participante do Grupo Ereko e que me orientaram e me suportaram por alguns anos, transmitindo diversos ensinamentos, sempre me tratando com muito respeito e carinho. Foram e continuam a ser uma grande inspiração de pessoas e profissionais para mim. Agradeço por guiarem meu caminho com sabedoria e paciência, moldando-me não apenas como um estudante, mas como um pensador crítico.

Por fim, agradeço a Deus por me guiar por todo o caminho que tracei e me possibilitar chegar até aqui, colocando pessoas tão especiais em minha vida.

"Não há conquista que valha a pena que seja fácil; as coisas mais valiosas são aquelas que exigem esforço, perseverança e determinação. O caminho em direção ao sucesso pode ser árduo e repleto de desafios, mas é precisamente essa jornada que nos molda e nos capacita a alcançar nossos objetivos mais elevados. Cada obstáculo superado é um degrau em direção ao topo da montanha da realização. Portanto, não tema as dificuldades, pois são elas que nos ensinam, nos fortalecem e nos conduzem às verdadeiras vitórias."

(Albert Einstein)

Resumo

A balística forense vem cada vez mais abraçando a tecnologia e seus avanços, melhorando seus resultados e trazendo novos e diferenciados laudos que têm mudado a dinâmica da criminalística. Ainda que a cada dia novos avanços sejam registrados e novas conquistas sejam alcançadas, há diversas formas de se explorar a tecnologia e muitas delas ainda não foram aplicadas. Dessa forma, em meio aos laudos da balística forense, um novo conceito surge, levar a comparação balística para o campo. Diversos ramos da perícia criminal já atuam em campo e é notável o benefício que tal serviço traz à criminalística, estimulando a extensão da balística para análises em campo.

Sendo assim, este trabalho visa perseguir essa ideia, trazendo nesse relatório um desenvolvimento de um protótipo físico capaz de escanear os projéteis e um software capaz de transformar a casca cilíndrica de um projétil em plana por meio de várias fotos. O software também aplica técnicas de visão computacional e processamento de sinais para analisar marcas deixadas por armas de fogo nos projéteis. O trabalho tem como objetivo testar algumas metodologias para chegar ao dispositivo e software mencionados, permitindo a realização de análises preliminares para auxiliar em cenas de crimes e possibilitar a agilização na identificação e apreensão de criminosos.

O presente trabalho apresenta uma revisão bibliográfica acerca do tema, situando o leitor na origem das armas e munições, no funcionamento da balística forense e nos atuais métodos de se realizar confrontos balísticos. São desenvolvidos fundamentos teórico importantes para o correto entendimento das metodologias utilizadas, mostrando métodos de processamento de imagens e de sinais utilizados. A metodologia é explanada e os resultados são evidenciados mostrando que os objetivos foram atingidos e obteve-se sucesso na criação do dispositivo, no escaneamento do projétil, no processamento das imagens, na extração dos sinais e na comparação realizada. São concluídos aspectos importantes sobre todo o processo realizado e toda a metodologia aplicada, sugerindo-se alterações e evoluções para trabalhos futuros visando aprimorar o dispositivo desenvolvido e seu software.

Palavras-chave: Balística Forense. Confronto Balístico. Microestriamentos. Processamento de Imagens e Sinais.

Abstract

Forensic ballistics is increasingly embracing technology and its advancements, improving its results and generating new and differentiated reports that have changed the dynamics of criminalistics. Despite the continuous progress and achievements, there are several unexplored ways to leverage technology. Thus, a new concept emerges in the realm of forensic ballistics - taking ballistic comparison to the field. Many branches of criminal investigation are already operating in the field, and the benefits of such services to criminalistics are evident, encouraging the expansion of ballistics for on-site analysis.

Therefore, this work aims to pursue this idea, presenting in this report the development of a physical prototype capable of scanning projectiles and a software able to transform the cylindrical shell of a projectile into a flat one using multiple photos. The software also applies computer vision and signal processing techniques to analyze firearm marks on the projectiles. The objective of this work is to test some methodologies to reach the mentioned device and software, allowing for preliminary analysis to assist in crime scenes and expedite the identification and apprehension of criminals.

This study provides a literature review on the subject, providing readers with insights into the origin of firearms and ammunition, the functioning of forensic ballistics, and current methods for ballistic comparison. Important theoretical fundamentals are developed to ensure a proper understanding of the methodologies used, showcasing image and signal processing methods employed. The methodology is explained, and the results are presented, demonstrating that the objectives were achieved and success was achieved in creating the device, scanning the projectile, processing the images, extracting the signals, and conducting the comparison. Significant aspects of the entire process and the applied methodology are concluded, suggesting changes and advancements for future works to enhance the developed device and its software.

Keywords: Forensic Ballistics. Ballistic Comparison. Microstriae. Image and Signal Processing.

Lista de ilustrações

Figura 2.1 – Imagens das Montagens das Munições - Fonte: (NETO, C. F. P., 2021) . . .	23
Figura 2.2 – Exemplo de Canos com almas raiadas e lisas - Fonte: (TIRO, 2015)	25
Figura 2.3 – Exemplo de Cano Ligado Câmara de Expansão dos Gases - Fonte: (FREI- TAS, s.d.)	26
Figura 2.4 – Imagens do Processo de Disparo e Recuo da Culatra - Fonte: (SÁ SOARES, 2002a)	27
Figura 2.5 – Arma Glock 30 Desmontada - Fonte: (SÁ SOARES, 2002b)	27
Figura 2.6 – Pino, Culatra e Mola Recuperadora - Fonte: (SÁ SOARES, 2002a)	28
Figura 2.7 – Marcas na cápsula geradas pela culatra e pelo pino de percussão - Fonte: (PEREIRA, 2010)	29
Figura 2.8 – Diferença entre calibre do projétil e calibre do cano raiado - Fonte: (SIL- VEIRA, 2013)	30
Figura 2.9 – Marcas em projéteis deixadas por canos de alma raiada - Fonte: (FO- RENSE, 2017)	31
Figura 2.10–Exemplos de Coletores de Projéteis - Fonte: (SANTOS, L. S. dos, 2015) . .	33
Figura 2.11–Exemplos de Confrontos Balísticos - Fonte: (SANTOS, L. S. dos, 2015) . .	35
Figura 2.12–Microscópio de Confronto - Fonte: (SANTOS, L. S. dos, 2015)	36
Figura 3.13–Representação das Lentes Ocular e Objetiva em um Microscópio - Fonte: (HECHT, 2012)	38
Figura 3.14–Representações de uma Câmera	39
Figura 3.15–Demonstração da Diferença entre Campo Claro (esquerda) e Campo Escuro (Direita) - Fonte: (MURPHY, 2001)	40
Figura 3.16–Variação de Quantidade de Pixels Gerados - Fonte: (LONDON; STONE; UPTON, 2017)	42
Figura 3.17–Convenção para as Coordenadas Espaciais dos <i>pixels</i> - Fonte: (FILHO; NETO, H. V., 1999)	42
Figura 3.18–Elementos de um Sistema de Processamento de Imagens em Visão Com- putacional - Fonte: (FILHO; NETO, H. V., 1999)	45
Figura 3.19–Kernels de Mesma Área e Regiões em Escalas Diferentes - Fonte: Autor	47
Figura 3.20–Espaço de Escalas	48
Figura 3.21–Difference of Gaussian	49
Figura 3.22–Comparação do <i>pixel</i> 'x' com os 26 mais próximos - Fonte: (LOWE, 2004)	50
Figura 3.23–Orientação e Magnitude dos <i>Keypoints</i> - Fonte: (SINGH, A., 2019)	51
Figura 3.24–Criação de Descritores - Fonte: (SINGH, A., 2019)	52
Figura 3.25–Criação de Imagem Integral - Fonte: (SZELISKI, 2011)	53

Figura 3.26–Criação de Diferentes Escalas de Máscaras Gaussianas - Fonte: (BAY; ESS et al., 2008)	54
Figura 3.27–Wavelet de Vizinhança de Raio 6s - Fonte: (BAY; ESS et al., 2008)	55
Figura 3.28–Quadrados de lado 20s Orientados Pelos Pontos de Interesse - Fonte: (BAY; TUYTELAARS; VAN GOOL, 2006)	55
Figura 3.29–Círculo de Bresenham com $R = 3$ e $K = 16$ - Fonte: (LEE; FULLER; MCCOWN, 2020)	56
Figura 3.30–Espaço de Escala BRISK - Fonte: (AZIMI et al., 2019)	60
Figura 3.31–Espaço de Escala BRISK com Respectiva Parábola Encontrada - Fonte: (LEUTENEGGER; CHLI; SIEGWART, 2011)	61
Figura 3.32–Padrões Utilizados para Determinar os Descritores - Fonte: (FAN; WANG; WU et al., 2015)	62
Figura 3.33–Mapeamento entre Planos - Fonte: (HARTLEY; ZISSERMAN, 2003)	64
Figura 3.34–Distância Euclidiana - Fonte: (O’NEILL, 2006)	68
Figura 3.35–Exemplo Distância de Hamming Igual a 3 - Fonte: (ISHENGOMA, 2014)	69
Figura 3.36–Etapas do Processamento de Sinais - Fonte: (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006)	70
Figura 4.37–Imagens da <i>WebCam</i> Escolhida	76
Figura 4.38–Imagens do Microscópio Escolhido - Marca PalyShop Eletrônicos e Modelo Play16 - Fonte: Autor	78
Figura 4.39–Motor de Passo e sua placa de driver - Fonte: Autor	79
Figura 4.40–Arduíno UNO - Fonte: Autor	80
Figura 4.41–Suporte do Motor de Passo em Algumas Vistas e seu Modelo 3D - Fonte: Autor	82
Figura 4.42–Mandril Adquirido e seu Modelo 3D - Fonte: Autor	83
Figura 4.43–Suporte do Rolamento em Algumas Vistas e seu Modelo 3D - Fonte: Autor	84
Figura 4.44–Suporte de Conexão entre Mandril, Rolamento e Motor e seu Modelo 3D - Fonte: Autor	84
Figura 4.45–Estrutura Montada para Rotar o Projétil - Fonte: Autor	85
Figura 4.46–Suporte do Microscópio e seu Modelo 3D - Fonte: Autor	86
Figura 4.47–Trilho para ajustar a Distância do Microscópio ao Projétil - Fonte: Autor	86
Figura 4.48–Suporte do Microscópio e seu Modelo 3D - Fonte: Autor	87
Figura 4.49–Estrutura Montada para Posicionar o Microscópio - Fonte: Autor	87
Figura 4.50–Suporte para LEDs para o Microscópio e seu Modelo 3D - Fonte: Autor	88
Figura 4.51–Capa e Difusor para o Suporte de LEDs do Microscópio e seu Modelo 3D - Fonte: Autor	89
Figura 4.52–Suporte para LEDs para a Lateral e seu Modelo 3D - Fonte: Autor	90
Figura 4.53–Capa para o Suporte de LEDs da Lateral e seu Modelo 3D - Fonte: Autor	90
Figura 4.54–Estrutura Completa Montada - Fonte: Autor	91

Figura 4.55–Seção de Visibilidade do Projétil com Medidas - Fonte: Autor	92
Figura 4.56–Duas Imagens com Pontos em Comum - Fonte: Autor	94
Figura 4.57–Corte e União das Imagens - Fonte: Autor	95
Figura 4.58–Área de Comparação Efetiva - Fonte: Autor	96
Figura 4.59–Quatro Pontos Escolhidos Para a Homografia - Fonte: Autor	97
Figura 4.60–Ângulo de Inclinação da Ranhura de Um Projétil - Fonte: Autor	99
Figura 4.61–Deslocamento Horizontal de 1 <i>pixel</i> Após o Deslocamento Vertical de 10 <i>Pixels</i> - Fonte Autor	100
Figura 4.62–Distribuição Em Função de Densidade de Probabilidade	102
Figura 5.63–Projétil e sua imagem capturada pelo microscópio montado - Fonte: Autor	103
Figura 5.64–Peça Preta e sua imagem capturada pelo microscópio montado - Fonte: Autor	104
Figura 5.65–Lima, Lima com zoom e sua imagem capturada pelo microscópio montado - Fonte: Autor	105
Figura 5.66–Projétil e Primeira Imagem Capturada com o Microscópio - Fonte Autor	106
Figura 5.67–Projétil Utilizado para Testes e Ajustes - Fonte: Autor	106
Figura 5.68–Teste de Iluminação - Fonte Autor	107
Figura 5.69–Teste do Dispositivo Completo - Fonte Autor	108
Figura 5.70–Teste do Método SIFT - Fonte Autor	109
Figura 5.71–Teste do Método SURF - Fonte Autor	110
Figura 5.72–Teste do Método ORB - Fonte Autor	111
Figura 5.73–Teste do Método BRISK - Fonte Autor	111
Figura 5.74–Exemplo da Homografia Realizada - Fonte Autor	112
Figura 5.75–Exemplo de uma Montagem da Face Lateral Planificada de um Projétil - Fonte Autor	113
Figura 5.76–Comparação Sinal da Linha 90 e Sinal da Média entre as Linhas de 90 a 99 - Fonte: Autor	114
Figura 5.77–Autocorrelação do Sinal Obtido a partir do Projétil 9mm - Fonte: Autor .	114
Figura 5.78–Resultado da Translação Vertical do Sinal - Fonte Autor	115
Figura 5.79–Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 2 da Arma 1 - Fonte: Autor	116
Figura 5.80–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 1 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	116
Figura 5.81–Correlação Cruzada do Projétil 1 da Arma 2 com o Projétil 2 da Arma 2 - Fonte: Autor	117
Figura 5.82–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 2 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	117
Figura 5.83–Correlação Cruzada do Projétil 1 da Arma 3 com o Projétil 2 da Arma 3	118

Figura 5.84–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 3 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	118
Figura 5.85–Correlação Cruzada do Projétil 1 da Arma 4 com o Projétil 2 da Arma 4 - Fonte: Autor	119
Figura 5.86–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 4 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	119
Figura 5.87–Correlação Cruzada do Projétil 1 da Arma 5 com o Projétil 2 da Arma 5 - Fonte: Autor	120
Figura 5.88–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 5 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	120
Figura 5.89–Correlação Cruzada do Projétil 1 da Arma 6 com o Projétil 2 da Arma 6 - Fonte: Autor	121
Figura 5.90–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 6 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	121
Figura 5.91–Correlação Cruzada do Projétil 1 da Arma 7 com o Projétil 2 da Arma 7 - Fonte: Autor	122
Figura 5.92–Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 7 - Fonte: Modificado de (SILVINO JUNIOR, 2010)	122
Figura 5.93–Alinhamento Errado Semi-positivo - Fonte: Modificado de (SILVINO JUNIOR, 2010)	123
Figura 5.94–Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 2 da Arma 2 - Fonte: Autor	124
Figura 5.95–Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 1 da Arma 7 - Fonte: Autor	124
Figura 5.96–Correlação Cruzada do Projétil 1 da Arma 7 com o Projétil 2 da Arma 5	125
Figura 5.97–Correlação Cruzada do Projétil 2 da Arma 4 com o Projétil 1 da Arma 6 - Fonte: Autor	125
Figura 5.98–Correlação Cruzada do Projétil 2 da Arma 2 com o Projétil 2 da Arma 6 - Fonte: Autor	125
Figura 5.99–Correlação Cruzada do Projétil 2 da Arma 3 com o Projétil 2 da Arma 4 - Fonte: Autor	126
Figura 5.100–Correlação Cruzada do Projétil 2 da Arma 3 com o Projétil 2 da Arma 5 - Fonte: Autor	126
Figura 5.101–Distância Euclidiana - Fonte: Autor	127
Figura 5.102–Distância de Manhattan - Fonte: Autor	128
Figura 5.103–Distâncias de Minkowski - Fonte Autor	129
Figura 5.103–Distâncias de Minkowski (continuação) - Fonte Autor	130
Figura A.104–Desenho Técnico da Base do Motor de Passo	143
Figura A.105–Desenho Técnico do Suporte do Rolamento	144

Figura A.106	Desenho Técnico do Encaixe entre Eixo do Motor, Rolamento e Parafuso do Mandril	145
Figura A.107	Desenho Técnico do Suporte do Microscópio	146
Figura A.108	Desenho Técnico da Base do Microscópio e da Iluminação Lateral . . .	147
Figura A.109	Desenho Técnico do Trilho para o Microscópio	148
Figura A.110	Desenho Técnico do Encaixe do Trilho do Microscópio	149
Figura A.111	Desenho Técnico da Base do Suporte de LEDs Lateral	150
Figura A.112	Desenho Técnico da Placa de LEDs das Laterais	151
Figura A.113	Desenho Técnico da Capa e Suporte de Difusor para Placa de LEDs das Laterais	152
Figura A.114	Desenho Técnico da Placa de LEDs para o Microscópio	153
Figura A.115	Desenho Técnico da Capa Traseira da Placa de LEDs do Microscópio . .	154
Figura A.116	Desenho Técnico do Suporte de Difusor para Placa de LEDs do Microscópio	155
Figura A.117	Projéteis da Arma 1 - Fonte: (SILVINO JUNIOR, 2010)	180
Figura A.118	Projéteis da Arma 2 - Fonte: (SILVINO JUNIOR, 2010)	181
Figura A.119	Projéteis da Arma 3 - Fonte: (SILVINO JUNIOR, 2010)	181
Figura A.120	Projéteis da Arma 4 - Fonte: (SILVINO JUNIOR, 2010)	182
Figura A.121	Projéteis da Arma 5 - Fonte: (SILVINO JUNIOR, 2010)	182
Figura A.122	Projéteis da Arma 6 - Fonte: (SILVINO JUNIOR, 2010)	183
Figura A.123	Projéteis da Arma 7 - Fonte: (SILVINO JUNIOR, 2010)	183

Lista de abreviaturas e siglas

BRIEF	Binary Robust Independent Elementary Features	46
BRISK	Binary Robust Invariant Scalable Keypoints	46
CMYK	Cyan Magenta Yellow Key/Black	43
DoG	Difference of Gaussian	48
FAC	Função de Autocorrelação - <i>Autocorrelation Function</i>	72
FAST	Features from Accelerated Segment Test	46
FCC	Função de Correlação Cruzada - <i>Cross-Correlation Function</i>	72
HSV	Hue Saturation Value	43
IDE	Integrated Development Environment	92
ORB	Oriented FAST and Rotated BRIEF	46
RGB	Red Green Blue	43
SIFT	Scale-Invariant Feature Transform	46
SURF	Speeded-Up Robust Features	46

Sumário

1	Introdução	18
1.1	Motivação	18
1.2	Definição do problema	18
1.3	Objetivos do projeto	19
1.3.1	Objetivo Geral	19
1.3.2	Objetivo Específico	19
1.4	Estrutura do Trabalho	20
2	Revisão Bibliográfica	21
2.1	Armas e Munições	21
2.1.1	Partes de uma Munição	22
2.1.2	Partes de uma Arma	24
2.2	Marcas Resultantes de Disparos	28
2.2.1	Marcas no Estojo	28
2.2.2	Marcas no Projétil	29
2.3	Balística Forense	31
2.3.1	Recolhimento de Materiais	32
2.3.2	Perícia	32
2.3.3	Confronto Balístico	34
3	Fundamentação Teórica	37
3.1	Microscopia e Lentes	37
3.1.1	Câmeras Fotográficas	38
3.1.2	Campos Claros e Campos Escuros	39
3.2	A Imagem no Âmbito Digital	41
3.3	Visão Computacional	43
3.3.1	Processamento de Imagens	44
3.3.2	Medidas de Similaridade e Distâncias para Comparação de Características	67
3.3.3	Processamento de Sinais	70
4	Metodologia	75
4.1	Dispositivo	75
4.1.1	Captura da Imagem	75
4.1.2	Estrutura	81
4.2	Software	91

4.2.1	Acionamento do Motor e Captura de Imagens	91
4.2.2	União das Imagens	93
4.2.3	Extração de Sinais para Comparação	98
4.2.4	Comparação de Sinais	100
5	Resultados	103
5.1	Captura das Imagens	103
5.1.1	Câmara - <i>WebCam</i>	103
5.1.2	Microscópio	105
5.2	Processamento de Imagens	109
5.2.1	União das Imagens	109
5.3	Processamento de Sinais	113
5.3.1	Autocorrelação e Correlação Cruzada	114
5.3.2	Comparação Estatística	127
6	Considerações Finais	132
6.1	Conclusões e Discussões	132
6.2	Perspectivas para Trabalhos Futuros	133
	Referências	135
	Apêndices	141
	Apêndice A Desenhos Técnicos	142
	Apêndice B Códigos de programação	156
B.1	Código para Movimentação do Motor e Captura das Imagens	156
B.2	Código para Testar os Métodos Descritores	158
B.3	Código para União das Imagens	160
B.4	Código para Extração de Sinal das Imagens Montadas	169
B.5	Código que Realiza a Autocorrelação e Correlação Cruzada	172
B.6	Código que Realiza a Comparação Estatística	175
	Anexos	179
	Anexo A Imagens Externas Utilizadas	180
A.1	Imagens de Projéteis Utilizadas Para Análise	180

1 Introdução

1.1 Motivação

A utilização de tecnologias no trabalho da criminalística está se tornando cada vez mais fundamental, tem trazido grandes melhorias no trabalho policial. Testes, investigações e resultados que pouco tempo atrás eram difíceis ou até impossíveis de se realizar, com determinados avanços e criações tecnológicas passaram a ser factíveis.

Atualmente, a criminalística tem incrementado seus laboratórios com diversas tecnologias, porém o campo ainda se encontra relativamente fraco nesse aspecto, nem todas as áreas da perícia apresentam tecnologia para trabalhos em campo. A dependência completa dos laboratórios torna o trabalho de campo mais lento e menos eficaz, o que prejudica o trabalho da criminalística em conclusão.

Com esse pensamento, decidiu-se estudar, projetar e desenvolver um dispositivo para auxiliar o trabalho em campo a diminuir sua dependência do trabalho laboratorial. Como dispositivo, escolheu-se seguir a área da balística forense e projetar um comparador balístico portátil que possibilitaria uma análise preliminar de projéteis ainda na cena do crime, facilitando e direcionando a investigação criminal em campo.

1.2 Definição do problema

Um comparador balístico, em suma, é um dispositivo que realiza comparações entre projéteis e explicita a semelhança física entre as ranhuras deixadas em cada projétil, podendo diagnosticar, com alta probabilidade, de qual arma aquele dado projétil foi disparado.

Laboratórios criminais, atualmente, utilizam comparadores balísticos em seus setores de balística, porém esses comparadores necessitam estar dentro dos laboratórios já que muitas vezes são de grande porte, pesados ou necessitam de alguma estrutura do laboratório para funcionar.

Um comparador balístico portátil, para ser funcional e não depender de um laboratório criminal, precisa apresentar algumas características específicas. É necessário que o comparador seja de pequeno porte (o menor possível), que não necessite de conexão com tomadas, que consiga capturar imagens completas da face lateral dos projéteis, que seja capaz de analisar as imagens e realizar micro-comparações com qualidade o suficiente para gerar confiança em seus resultados, mesmo necessitando de análise laboratorial posterior.

Com as características definidas, primeiramente será necessário projetar, montar e

testar a parte física do dispositivo, o que inclui o mecanismo no qual o projétil será acoplado e o mecanismo de captura de imagens. Em segunda fase, será necessário desenvolver um programa computacional (software) que controle os mecanismos da parte física e que também seja capaz de extrair informações das imagens e utilizá-las para realização das comparações balísticas.

1.3 Objetivos do projeto

1.3.1 Objetivo Geral

O presente trabalho traz como objetivo geral estudar a comparação balística, criar um projeto e o colocar em prática desenvolvendo um protótipo de um dispositivo portátil com a capacidade de, em campo, realizar análises e micro-comparações balísticas de projéteis de forma automática. Ressalta-se que o atual trabalho será um início dos estudos para o desenvolvimento do comparador, focando no desenvolvimento de um protótipo, esperando-se que ao fim do projeto seja possível avaliar as metodologias propostas e definir a viabilidade do projeto proposto em conjunto com os seus objetivos e pré-requisitos específicos.

1.3.2 Objetivo Específico

Como objetivos específicos, evidenciam-se características do desenvolvimento do dispositivo portátil e o que se deseja alcançar no presente trabalho.

Para esse trabalho, deseja-se atingir os seguintes objetivos específicos:

- Desenvolver a mecânica completa do dispositivo requerido, passando por todas as etapas de teste e melhoramentos atingindo um dispositivo mecanicamente funcional;
- Desenvolver o software que realizará a captura das imagens e processamento gerando uma única imagem completa da face lateral aberta de um projétil;
- Desenvolver o software que é capaz de realizar as comparações entre dois projéteis.

Alguns pré-requisitos necessários para que estes objetivos sejam atingidos podem ser listados:

- Ser acessível, isto é, apresentando um baixo custo de construção e fácil manipulação;
- Ser invariável, ou seja, não depender de quem o utiliza ou do ambiente onde se localiza, não sendo afetado por variação de luz e afins;
- Ser confiável, isto é apresentar resultados que sejam suficientemente confiáveis em uma pré-análise.

1.4 Estrutura do Trabalho

O presente trabalho consta com 6 capítulos além de referências bibliográficas. No capítulo 1, atual capítulo, é explicada a motivação para a realização deste projeto, o problema a ser estudado e os objetivos a serem atingidos. O capítulo 2, por outro lado, trará as revisões bibliográficas realizadas, armas e munições serão detalhadas, assim como as marcas resultantes de disparos e ainda serão abordadas noções de balística. Em seguida, o capítulo 3 trará a fundamentação teórica necessária para o desenvolvimento do projeto, passando pelo estudo de lentes, câmeras, visão computacional, processamento de imagens e de sinais. Finalmente, no capítulo 4, são discriminadas as decisões tomadas quanto a montagem da parte física do dispositivo e suas evoluções até o protótipo final, além da metodologia utilizada para criação dos softwares de controle, captura de imagens e comparação. No capítulo 5, os resultados são demonstrados e explanados. Por fim, no último capítulo, capítulo 6, conclui-se o trabalho realizado e indicam-se as possíveis melhorias a serem realizadas futuramente, trazendo as perspectivas para trabalhos futuros.

2 Revisão Bibliográfica

O capítulo atual traz uma revisão bibliográfica que atuou como suporte para iniciar o desenvolvimento do projeto. São mencionados conceitos importantes e é mostrado o estado da arte referente à comparação balística forense.

2.1 Armas e Munições

Por definição, armas são tudo aquilo que possa ser utilizado para proporcionar vantagens em ataques ou defesas. São dispositivos projetados para causar danos físicos, ferir incapacitar ou matar seres vivos, além de serem capazes de destruir ou danificar objetos materiais. Existem várias categorias nas quais as armas podem ser divididas como por exemplo armas brancas, armas químicas, armas nucleares, armas de fogo ou qualquer arma improvisada. As armas que se encaixam no escopo deste projeto estão na categoria armas de fogo.

Armas de fogo são armas que utilizam a pressão gerada por gases originados na combustão de um propelente para impulsionarem projéteis através de um cano em direção ao alvo (DK PUBLISHING, 2014). Três elementos principais são necessários para uma arma ser considerada arma de fogo, sendo eles a munição, o mecanismo de disparo e a arma em si onde serão inseridos os dois anteriores (DK PUBLISHING, 2014).

Por sua vez, projéteis, ou munições, são quaisquer objetos propulsionados contra algum alvo com o mesmo objetivo das armas, atacar e/ou defender-se. Estes podem estar ou não relacionados a armas de fogo, dependendo apenas do mecanismo de disparo utilizado.

Segundo Norris (2011), não se sabe com absoluta certeza quando ou onde a pólvora foi utilizada em conjunto com um tubo para propelir um projétil. A pólvora foi o primeiro propelente utilizado amplamente em armas de fogo e sua primeira fórmula com registros escritos está contida em um tratado militar atribuído a Wu Ching Tsung Yao, ressalta Norris (2011).

Independentemente de suas origens, um dos primeiros registros conhecidos do uso da pólvora como propulsor de projéteis em uma arma ocorreu por meio de um dispositivo de bambu de cano longo, datado por volta de 1132 d.C, afirma Norris (NORRIS, 2011), explicando ainda que a invenção desse dispositivo é atribuída ao General Chinês Ch'en Gui, que comandava a guarnição de Anlu, na província de Hopei. Acredita-se que o cano tenha sido reforçado com corda ou couro para maior resistência.

Com o tempo, a tecnologia associada às armas de fogo melhorou e se espalhou para diferentes partes do mundo. Ao longo dos séculos que se seguiram, o design e a eficiência

das armas de fogo avançaram significativamente, resultando em armas mais portáteis, mais precisas e mais rápidas de carregar. Essas evoluções criaram armas e munições mais tecnológicas com diversas combinações e montagens capazes de gerar efeitos diferenciados em relação a propulsão, a letalidade, a estabilidade, entre outros.

2.1.1 Partes de uma Munição

Munições, como mencionado anteriormente, não estão necessariamente relacionadas a armas de fogo, portanto munições de armas de fogo foram objetos reaproveitados de outras categorias de armas, como por exemplo flechas e afins. Posteriormente foram introduzidas munições de metal com montagens específicas que incluem nas munições o propelente (WALKER, 2012).

A montagem das munições, em geral, é a mesma para todos os tipos de armas, salvo algumas munições com funções especiais como em espingardas e afins. Esse modelo de montagem não sofreu muitas mudanças desde a criação das munições específicas para armas de fogo (SILVA, I. O., 2022).

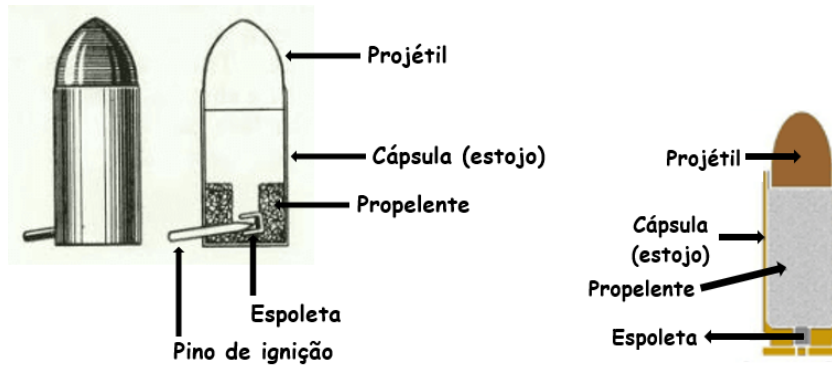
A primeira munição criada foi montada com 5 partes, estojo, pino de ignição, espoleta, propelente e projétil. Atualmente, agora com 4 partes, as munições são compostas por estojo, espoleta, propelente e projétil (OLIVEIRA JÚNIOR, 2015).

2.1.1.1 Estojo

O estojo, ou cápsula, de uma munição é a parte mais externa que tem a função de agregar as outras partes em suas posições específicas. Essa peça da munição existe desde a primeira munição criada para armas de fogo e tem a mesma função desde então (SILVA, I. O., 2022).

Nas primeiras munições, a cápsula era feita de papelão, o que com o tempo foi modificado devido a sua resistência, eficiência e segurança. Atualmente a maioria das cápsulas são feitas de metal, mais comumente de latão (SANTOS, L. S. d., 2015).

Na figura 2.1 é possível visualizar a posição em que se encontrava a cápsula na primeira munição e onde ela se encontra nas munições atuais. Além disso, é possível reparar na mudança do formato da base da cápsula, o que foi criado para facilitar o encaixe da munição na posição de disparo.



(a) Montagem da Primeira Munição

(b) Montagem da Munição Atual

Figura 2.1 – Imagens das Montagens das Munições - Fonte: (NETO, C. F. P., 2021)

2.1.1.2 Propelente

O propelente é uma espécie de combustível que é responsável pela expulsão do projétil da munição. O propelente se encontra dentro da cápsula e abaixo do projétil, em contato direto com ele.

Em munições são utilizados propelentes sólidos, em grãos (SILVA, I. O., 2022). Alguns exemplos de propelentes em grãos são a pólvora, a nitrocelulose ou combustíveis especiais feitos de perclorato de amônio ou borracha sintética. Para armas de fogo, desde a primeira munição, é utilizada a pólvora. Antigamente não se sabia as propriedades de propelentes e a pólvora foi o primeiro propelente a ser criado, originalmente para fogos de artifício. Atualmente a pólvora ainda é a principal escolha de propelente para se utilizar em armas de fogo (AMAZONAS, 2012).

Pela figura 2.1 consegue-se ver a posição em que se encontrava a pólvora na primeira munição e onde ela se encontra nas munições atuais.

2.1.1.3 Projétil

O projétil consiste na parte da munição que é expelido e impulsionado contra o alvo. Para que a munição seja eficaz, o projétil deve ser preso a cápsula de forma justa para que aguente por determinado tempo a pressão que será gerada pelos gases liberados pelo propelente e para que não deixe escapar os gases antes de sua expulsão (SANTOS, L. S. d., 2015).

Nas primeiras munições os projéteis eram esferas que eram feitas de chumbo maciço ou chumbo associado a algum outro material para formar ligas mais duras ou mais moles (SILVA, I. O., 2022). O uso do chumbo se devia a facilidade da manipulação dele, como seu ponto de fusão é baixo, facilmente era fundido e moldado (SILVA, I. O., 2022).

Atualmente, projéteis de chumbo ainda são os mais utilizados, porém com ressalvas. Como o chumbo tem seu ponto de fusão baixo, dependendo da velocidade em que o projétil

é expelido da arma, é possível, mesmo não chegando a temperatura de fusão, que o atrito faça com que partículas de chumbo fiquem para trás encrustadas nas raiais da arma e que o projétil deslize sem acompanhar as raiais, fazendo com que o projétil expelido perca potência e estabilidade, além de afetar o próximo disparo. Para corrigir esse problema, além de se utilizar projéteis de outros materiais, utiliza-se também projéteis de chumbo folheados com algum outro material como o cobre, para que o contato com a parede interna do cano não seja direta com o chumbo (SILVA, I. O., 2022).

A figura 2.1 mostra a posição em que se encontrava o projétil na primeira munição e onde ele se encontra nas munições atuais.

2.1.1.4 Espoleta

A espoleta, que se encontra na base da munição, é a parte responsável por gerar uma reação química que provocará a ignição do propelente, gerando gases que, por meio de pressão, expulsaram o projétil (SILVA, I. O., 2022).

Nas primeiras munições, a espoleta vinha acompanhada de um pino de ignição que, ao receber um choque mecânico externo, pressionava a espoleta e a forçava a iniciar a reação química. Como o pino gerava uma certa protuberância externa, eram comuns disparos acidentais acontecerem, uma pressão involuntária poderia acionar a espoleta até mesmo com a munição em mãos, sem necessitar de uma arma. Dessa forma, atualmente, o pino de ignição, chamado hoje de pino de percussão, é localizado na arma e apresenta a mesma função, mas exigindo uma arma para que a espoleta seja acionada (SILVA, I. O., 2022).

É possível visualizar na figura 2.1 a posição em que se encontrava a espoleta na primeira munição e onde ela se encontra nas munições atuais. Além disso, é possível reparar na mudança da forma como ela é ativada, sendo pela lateral da cápsula antiga e pela parte inferior da cápsula atual.

2.1.2 Partes de uma Arma

As partes de uma arma são de extrema importância para a concretização do objetivo que se tem para ela, seja alvos a longa distância, a curta distância, pequeno estrago, grande estrado e diversos outros.

Nesse capítulo serão estudadas as principais partes das armas de fogo presentes em quase todos os tipos de armas, as partes necessárias para que o efeito de propulsão seja atingido. Ainda, serão estudadas, principalmente, armas e projéteis de pequeno porte e os efeitos dos disparos nos projéteis.

2.1.2.1 Cano

O cano de uma arma de fogo é de extrema importância, de armas de grande porte a armas de pequeno porte, são os canos que guiam o projétil para fora da arma e direcionam sua propulsão e alvo.

Com o passar do tempo, os canos das armas passaram de apenas direcionadores para peças responsáveis por rotacionar o projétil e fazer com que ele mantenha sua direção e perca menos energia para o meio, atingindo com mais potência seu alvo (SILVA, I. O., 2022).

Um cano de arma é originalmente um cilindro maciço que é perfurado de uma face circular a outra. A ferramenta de furação deixa marcas expressivas na peça, marcas essas que são minimizadas ao máximo antes de criar-se o raiamento interno. Mesmo minimizando as marcas, algumas marcas únicas ainda ficam na parte interna do cano, as quais posteriormente gerarão marcas únicas no projétil. Em seguida são criados os raiamentos internos por meio da utilização de um torno. Algumas armas não apresentam raiamentos internos em seus canos, o que se chama de armas de alma lisa, enquanto as outras são chamadas de armas de alma raiada (SANTOS, L. S. d., 2015).

A alma de uma arma, ou de um cano de arma, é a parte interna do cano da arma, a parte por onde o projétil passará durante sua ejeção.

O calibre do cano utilizado é dependente de sua alma. Canos de alma lisa tem o calibre do mesmo tamanho que seus projéteis, mas canos de alma raiada não. Em canos de alma raiada, o diâmetro do projétil é igual ao diâmetro das raias (diâmetro D2 em azul na figura 2.2) e o restante do cano apresenta diâmetro menor (diâmetro D1 em vermelho na figura 2.2), forçando com que o projétil siga o movimento das raias (SANTOS, L. S. d., 2015).

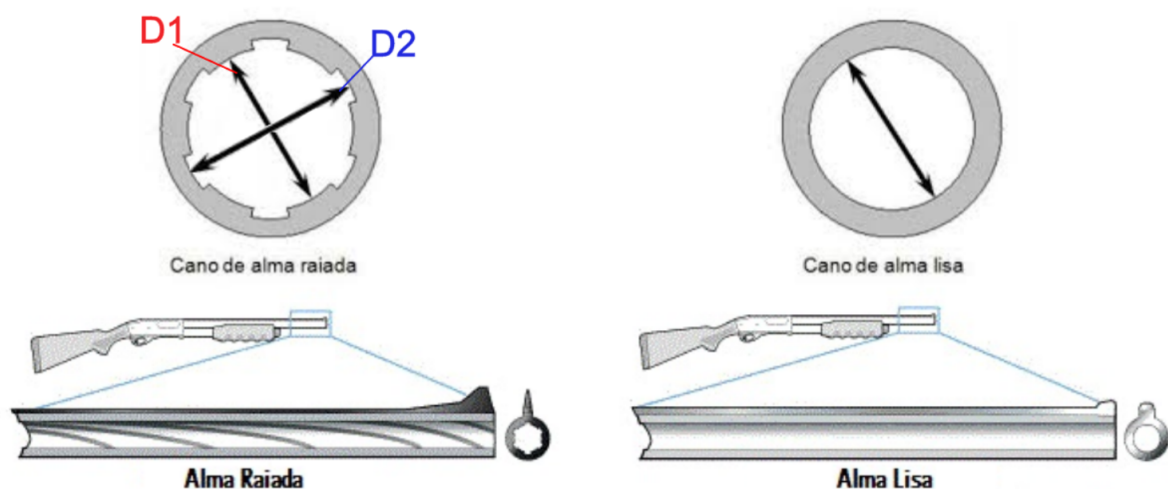


Figura 2.2 – Exemplo de Canos com almas raiadas e lisas - Fonte: (TIRO, 2015)

2.1.2.2 Câmara de expansão dos gases

A câmara de expansão dos gases, como o nome sugere, é um espaço presente na arma destinado a aguardar e a resistir à expansão dos gases que acontecerá internamente a cápsula. Essa câmara necessita de certa resistência para que a cápsula permaneça dentro da arma conectada ao cano para que o projétil seja expelido após a queima da pólvora.

Também chamada de câmara de explosão, a câmara de expansão se encontra ligada ao início do cano de forma que a cápsula não passe pelo cano, apenas o projétil, possibilitando o posterior descarte da cápsula (SILVA, I. O., 2022). Na figura 2.3 é possível visualizar a relação do cano com a câmara de expansão, ou explosão. Essa estrutura pode ser alterada em armas de maior porte.



Figura 2.3 – Exemplo de Cano Ligado Câmara de Expansão dos Gases - Fonte: (FREITAS, s.d.)

2.1.2.3 Culatra

A culatra é uma peça da arma que está associada com a mola recuperadora e com o pino de percussão. Se encontra atrás da câmara de expansão dos gases e é responsável por vedar a região onde ocorrerá a expansão (CARLUCCI; JACOBSON, 2018).

Em armas automáticas e semiautomáticas, a culatra, junto com a mola recuperadora, são responsáveis por ejetar a cápsula após o disparo e imediatamente encaixam a nova munição na câmara de expansão (PAULY, R. A.; PAULY, R., 2004). Quando há um disparo, pela terceira lei de Newton, a ação de expulsar o projétil para frente gera a reação de empurrar a culatra para trás, comprimindo a mola de recuperação. Esse curto momento de compressão da mola é suficiente para que a cápsula saia e a nova munição entre. Por fim, a mola que acabara de acumular energia elástica gasta essa energia se expandindo novamente e reposicionando a culatra em seu lugar inicial.

Uma sequência de imagens localizadas na figura 2.4 mostram o processo descrito.

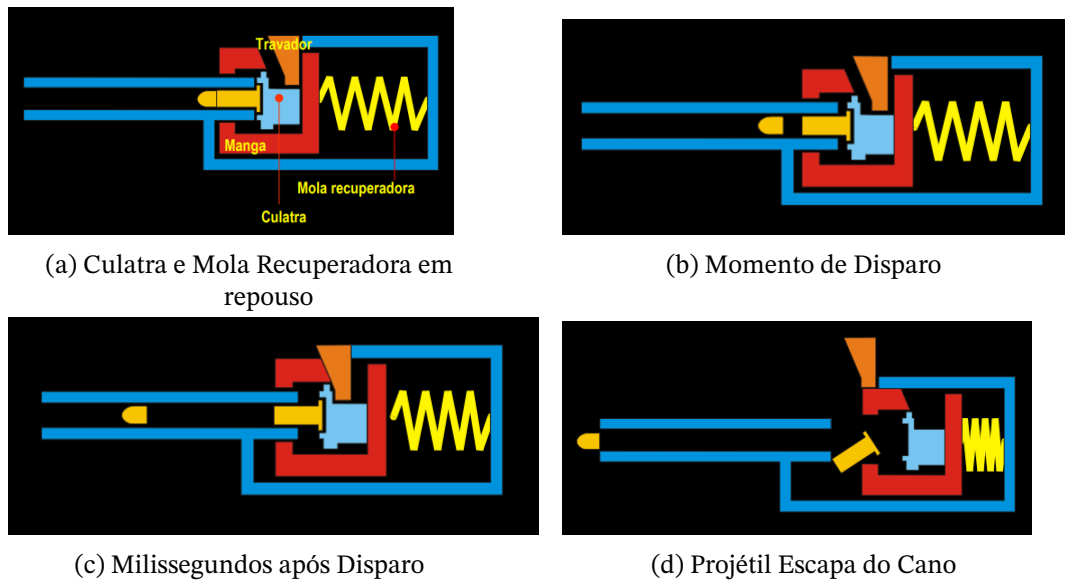


Figura 2.4 – Imagens do Processo de Disparo e Recuo da Culatra - Fonte: (SÁ SOARES, 2002a)

2.1.2.4 Ferrolho

O ferrolho é a parte da arma no qual ficam acopladas a culatra, a mola de recuperação e o pino de percussão. Inicialmente, o ferrolho foi criado para possibilitar o recarregamento no armamento, puxava-se o ferrolho que arrastava o conjunto completo para retirar a cápsula da munição recém disparada e para colocar a próxima munição (FIREARMS:..., 2014).

Atualmente, com o avanço da tecnologia bélica, em muitas armas não há a necessidade de movimentação manual do ferrolho, essa movimentação, como mencionada anteriormente, é realizada a partir do aproveitamento da energia liberada no disparo respeitando a terceira lei de newton. Ainda assim, a movimentação manual é possível e é importante para o caso de necessidade de retirada da munição da câmara de expansão sem que haja um disparo.

A figura 2.5 mostra uma Glock 30 desmontada sendo possível visualizar o ferrolho.



Figura 2.5 – Arma Glock 30 Desmontada - Fonte: (SÁ SOARES, 2002b)

2.1.2.5 Pino de Percussão

O pino de percussão, ou percutor, como já mencionado, está localizado juntamente com a culatra e com a mola de recuperação, associado ao ferrolho.

Essa peça tem a função de gerar a ignição do propelente (substância inflamável), gerando sua queima e a produção de gases que serão responsáveis pela expulsão do projétil. Para que ocorra essa ignição, o pino é acelerado em direção a cápsula do projétil, e se choça contra a espoleta, gerando uma reação química seguida da queima do propelente e do disparo do projétil (OLIVEIRA NETO, 2004).

Na figura 2.6 pode-se visualizar o pino de percussão e seu encaixe na culatra, ganhando acesso a cápsula da munição.

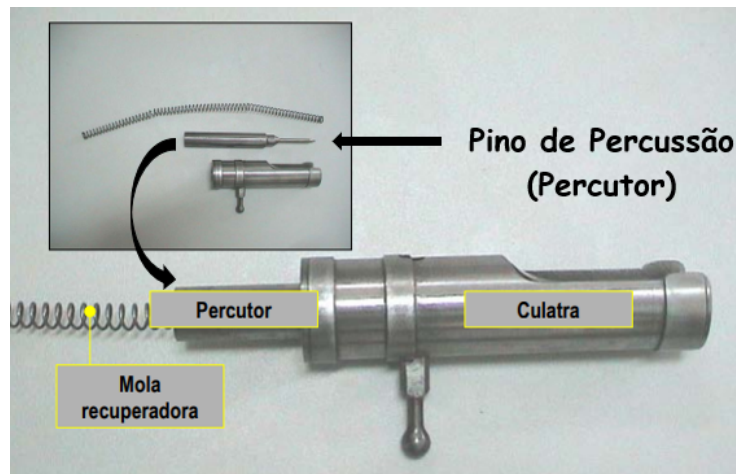


Figura 2.6 – Pino, Culatra e Mola Recuperadora - Fonte: (SÁ SOARES, 2002a)

2.2 Marcas Resultantes de Disparos

Um disparo de uma arma de fogo envolve uma grande quantidade de energia sendo dissipada em diversas formas. Como comentado anteriormente, o disparo envolve a ignição de um propelente que ao queimar gera gases que vão criar uma grande pressão a qual expulsará o projétil e gerará o recuo da cápsula. Sendo assim, tanto a cápsula quando o projétil acabam recebendo marcas no processo.

2.2.1 Marcas no Estojo

As marcas no estojo, ou cápsula, são geradas principalmente pelo contato com o pino de percussão, com a culatra, com o ferrolho e com a câmara de expansão dos gases (SANTOS, L. S. d., 2015).

As marcas geradas pela câmara de expansão dos gases são localizadas na face lateral da cápsula, são marcas de arraste do momento em que a cápsula recua dentro arma. Já as

marcas do ferrolho são marcas geradas na expulsão da cápsula, seja essa expulsão automática ou manual. Quando a cápsula é expulsa, ela se choca com parte do ferrolho enquanto passa pela cavidade existente em sua lateral (SANTOS, L. S. d., 2015). Essa duas marcas são as menos expressivas das quatro principais citadas.

A culatra, por sua vez, gerará uma marca de impacto na base da cápsula resultante do recuo realizado. Essa marca será a mais expressiva na cápsula propriamente dita, visto que a força com a qual a cápsula recua deve ser equivalente a força com a qual o projétil é expelido (SANTOS, L. S. d., 2015).

Por último, a marca deixada pelo pino de percussão. Essa marca é registrada na espoleta, porém é considerada uma marca na cápsula devido ao fato de a espoleta estar fixada na cápsula. O pino de percussão necessita de uma certa força para acionar a espoleta e, ainda, a espoleta apresenta um rigidez menor que a cápsula, logo a marca resultante do impacto do pino na espoleta acaba sendo a marca mais expressiva de todas (OLIVEIRA NETO, 2004).

Sendo a marca mais expressiva de todas, a marca deixada pelo pino de percussão é mais facilmente distinguível entre modelos de pinos em comparação com as marcas das outras peças. Sendo assim, apenas com a marca resultante na espoleta é possível ter alguma noção de qual arma realizou o disparo daquela munição. Na figura 2.7 é possível ver as marcas geradas pela culatra e pelo pino de percussão na cápsula de uma munição.

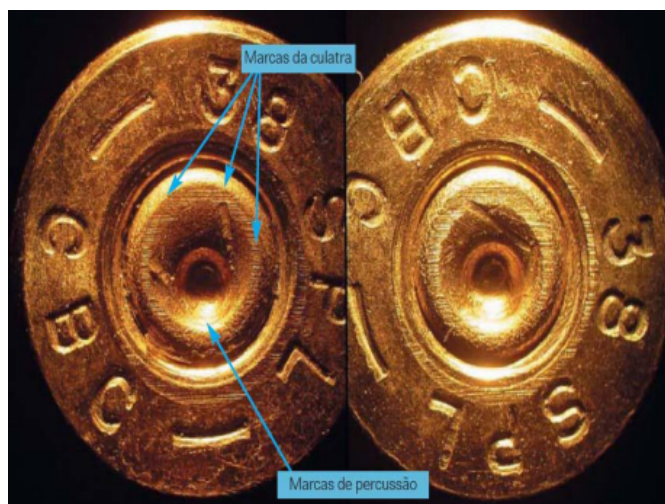


Figura 2.7 – Marcas na cápsula geradas pela culatra e pelo pino de percussão - Fonte: (PEREIRA, 2010)

2.2.2 Marcas no Projétil

As marcas geradas no projétil são resultado do contato com o cano da arma (SANTOS, L. S. d., 2015). Como abordado anteriormente, há dois tipos principais de canos de arma, canos com alma lisa e canos com almas raiadas. Os canos com alma lisa, que não registram marcas muito expressivas, são mais utilizados em armas que disparam múltiplos projéteis,

como espingardas por exemplo, que podem, em uma única munição, disparar dezenas de pequenas esferas. Sendo assim, o foco das marcas é nas armas de cano raiado.

Nos canos raiados, como já mencionado, os raiamentos apresentam o calibre próximo do projétil, enquanto o calibre real do cano é menor (figura 2.8). Por esse motivo, os projéteis são forçados a passarem pelo cano raiado gerando deformações em sua face lateral e os obrigando a rotacionar em torno de seus próprios eixos (SANTOS, L. S. d., 2015). A figura 2.8 mostra 4 raias no cano desenhado, porém o número de raias é arbitrário, depende apenas da fabricante da arma.

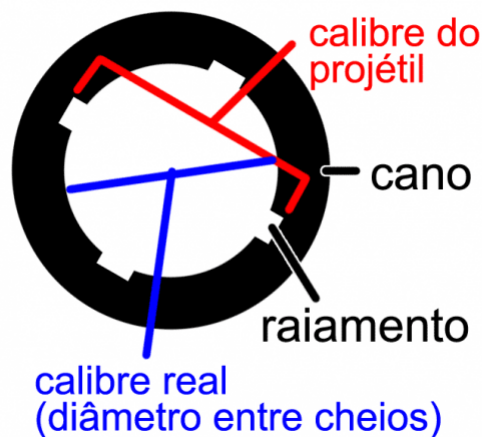


Figura 2.8 – Diferença entre calibre do projétil e calibre do cano raiado - Fonte: (SILVEIRA, 2013)

A pressão dentro do cano é bem expressiva, as marcas geradas são bem evidentes e bem distintas. Além das marcas geradas pelos raiamentos do cano, ainda há marcas que as paredes do cano têm e que são impressas no projétil. Essas marcas são devidas as ferramentas utilizadas na produção do cano (SANTOS, L. S. d., 2015) e também aos desgastes causados pelos disparos, ou seja, essas marcas são singulares, únicas, e tão únicas que e nem dois projéteis disparados pela mesma arma, um em sequência do outro, são 100% compatíveis.

Mesmo existindo diferença entre projéteis disparados pela mesma arma, as semelhanças atingem um nível suficiente para se poder dizer que ambos os projéteis foram expelidos pela mesma arma. Essas características são utilizadas para que se possa identificar de qual arma foram disparados certos projéteis.

A figura 2.9 mostra projéteis nos quais é possível visualizar as marcas deixadas por um cano de alma raiada.

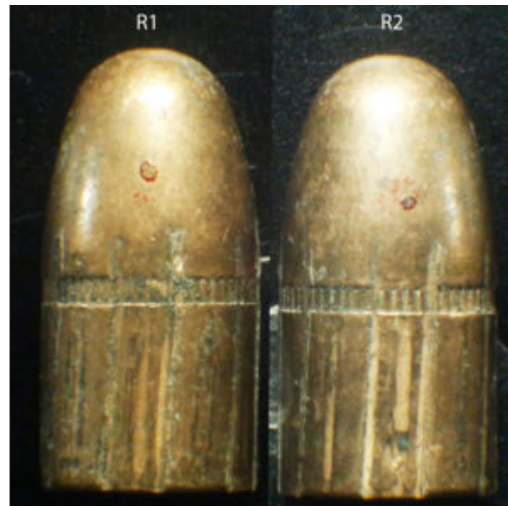


Figura 2.9 – Marcas em projéteis deixadas por canos de alma raiada - Fonte: (FORENSE, 2017)

2.3 Balística Forense

Para Heard (HEARD, 2011), a balística pode ser dividida em três tipos, a balística interna, a balística externa e a balística terminal. Essas três grandes áreas são muito estudadas com o objetivo de elucidação de crimes que envolvam armas, com ou sem vítimas.

A balística interna é definida, por Heard, como o estudo de tudo aquilo que acontece dentro de uma arma desde o momento em que o pino de percussão atinge a espoleta até o instante em que o projétil deixa o cano. Dessa forma, o estudo é focado em assuntos como a aceleração do projétil, o movimento de rotação forçado pelos raiamentos, a pressão gerada pelo propelente, o recuo e expulsão da cápsula, a temperatura interna, a velocidade de ignição do propelente com relação ao momento do choque do pino de percussão com a espoleta, as marcas geradas na cápsula e no projétil, entre outros.

Como consequência da balística interna, tem-se a balística externa, que estuda tudo aquilo que acontece com o projétil desde o momento em que ele sai do cano da arma até o instante em que ele atinge o alvo. Essa parte da balística é complexa visto que não depende só da arma e do projétil, depende fortemente do meio ambiente, da temperatura, do vento, da chuva (caso haja), da pressão atmosférica, da densidade do material do projétil, de sua forma, se foi um disparo dentro de algum fluido que não o ar, dentro de um ambiente em movimento, entre outros.

Por último, a última área da balística que é consequência das outras, a balística terminal estuda tudo aquilo que acontece a partir do momento em que o projétil acerta o alvo, o comportamento do projétil a partir desse momento. Assim como na balística interna, aqui pode ser estudada a ação do projétil em algum fluido que não o ar, a diferença nesse caso está no fato de que a balística interna estuda o caso em que o projétil é expelido pela arma já dentro do fluido, enquanto para a balística terminal o interesse está no momento

em que o projétil entra no fluido tendo percorrido o ar primeiramente. Outros meios são estudados como entrada de projéteis em madeira, argila, terra, concreto, tijolo e também em carne e ossos.

O foco desse trabalho está na balística interna, no estudo dos efeitos que o disparo exerce no projétil ainda dentro da arma de forma a deixar marcações em sua face lateral.

Por mais que exista essa separação em três áreas, só é possível estudar cada uma delas após a realização prática das três em conjunto, ou seja, só após o disparo de uma arma. Em geral, os casos estudados são casos controlados, onde testes são feitos em laboratório, dessa forma, quando houver casos não controlados, como cenas de crimes, o processamento e interpretação dos dados serão mais fáceis, possibilitando que laudos sejam gerados e que criminosos sejam encontrados.

Nesse sentido, alguns tópicos importantes a serem estudados são, o recolhimento de materiais, a perícia e o confronto balístico.

2.3.1 Recolhimento de Materiais

O primeiro estágio de qualquer análise balística, seja interna, externa ou terminal, é o recolhimento do material. Para se ter uma análise correta, é necessário o correto recolhimento dos materiais, desde armas a projéteis.

Quanto às armas, é necessário desmunicar a arma para transporte e manuseio com segurança. É necessário embalar a arma preservando todas as impurezas que há nela para assim poder determinar onde ela foi disparada, em que meio e por quem, esse último caso utilizando outra ciência forense, a papiloscopia.

Quanto aos projéteis, o mesmo processo é necessário, recolher os projéteis preservando as impurezas que nele constam para saber por onde ele passou, o que perfurou. Ainda, é importante registrar a posição na qual o projétil estava quando encontrado, para assim facilitar a determinação da direção de disparo.

2.3.2 Perícia

A perícia consiste na análise detalhada de algum objeto, região, local, com o objetivo de encontrar e estudar vestígios para elucidação de crimes. Os principais tipos de perícias na balística forense envolvem armas de fogo, resíduos de disparos e as munições.

2.3.2.1 Armas de Fogo

A perícia em armas de fogo na balística forense é geralmente associada a identificação da arma, a sua testagem e catalogação. A forma de identificação da arma é por meio de seu número de série (DEBETIL et al., 2015), porém, em muitos casos os números de série

das armas são raspados, lixados, fundidos, entre outras formas de se tentar apaga-los ou esconde-los.

Quanto a esses casos, a perícia trabalha com a recuperação desses números de série por meio da utilização de soluções químicas, lixas, e produtos específicos capazes de revelar o número danificado (BATISTA, 2019). Essas recuperações nem sempre são possíveis devido ao nível de danificação que o número recebeu. Com esse problema em mente, atualmente vem se desenvolvendo uma técnica de análise das regiões danificadas utilizando microscópios eletrônicos de milhões de vezes de aumento para conseguir revelar microscopicamente os números gravados na peça.

Ainda na perícia de armas, essas são testadas em tanques próprios, podendo ser de água, com espuma, bolas de plástico e qualquer material que freia o projétil sem danifica-lo, preservando as ranhuras resultantes da passagem pelo cano. Esse projétil e sua cápsula são, então, guardados para perícias de munição. Dois exemplos de tanques podem ser vistos na figura 2.10.



(a) Exemplo 1 - Tanque de Água



(b) Exemplo 2 - Tanque de Espuma

Figura 2.10 – Exemplos de Coletores de Projéteis - Fonte: (SANTOS, L. S. dos, 2015)

2.3.2.2 Resíduos de Disparos

Ao se disparar uma arma, o propelente é incendiado a fim de gerar uma liberação de gases que expulsaram o projétil da arma. Após essa queima, a cápsula sofre um recuo e é ejetada da arma. Nesse momento, partículas de propelente queimadas e não queimadas são dispersadas e entram em contato com o meio em volta e com o atirador, principalmente com as mãos ou mão utilizada para puxar o gatilho. Além de partículas do propelente, podem ser encontradas resíduos gerados pela combustão do propelente, envolvendo dióxido de carbono, monóxido de carbono, água e vapor de óxidos de nitrogênio (FREITAS, J. C. D. d., 2010) Por meio de reações químicas, é possível saber onde há resíduos de um disparo. Muitas vezes uma simples lavagem de mãos não é suficiente para eliminar os menores resíduos do disparo, possibilitando a identificação tardia.

A detecção de resíduos de disparos é muito utilizada para identificar o atirador por

meio de resíduos nas mãos, nas roupas, nos cabelos, mas além disso, também é possível descobrir com relativa precisão o local em que disparos foram efetuados, em qual direção e muitas vezes é possível julgar o porte da arma utilizada dada a área e intensidade de espalhamento dos resíduos.

2.3.2.3 Munições

O processo de perícia em munições, ou melhor dizendo, em projéteis e cápsulas, são focados em determinar qual arma foi responsável pelos disparos em certa ocasião. Pode ser útil para confirmar que a arma encontrada é a que disparou assim como pode ser útil para procurar armas no sistema ou ainda para registrar armas ao sistema.

Para se confirmar que uma arma é responsável pelos disparos realizados, é necessário comparar os projéteis visando analisar as marcas deixadas pelo cano da arma. Para isso são realizados confrontos balísticos entre projéteis e entre cápsulas. Esse é o principal assunto do trabalho e por isso esse tópico será tratado a seguir em um tópico separado.

2.3.3 Confronto Balístico

O confronto balístico é uma análise microscópica comparativa capaz de evidenciar as semelhanças entre materiais questionados e materiais padrão (SANTOS, L. S. d., 2015). Um material questionado é o material recolhido em campo, provindo de cenas de crime e que necessita ser identificado e associado a alguma pessoa. Com essa identificação facilita-se que a pessoa seja condenada na justiça por algum crime que cometeu. Por outro lado, o material padrão é aquele recolhido na perícia de armas de fogo, são projéteis e cápsulas recolhidas em laboratório sem marcas provindas do meio, apenas com marcas provindas da arma.

Com a disponibilidade de materiais padrão, utiliza-se o confronto balístico para confirmar ou refutar que tal projétil recolhido em campo foi expelido por alguma arma suspeita.

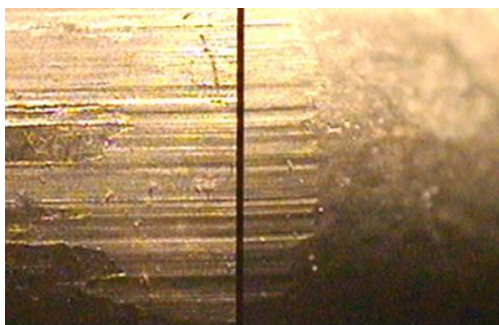
Para realizar os confrontos entre projéteis, primeiramente deve-se analisar o estado do projétil, caso esteja muito danificado, pode não ser possível fazer essa análise. Em seguida, deve-se conferir o calibre do projétil e a quantidade de raias que a arma deixou marcada no projétil. Caso esses dois números sejam compatíveis com a arma em questão, o confronto pode ser feito, caso contrário, já está confirmado a não compatibilidade da arma e do projétil questionado (SANTOS, L. S. d., 2015).

Para se realizar o confronto entre cápsulas, primeiramente é necessário analisar o calibre da cápsula em relação ao calibre da arma em questão, se não for compatível a comparação já é negativa, caso contrário deve-se comparar a posição da deformação gerada pelo pino de percussão. Caso as posições sejam muito divergentes, o resultado já pode ser considerado negativo, caso contrário pode-se prosseguir para o confronto balístico.

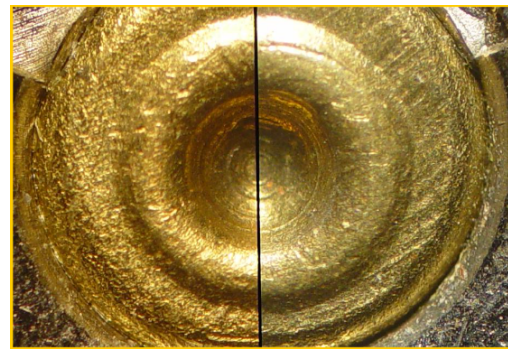
O confronto de projéteis consiste em ampliar as ranhuras existentes nos projéteis padrão e questionado e colocar essas imagens lado a lado para que as ranhuras sejam comparadas micrometricamente.

Para as cápsulas o confronto é diferente, a análise é na deformação gerada pelo pino de percussão (OLIVEIRA NETO, 2004). As imagens também são colocadas lado a lado com ampliação e busca-se encaixar metade de uma cápsula com metade da outra e encontrar um encaixe perfeito quanto a posição da marca, a profundidade, a forma e ainda podendo associar as marcas da culatra para ajudar na confirmação ou refutação da arma.

Exemplos de confrontos podem ser vistos na figura 2.11.



(a) Exemplo 1 - Confronto de Projétil



(b) Exemplo 2 - Confronto de Cápsulas

Figura 2.11 – Exemplos de Confrontos Balísticos - Fonte: (SANTOS, L. S. dos, 2015)

Para realizar esses confrontos, hoje existem duas formas, a forma mecânica e a forma automática.

2.3.3.1 Confronto Mecânico

O confronto mecânico é realizado por meio de um microscópio eletrônico de alta resolução, geralmente com uma tela acoplada. O perito, responsável por realizar os confrontos, fixa um projétil ou uma cápsula de um lado do microscópio logo abaixo de uma das lentes e em seguida fixa o outro projétil ou cápsula logo abaixo da outra lente do microscópio. Olhando nas oculares do microscópio, o perito já consegue aproximar as imagens e comparar visualmente.

No caso dos projéteis, o perito roda um dos projéteis comparando todas as ranhuras. Caso se encontre uma possível compatibilidade, ambos os projéteis serão rotacionados juntos para que confirme que todas as outras ranhuras também se alinharam (SANTOS, L. S. d., 2015). Dessa forma, é possível positivar o resultado com alta precisão.

No caso das cápsulas, tenta-se alinhar as deformações na espoleta das duas cápsulas (COSTA, 2008). Em caso de as deformações estarem deslocadas do centro, é possível mover a cápsula ou também a posição do corte de uma para a outra e assim alinhar melhor as duas.

A existência de um monitor acoplado a esse microscópio deve-se à facilidade de ampliar a imagem comparativa e assim poder haver uma discussão entre peritos para darem o veredicto em relação as peças comparadas.

O maior problema do confronto mecânico é que quando não se tem uma arma para comparar e deseja-se conferir as armas registradas anteriormente, é necessário conferir um por um até que se encontre uma compatibilidade ou nenhuma.

Um exemplo de microscópio de confronto pode ser visto na figura 2.12.



Figura 2.12 – Microscópio de Confronto - Fonte: (SANTOS, L. S. dos, 2015)

2.3.3.2 Confronto Automatizado

O confronto automatizado é uma forma de realizar os confrontos sem precisar de que um perito avalie as semelhanças (SANTOS, L. S. d., 2015). Um programa computacional é responsável por capturar as imagens dos objetos sendo confrontados e assim realizar uma comparação computacional. O sistema não necessita da presença de um exemplar padrão, seja de projétil ou cápsula. O sistema tem um banco de dados onde ficam armazenados todos os modelos padrão e ele é capaz de acessar esse banco de dados e dar conclusões baseadas neles. É possível realizar o confronto direto, sem utilizar o banco de dados, apenas comparando com uma única amostra.

Alguns exemplos de dispositivos de confrontos automatizados são:

- Evofinder
- IBIS
- Lopus

3 Fundamentação Teórica

No presente capítulo, será abordada a teoria necessária para o melhor entendimento e desenvolvimento do trabalho, explorando os principais conceitos utilizados para elaborar o projeto.

O projeto visa realizar comparações balísticas e, para isso, um possível rumo a se tomar é a captura de imagens do projétil, seja utilizando câmeras ou microscópios. Com isso em mente, a seguir será abordado o básico de microscopia e lentes, possibilitando entendimentos futuros sobre a metodologia utilizada.

3.1 Microscopia e Lentes

Microscópios são equipamentos que permitem a visualização de imagens em tamanhos maiores do que se pode ver a olho nu, muitas vezes permitindo o acesso a imagens impossíveis de se ver a olho nu. A criação desse dispositivo abriu novas portas para a ciência, podendo explorar e estudar o mundo microscópico.

Inicialmente, o poder de ampliação dos microscópios era baixo e fixo, eles tinham apenas uma lente para aumentar as imagens. Essa lente é chamada de objetiva e é capaz de ampliar imagens pequenas, invertendo a imagem.

A lente objetiva está presente em câmeras, microscópios, telescópios, entre outros. Ela apresenta o papel de capturar e focalizar a luz que entra no sistema. Seu funcionamento básico é baseado no efeito de refração da luz, isto é, quando a luz passa de um meio para outro, a diferença de densidade entre os meios provoca a mudança de direção dessa luz (PEDROTTI, F. L.; PEDROTTI, L. M.; PEDROTTI, L. S., 2017).

A objetiva é fabricada com uma forma e curvatura de superfície específica que utiliza o efeito de refração em seu favor, focalizando a luz em um ponto específico, o chamado ponto focal, local onde é formado o objeto que foi exposto a essa lente. A partir dessas características, essa lente é capaz de controlar a ampliação ou redução de uma imagem que está a sua frente (HECHT, 2012).

Com o avanço da tecnologia surgiram os microscópios compostos, os quais apresentam duas lentes, uma lente ocular e uma lente objetiva, como é possível observar na figura 3.13. A figura mostra os raios de luz que se originam no objeto que está localizado ao lado esquerdo, em seguida passam pela lente objetiva mais ao centro, chegando à lente ocular bem próxima ao olho. Por fim é possível ver o objeto imaginário que é formado na retina do olho mostrado com o nome de imagem final. É possível reparar a ampliação e inversão do objeto imaginário resultante, como mencionado anteriormente. Nessa configuração, a lente

objetiva continua com o mesmo papel, ampliação e inversão da imagem, já a lente ocular tornou-se responsável por reforçar a ampliação da imagem, dependendo apenas da distância entre ela e o ponto focal da lente objetiva.

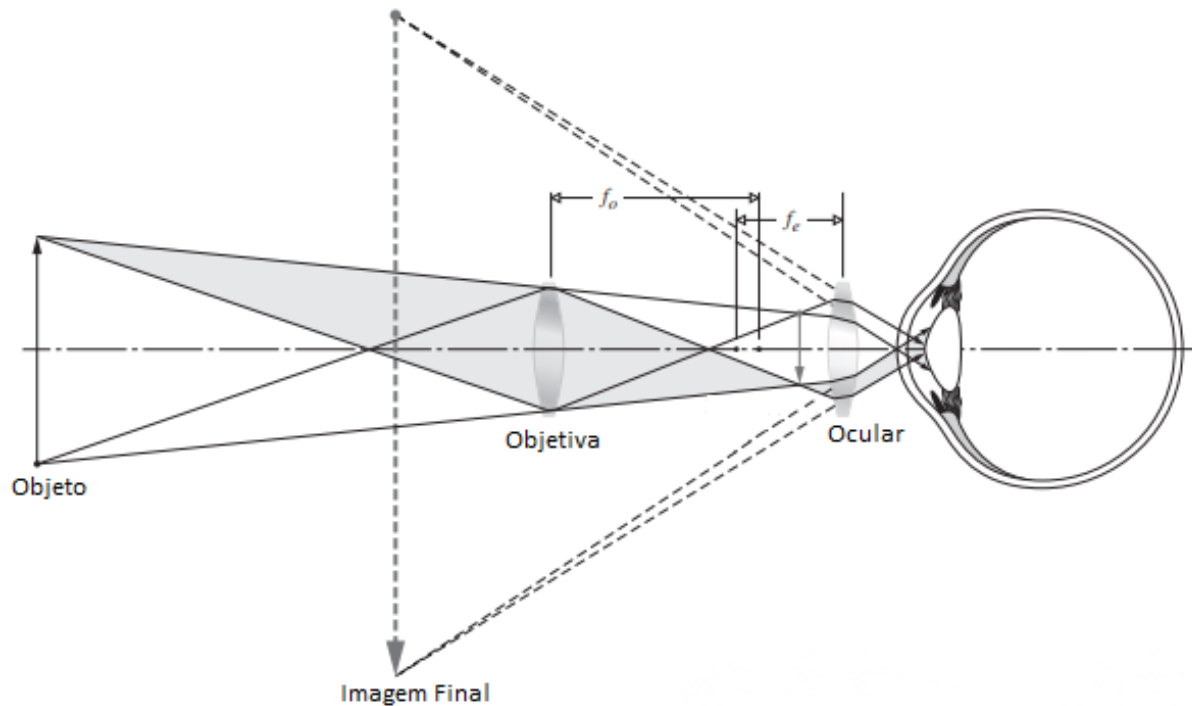


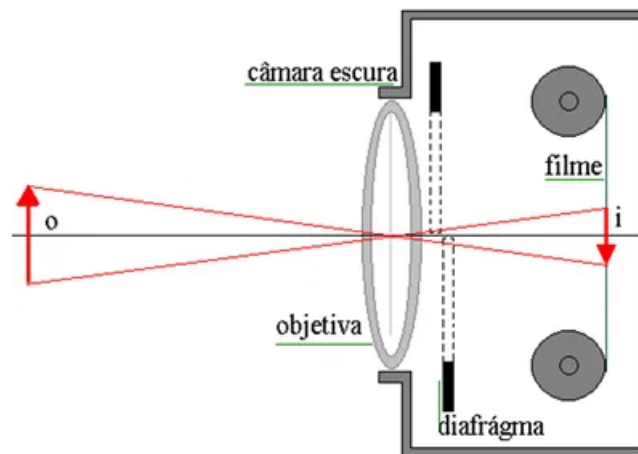
Figura 3.13 – Representação das Lentes Ocular e Objetiva em um Microscópio - Fonte: (HECHT, 2012)

3.1.1 Câmeras Fotográficas

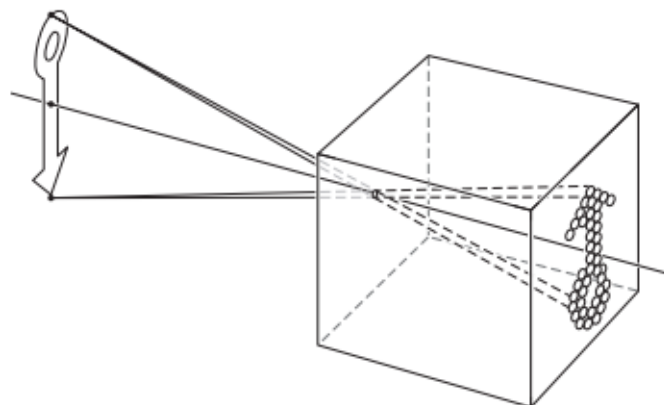
Como citado, as lentes também estão presentes em câmeras fotográficas nas quais apresentam um papel similar aos exercidos nos microscópios. Enquanto nos microscópio o objeto real é representado com ampliação, nas câmeras o objeto real é representado com redução, ou seja, o tamanho dos objetos em uma foto é inferior ao seu tamanho real.

Colocando em comparação uma câmera e o microscópio com as mesmas estruturas de lentes, a diferença principal estará na imagem gerada como mencionado, ou seja, no objeto imaginário criado. Ambos usam o mesmo par de lentes, porém o efeito é oposto. Isto está relacionado com a estrutura das lentes, como de um lado a lente recebe os raios de luz e focaliza-os de forma a reduzir a imagem observada, caso a lente receba os raios de luz pelo lado oposto, os raios de luz serão focalizados de forma a ampliar a imagem, realizando a função oposta. Dessa forma, em uma visão geral bem simplista e pouco técnica, é possível dizer que as estruturas de lentes de um microscópio e de uma câmera são as mesmas, mudando apenas o lado e a posição das lentes.

Pensando em uma câmera simples de apenas uma lente, essa lente será uma objetiva e ficará em frente a um filme ou receptor foto sensível para que a foto seja captada. A imagem será reduzida e invertida ao ser gravada. Esse efeito pode ser visto na figura 3.14 na letra a. Para que seja efetivo, é necessário que a câmera seja escura, ou seja, não se pode haver outra fonte de luz a chegar no receptor além dos raios de luz provenientes do objeto a ser fotografado. A figura 3.14 letra b mostra de forma simplificada como deve ser a estrutura externa da câmera para que seja escura.



(a) Montagem de uma Câmera - Fonte: (SILVA, D. C. M. da, 2012)



(b) Estrutura Externa de uma Câmera - Fonte: (HECHT, 2012)

Figura 3.14 – Representações de uma Câmera

3.1.2 Campos Claros e Campos Escuros

Como mencionado anteriormente, os dispositivos que utilizam lentes necessitam que as luzes que alcançam suas lentes sejam provenientes do(s) objeto(s) que se deseja(m) observar. Considerando nessa análise apenas objetos que não apresentam luz própria, há apenas duas formas de se obter luzes provenientes do objeto em análise, sendo a primeira a

existência de uma fonte de luz externa fazendo com que a luz transpasse o objeto e atinja as lentes e a segunda a existência também de alguma fonte de luz externa que ilumina o objeto que por sua vez reflete a luz, direcionando-a às lentes.

O conceito que explica as duas formas de obtenção de luz pelas lentes é chamado de campos claros e campos escuros. Segundo [Murphy \(2001\)](#), campos claros podem ser descritos como campos onde a fonte de luz é colocada atrás da amostra a ser observada, fazendo com que a luz recebida pelas lentes seja direta. Esse efeito permite visualização das bordas existentes nas amostras uma vez que os pontos onde há falta de luz ocorrem devido ao bloqueio da luz pela amostra. Amostras que apresentam superfícies translúcidas podem permitir a passagem parcial da luz, apresentando uma intensidade variada de luz que atinge as lentes. Dessa forma, a imagem resultante apresentará tons de cinza já que há apenas a variação de intensidade de luz que transpassa a amostra e atinge as lentes ([MURPHY, 2001](#)).

Por outro lado, [Murphy \(MURPHY, 2001\)](#) descreve os campos escuros como campos onde a fonte de luz se encontra de forma oblíqua à amostra, fazendo com que a única luz que atinge as lentes seja proveniente de uma reflexão na amostra. Essa configuração permite a visualização de detalhes da estrutura da amostra em contraste com um fundo escuro. Em geral a luz direcionada à amostra passa por um difusor ao sair da fonte, gerando o efeito de dispersão da luz, atingindo a maior área possível da amostra de forma aproximadamente uniforme, dessa forma, a imagem resultante após a luz passar pelas lentes terá o máximo de detalhes possível com uniformidade na iluminação e contraste da imagem. Para esse tipo de campo as imagens resultantes podem ser coloridas, a depender do tipo de receptor utilizado, visto que a origem da luz é a reflexão em superfície ([MURPHY, 2001](#)).

A figura 3.15 mostra ao lado esquerdo uma amostra biológica visualizada ao utilizar campo claro e ao lado direito pode-se ver a mesma amostra visualizada em campo escuro.

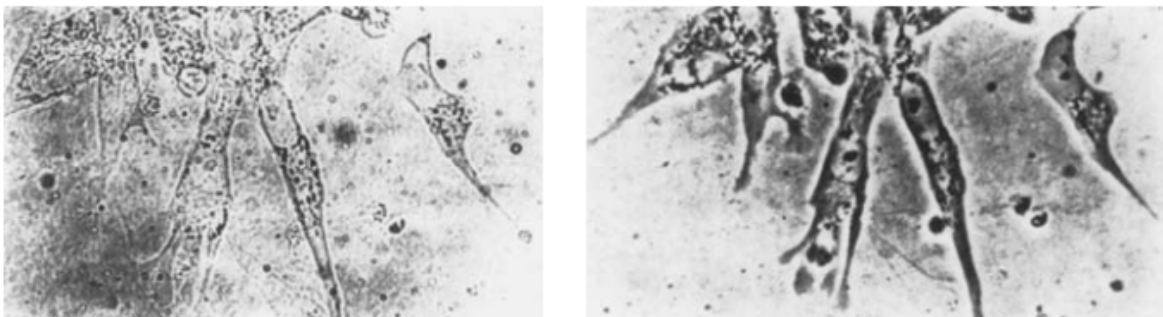


Figura 3.15 – Demonstração da Diferença entre Campo Claro (esquerda) e Campo Escuro (Direita) -
Fonte: ([MURPHY, 2001](#))

3.2 A Imagem no Âmbito Digital

Anteriormente foi discutido o conceito de câmera e sua funcionalidade. Como produto final, a câmera gera uma imagem em seu ponto focal e lá é colocado um anteparo capaz de registrar a imagem gerada. Existem duas classes principais para registros de imagens, registro analógico e registro digital.

O registro analógico é realizado por meio de anteparos fotossensíveis, os quais são marcados definitivamente ao entrarem em contato com a luz. O registro é realizado por meio de reações químicas que ocorrem no anteparo quando este entra em contato com a luz. Nesse processo, em geral, o anteparo é chamado de filme fotográfico e apenas registra as alterações químicas que ocorrem no processo, sendo levado posteriormente para o processo de revelação que nada mais é que a transformação das alterações químicas em imagens visíveis (LONDON; STONE; UPTON, 2017). Neste trabalho os registros analógicos não foram trabalhados, portanto o tópico não será aprofundado.

O registro digital, por outro lado, utiliza sensores na posição do anteparo. Sensores podem ser ditos como dispositivos que são capazes de perceber alguma interação e gerar alguma resposta com relação a tal interação. Os sensores utilizados nas câmeras são capazes de converter a luz recebida em sinais elétricos e estes são posteriormente convertidos em valores digitais. Esses valores digitais caracterizam a imagem especificando valores discretos para intensidade de cor e luminosidade. Os sinais elétricos geram os valores digitais mencionados para seções da imagem capturada, cada seção é chamada de *pixel* (LONDON; STONE; UPTON, 2017). Existem sensores variados e cada sensor tem capacidade de gerar uma certa quantidade de sinais elétricos, e quanto mais sinais forem possíveis, mais seções serão representadas na imagem gerada, mais *pixels*. Dessa forma, define-se resolução, ou seja, quanto mais *pixels* a imagem apresentar, maior é a resolução da imagem, maior a qualidade e mais detalhes são registrados. A figura 3.16 mostra o aumento da qualidade da imagem capturada de acordo com o aumento da resolução de captura, aumentando da esquerda para a direita.

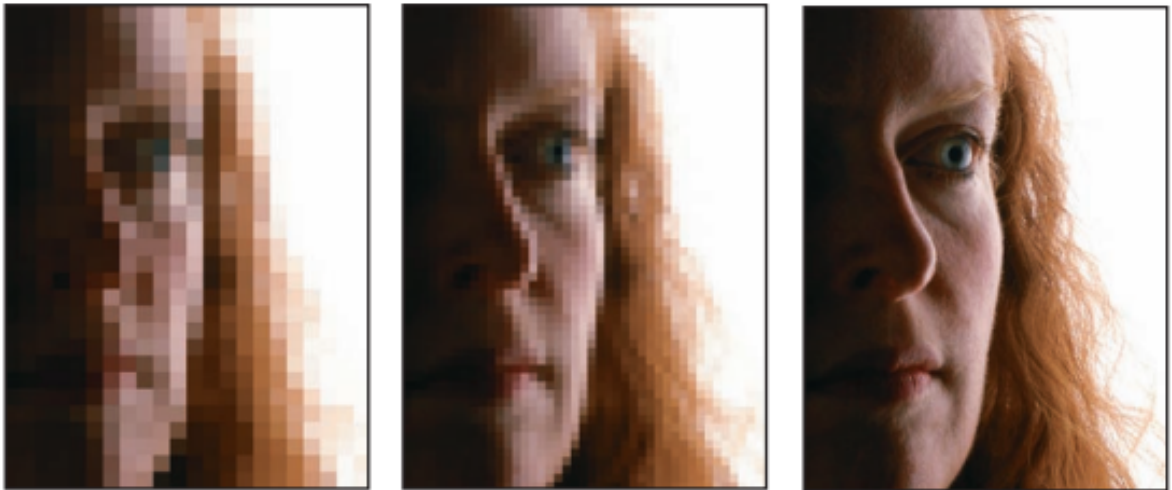


Figura 3.16 – Variação de Quantidade de Pixels Gerados - Fonte: (LONDON; STONE; UPTON, 2017)

Com a configuração de *pixels* estabelecida, é possível representar qualquer imagem a partir de uma função bidimensional $f(x,y)$ onde x e y representam coordenadas espaciais para localizar cada *pixel* e f é a função da intensidade luminosa de cada um deles (FILHO; NETO, H. V., 1999). A figura 3.17 mostra a convenção estabelecida para os eixos das coordenadas x e y .



Figura 3.17 – Convenção para as Coordenadas Espaciais dos *pixels* - Fonte: (FILHO; NETO, H. V., 1999)

Como é possível notar, a imagem mostrada na figura 3.17 apresenta coloração em tons de cinza e isso se deve ao tipo de sensor utilizado para capturar esta imagem assim como ao método de transformação de sinais elétricos em valores digitais. Essa coloração observada pode ser chamada de monocromática, ou seja, apresenta apenas uma cor com variações de tons. Existem diversos outros formatos para se registrar uma imagem, como por exemplo, binário, ponto flutuante, *Red Green Blue* (RGB), *Cyan Magenta Yellow Key/Black* (CMYK), *Hue Saturation Value* (HSV), entre outras (GONZALEZ; WOODS, 2002).

Os formatos binário e ponto flutuante se comportam como o formato monocromático, são representados por uma função $f(x,y)$ que caracteriza a intensidade luminosa. Enquanto o monocromático apresenta diversos valores discretos inteiros para a intensidade luminosa, o binário apresenta somente dois valores, 0 e 1, sendo o primeiro a cor preta e o segundo a cor branca. Já o ponto flutuante apresenta valores discretos fracionários para descrever a luminosidade (GONZALEZ; WOODS, 2002).

Por outro lado, os formatos RGB, CMYK, HSV e diversos outros são formatos que trazem mais de uma característica para a imagem, apresentam bandas distintas com propósito de trazer mais informações e enriquecer a imagem resultante. No caso do CMYK há quatro bandas, uma banda para a cor ciano, uma banda para a cor magenta, uma banda para a cor amarela e uma banda para a cor preta. Dessa forma, é possível associar as cores das bandas e ter como resultado uma imagem colorida. Já o HSV apresenta três bandas, uma banda para o matiz ou tonalidade, uma banda para a saturação e uma banda para o brilho, sendo essa outra forma de se obter uma imagem colorida como resultado. Por fim, RGB apresenta três bandas, uma banda para a cor vermelha, uma banda para a cor verde e uma banda para a cor azul, funcionando de forma semelhante ao formato CMYK (GONZALEZ; WOODS, 2002).

Para representar estes e outros formatos de imagem é necessário uma função $f(x,y)$ para cada banda. Para o RGB, por exemplo, a cor vermelha (*red*) tem a função $f_R(x,y)$, a cor verde (*green*) tem a função $f_G(x,y)$ e a cor azul (*blue*) tem a função $f_B(x,y)$. Essa representação evidencia que é possível considerar que para cada cor existe uma imagem monocromática e sua paleta, sendo a imagem final a união das imagens monocromáticas geradas. Essa união pode ser escrita como mostra a fórmula (3.1) (FILHO; NETO, H. V., 1999).

$$f(x,y) = f_R(x,y) + f_G(x,y) + f_B(x,y) \quad (3.1)$$

3.3 Visão Computacional

Após conseguir capturar as imagens utilizando a teoria explanada até aqui, o projeto exige tratamento das imagens. É necessário ter uma imagem completa do projétil para analisá-lo, ou seja, é necessário unir todas as imagens capturadas em uma só. Para isso, é

necessário extrair informações importantes das imagens para que seja possível manipulá-las. Com isso em mente, serão explorados a seguir conceitos sobre visão computacional.

O mundo na visão dos seres humanos apresenta cores, dimensões, profundidade, emoções, padrões e todas essas características são facilmente identificadas, pode-se dizer que são intrínsecas à interpretação humana. Máquinas, por sua vez, não apresentam tais características e não são capazes de interpretar imagens e vídeos como humanos (FORSYTH; PONCE, 2012). Como foi visto, uma imagem capturada por uma câmera é interpretada no computador como uma função $f(x,y)$, ou seja, o mundo tridimensional se torna apenas bidimensional digitalmente, fato esse que se estende para os vídeos.

Com tais problemas em vista, surge a visão computacional, um campo interdisciplinar que visa recuperar a tridimensionalidade, a profundidade, as interpretações e associações, ou seja, fazer com que os computadores consigam, de certa forma, interpretar imagens e vídeos de forma similar aos humanos (SZELISKI, 2011). É uma área da computação que envolve diversas esferas científicas, como por exemplo a visão biológica, o processamento de imagens, álgebra linear, ótica e fotografia, matemática e estatística, computação gráfica, processamento de sinais, inteligência artificial, aprendizado de máquina, entre outros.

Dentre as áreas mencionadas anteriormente, duas se sobressaem com relação ao escopo do atual trabalho, sendo elas processamento de sinais bidimensionais espaciais (imagens) e processamento de sinais unidimensionais.

3.3.1 Processamento de Imagens

O processamento de imagens é uma das áreas de visão computacional e nela tem-se a concentração de análises e manipulações de imagens digitais. Para isso, utiliza-se de algoritmos, extraíndo diversas informações e reconhecendo padrões, tornando possível um grande número de aplicações.

Segundo Filho e Hugo Vieira Neto (1999), atualmente o processamento de imagens chama muita atenção em duas grandes categorias:

1. O aprimoramento de informações melhorando a interpretação humana;
2. A análise automática de informações extraídas de uma imagem.

Com a evolução do processamento, ambas as categorias se dispersaram por diversos ramos das atividades humanas, podendo algum dia chegar a todas. Alguns exemplos são a medicina com auxílio no diagnósticos por imagem, a biologia com para análises microscópicas, a geografia com análises de imagens de satélite, controle de qualidade em indústrias e diversas outras áreas como geoprocessamento, meteorologia, sensoriamento, astronomia, segurança (FILHO; NETO, H. V., 1999).

Para a realização de toda a laboração de processamento, é necessário seguir um sistema com etapas básicas e necessárias. Um sistema de processamento de imagens focado em visão computacional pode ser visualizado na figura 3.18.

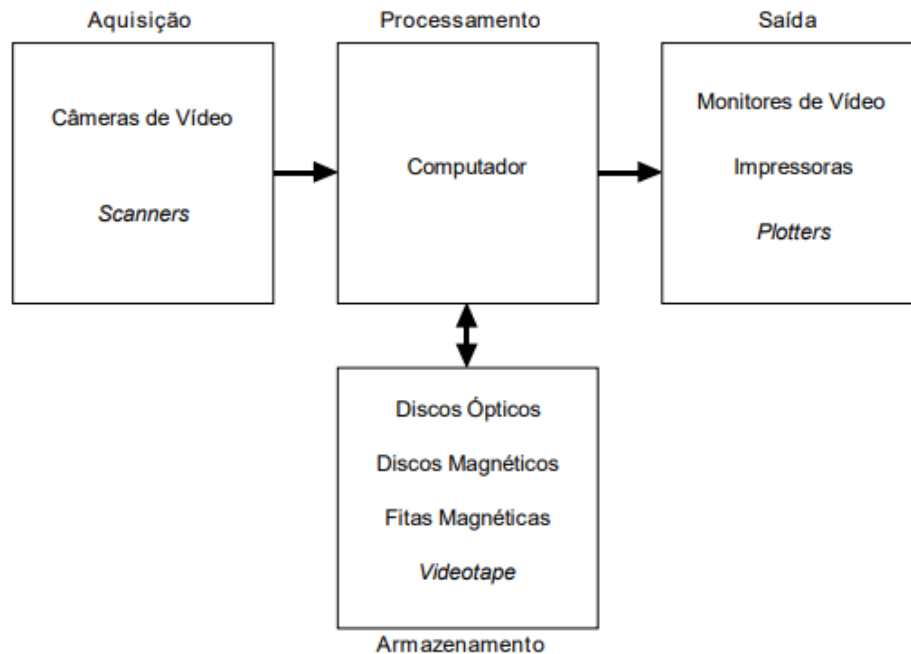


Figura 3.18 – Elementos de um Sistema de Processamento de Imagens em Visão Computacional -
Fonte: (FILHO; NETO, H. V., 1999)

A primeira etapa é chamada de aquisição e é nela onde a imagem é transformada em um sinal elétrico que posteriormente se torna uma representação numérica discreta passível de processamento digital. Para isso é necessário duas configurações de hardware, um dispositivo sensível capaz de receber um estímulo, seja por luz visível (câmeras), por raios x, radiação ultravioleta ou raios infravermelhos e gerar sinais elétricos analógicos. E um segundo dispositivo que possa receber os sinais elétricos analógicos e os transformar em sinais digitais.

A fase seguinte consiste em de fato processar as imagens digitais captadas anteriormente. Aqui diversos algoritmos computacionais são aplicados às imagens, podendo passar por correções, cálculos estatísticos, extração de estruturas, detecção de texturas, entre outros. Nessa etapa a sequência de processos escolhida depende do objetivo que se tem. A depender do processamento escolhido, é necessário utilizar algum meio de armazenamento das imagens digitais, podendo recorrer a imagens já processadas. Por fim, é necessário visualizar os resultados através de algum meio de saída como monitores, impressões, projeções.

Para o presente trabalho, o processamento de imagens está associado com a necessidade de unir as diversas imagens capturadas de um projétil. Para isso é necessário utilizar métodos capazes de identificar características importantes de cada imagens, características estas que servirão para relacionar as imagens consecutivas e uni-las. Além de identificar

tais características, é importante escolher métodos que não dependem de características como variação de iluminação, de escala e de inclinação visto que é desejado poder analisar diversos modelos de projéteis, maiores, menores, de vários materiais e formatos. Pensando nisso, ressaltam-se quatro métodos:

- Scale-Invariant Feature Transform - SIFT
- Speeded-Up Robust Features - SURF
- Oriented FAST and Rotated BRIEF - ORB
- Binary Robust Invariant Scalable Keypoints - BRISK

3.3.1.1 *Scale-Invariant Feature Transform* - SIFT

Uma opção de processamento de imagens é a utilização do método *Scale-Invariant Feature Transform* ou SIFT, que nada mais é que um algoritmo de detecção e descrição de características invariantes à escala, isto é, o SIFT foi projetado para encontrar pontos-chave (*keypoints*) em imagens de forma a ser robusto a variações de escala, de rotação, de iluminação e de diversos outros fatores.

Publicado em 1999 pelo pesquisador David Lowe ([LOWE, 2004](#)), professor do departamento de Ciências da Computação na *University of British Columbia* no Canadá, o SIFT se tornou uma técnica fundamental na área de visão computacional e processamento de imagens.

Esse método é dividido em etapas, sendo cada uma responsável por garantir a robustez a variações. Nesse pensamento, a primeira etapa visa garantir a invariância dos resultados com relação a mudanças de escala das imagens. Quando se altera a escala de uma imagem, muitas vezes perde-se informação. A figura 3.19 mostra duas imagens da mesma região em escalas diferentes, sendo que o kernel de análise para ambas tem a mesma área. É possível reparar que na imagem à esquerda cada kernel cobre completamente cantos existentes na curva desenhada, conseguindo identificar cantos, já na imagem à direita há apenas identificação de bordas, ou seja, linhas retas ou quase retas, sem identificação de cantos. Sendo assim, houve perda de informação, prejudicando o resultado final da comparação entre as imagens.

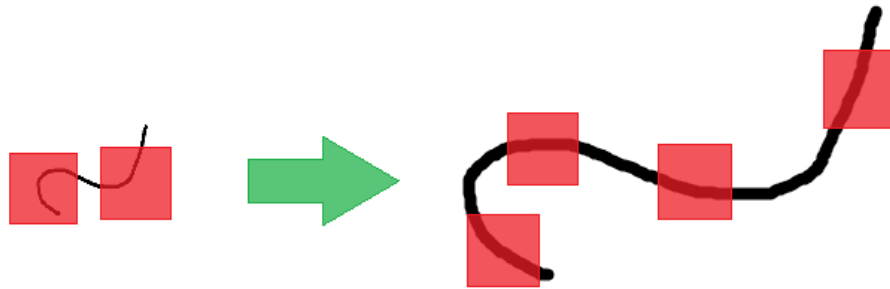


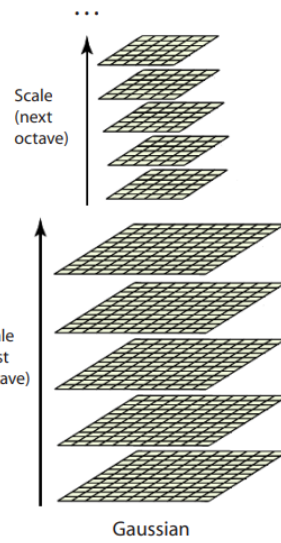
Figura 3.19 – Kernels de Mesma Área e Regiões em Escalas Diferentes - Fonte: Autor

A primeira etapa do método traz a construção de um espaço de escalas, isto é, é construída uma pirâmide de imagens a partir de uma imagem de entrada $I(x,y)$. São criadas novas versões da imagem de entrada utilizando um filtro de suavização gaussiano de escala variável, representado por $G(x,y,\sigma)$. Este filtro será responsável por realçar características marcantes da imagem e suprimir características indesejadas (LOWE, 2004).

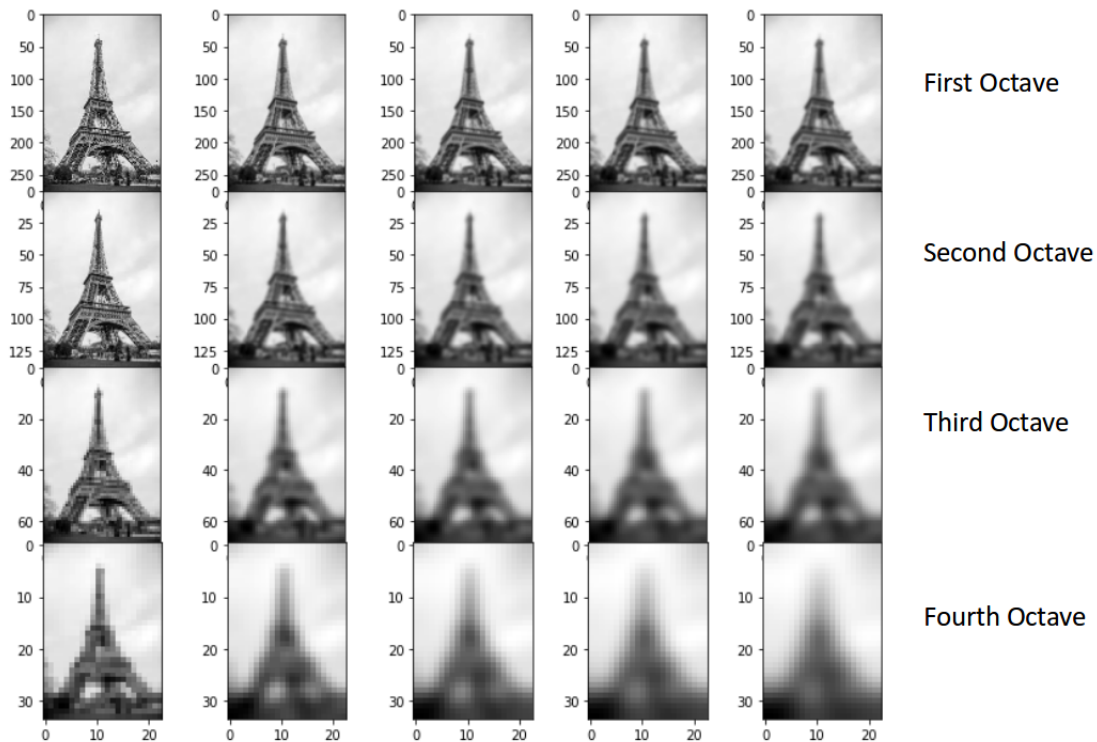
A primeira etapa do método traz a construção de um espaço de escalas, isto é, é construída uma pirâmide de imagens a partir de uma imagem de entrada $I(x,y)$. Para realçar características marcantes da imagem e suprimir características indesejadas, são utilizados filtros. Em processamento de imagens, filtros são matrizes de números aplicadas sobre os *pixels*, alterando seus valores de formas específicas e planejadas. No caso deste método, são criadas novas versões da imagem de entrada utilizando um filtro de suavização gaussiano de escala variável, representado por $G(x,y,\sigma)$ (LOWE, 2004). Este filtro suavizará algumas características específicas e realçará outras.

A figura 3.20 mostra uma representação da pirâmide de imagens na letra a e mostra um exemplo na letra b. No processo, a imagem de entrada é replicada algumas vezes e a cada réplica gerada, o filtro gaussiano é aplicado. Em seguida a escala da primeira imagem é reduzida e esta é replicada e filtrada e todo o processo se repete pela quantidade de vezes desejada. Cada grupo de imagens de mesma escala foi chamado de oitava (*octave*) por Lowe. Todo esse processo pode ser descrito pela equação (3.2), onde $L(x,y,\sigma)$ é a função resultante da convolução entre o filtro gaussiano e a imagem, e σ representa o intensidade de aplicação do filtro (LOWE, 2004).

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (3.2)$$



(a) Representação Teórica - Fonte: (LOWE, 2004)



(b) Exemplo Torre Eiffel - Fonte: (SINGH, A., 2019)

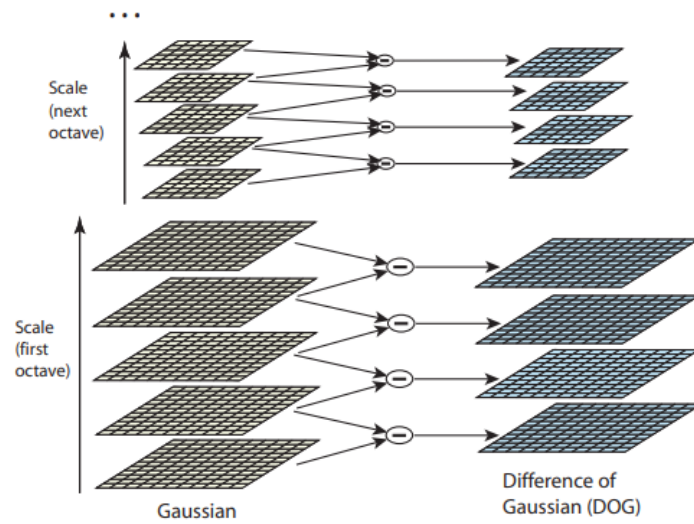
Figura 3.20 – Espaço de Escalas

Após a geração das gaussianas, a etapa seguinte é calcular a diferença entre as imagens de mesma oitava. A função *Difference of Gaussian* (DoG) é utilizada para realizar essa operação, sendo representada na equação (3.3), onde $D(x,y,\sigma)$ é a DoG e k representa uma variação do filtro gaussiano. A figura 3.21 mostra uma representação da DoG entre imagens na letra a e mostra um exemplo na letra b.

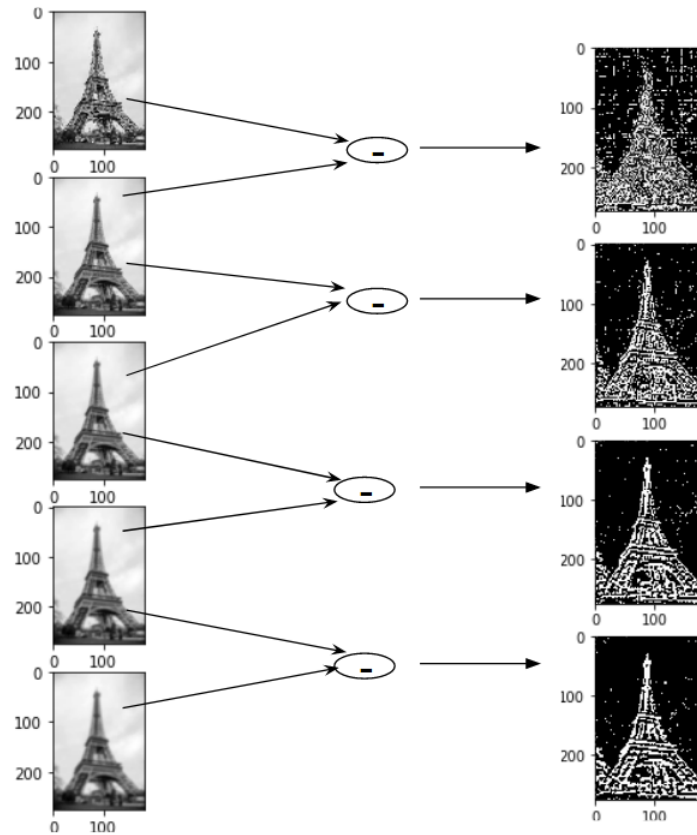
$$D(x,y,\sigma) = G(x,y,k\sigma) * I(x,y) - G(x,y,\sigma) * I(x,y)$$

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)$$

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (3.3)$$



(a) Representação Teórica - Fonte: (LOWE, 2004)



(b) Exemplo Torre Eiffel - Fonte: (SINGH, A., 2019)

Figura 3.21 – Difference of Gaussian

A partir deste momento serão calculados os pontos de extremo, isto é, os máximos e mínimos locais. Esses pontos apresentam características invariantes à escala, conferindo invariabilidade escalar para o método. Para calcular os extremos, deve-se comparar a intensidade de um *pixel* com seus 26 vizinhos mais próximos como mostra a figura 3.22. Na figura nota-se um *pixel* representado pela letra x que tem oito *pixels* a sua volta e mais nove *pixels* acima e abaixo. Os *pixels* representados a cima e abaixo são referentes as imagens das escalas adjacentes (LOWE, 2004).

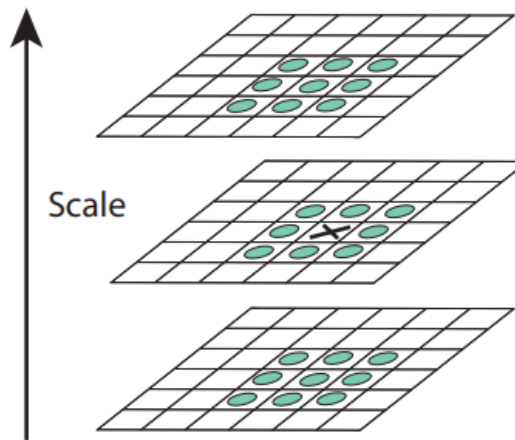


Figura 3.22 – Comparação do *pixel* 'x' com os 26 mais próximos - Fonte: (LOWE, 2004)

Os pontos-chave (*keypoints*) então foram encontrados, porém é necessário filtrar ainda mais tais pontos, visto que alguns desses pontos podem não ser robustos com relação a ruídos. Dois tipos de pontos são removidos, pontos de baixo contraste e pontos de borda.

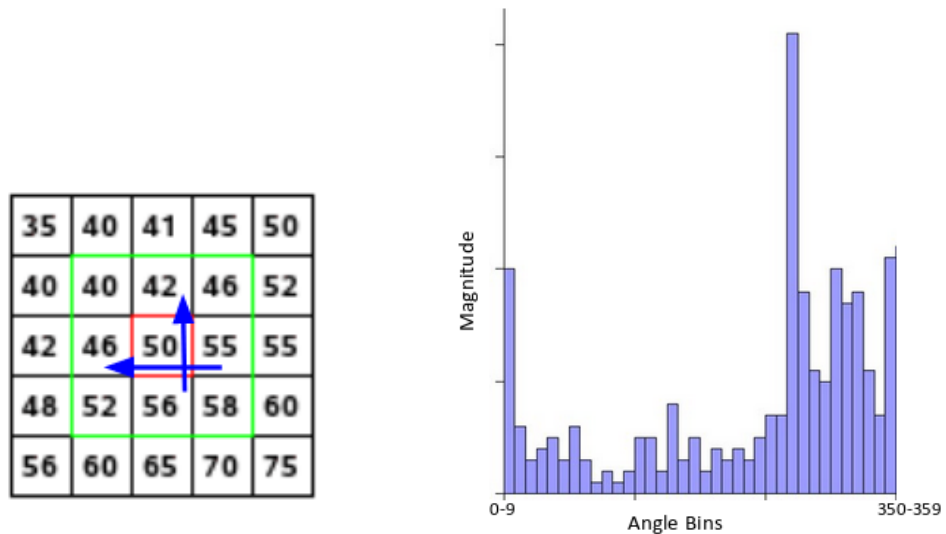
Com pontos estáveis selecionados, deve-se então designar orientação a eles para que o método seja invariante à rotação. Dois passos serão tomados aqui, calcular a magnitude e orientação e criar um histograma para esses valores. O cálculo parte das intensidades dos *pixels* adjacentes ao de referência. Considerando a figura 3.23 letra a, a magnitude e a orientação do *pixel* marcado em vermelho podem ser calculadas a partir do gradiente nas direções x e y. Para calcular este gradiente deve-se encontrar a diferença de intensidade entre os *pixels* adjacentes tanto para a direção x como para a direção y. Nesse exemplo têm-se essas variações como $G_x = 55 - 46 = 9$ e $G_y = 56 - 42 = 14$. A partir desses valores calcula-se a Magnitude e a Orientação pelas equações (3.4) e (3.5), consecutivamente (SINGH, A., 2019).

Seguindo o padrão mencionado, serão calculadas a magnitude e a orientação para todos os *pixels* que estão em volta de um *keypoint*. Todos esses valores encontrados serão utilizados para montar um histograma, como está exemplificado na figura 3.23 letra b. É possível observar um pico de magnitude em destaque no histograma, sendo essa a orientação final do ponto-chave (*keypoint*). Realizando esse procedimento para todos os ponto-chave

tem-se a orientação definida para todos eles e garante-se a robustez quanto a variação de orientação.

$$Magnitude = \sqrt{(G_x)^2 + (G_y)^2} \quad (3.4)$$

$$Orientação(\Phi) = \arctan\left(\frac{G_y}{G_x}\right) \quad (3.5)$$



(a) Matriz de *Pixels* de Intensidade (b) Histograma de Orientação por Magnitude

Figura 3.23 – Orientação e Magnitude dos *Keypoints* - Fonte: (SINGH, A., 2019)

O último passo a ser dado é a determinação dos descritores dos pontos-chave, ou *descriptors*. Eles são informações que tornam os *keypoints* bem distintos um com relação ao outro, ou seja, tem-se uma forma robusta para identificar unicamente cada ponto-chave.

Os descritores são calculados de forma semelhante às orientações e magnitudes calculadas anteriormente. Em torno de cada ponto-chave será selecionado um bloco de 16×16 *pixels*, os quais serão divididos em quatro seções de 8×8 *pixels*. Em cada seção serão gerados os histogramas como anteriormente e serão definidas orientações e magnitudes para cada *pixel*. Em seguida, em cada seção de 4×4 *pixels* as orientações e magnitudes dos 16 *pixels* serão reorganizadas para representarem apenas oito orientações e oito magnitudes. Dessa forma, cada bloco de 8×8 *pixels* terá quatro conjuntos de oito orientações e oito magnitudes e então os blocos de 16×16 apresentarão 16 conjuntos de oito orientações e oito magnitudes, resultando em 128 descritores ($16 \times 8 = 128$) para cada ponto-chave (LOWE, 2004). A figura 3.24 demonstra esse processo de definir os 128 descritores.

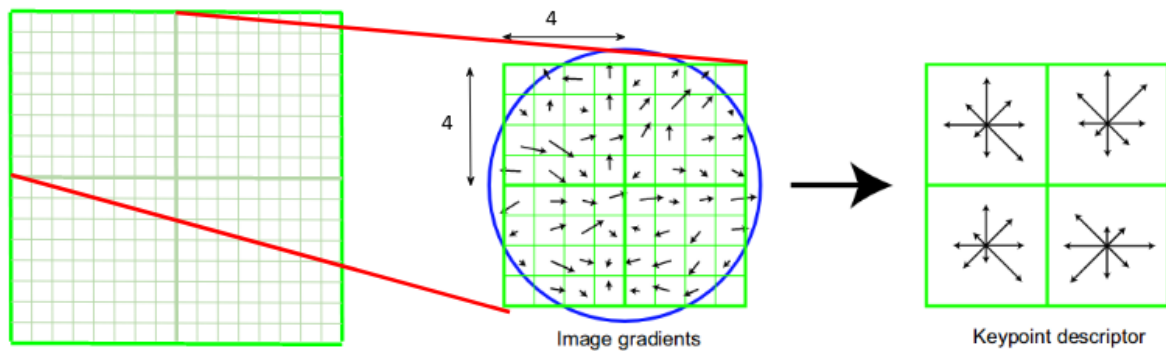


Figura 3.24 – Criação de Descritores - Fonte: (SINGH, A., 2019)

Ainda com tamanha robustez, existe a possibilidade dos descritores sofrerem influência da iluminação e serem modificados. Para evitar esse percalço, é realizada uma normalização unitária dos descritores, tornando os descritores invariantes à mudanças lineares na iluminação uma vez que estes descritores se baseiam em diferenças de gradientes. Porém é possível que haja variação não linear de iluminação, por esse motivo Lowe, experimentalmente, definiu um limiar de 0,2, limitando os valores do vetor de descritores e evitando a influência de grandes gradientes. Sendo assim, o vetor é limitado a valores de até 0,2 e em seguida é normalizado unitariamente novamente, tornando o método também invariável a mudança de iluminação (LOWE, 2004).

3.3.1.2 *Speeded-Up Robust Features* - SURF

Speeded-Up Robust Features (SURF) é um algoritmo de processamento de imagens que teve como base para sua criação o algoritmo SIFT, buscando características similares como a detecção de pontos de interesse dos quais serão extraídos descritores robustos e invariantes à escala e à rotação.

O SURF foi projetado para apresentar uma melhor performance de tempo de execução quando comparada com o SIFT e para isso são utilizadas imagens integrais e filtros simplificados ao invés de se utilizar convoluções de derivadas da função gaussiana como o SIFT implementa. Sendo assim, Bay, Ess et al. (2008) decidiram aproximar os resultados da aplicação do *Difference of Gaussian* (DoG) de Lowe utilizando filtros do tipo caixa.

Para compreender as etapas do método SURF é necessário primeiramente se familiarizar com o conceito de Imagem Integral. Esta imagem é criada a partir de uma imagem de referência, escolhe-se um ponto x na imagem de referência e cria-se um retângulo de origem em x . A imagem criada apresenta o tamanho do retângulo e cada *pixel* dela apresenta um valor que é resultante da soma das intensidades de todos os *pixels* que se encontram acima e a esquerda daquele *pixel* na imagem original. Essa configuração é importante por agiliza

o cálculo da intensidade geral da região retangular uma vez que é necessário apenas os quatro vértices do retângulo para realizar este cálculo (BAY; ESS et al., 2008). Ao observar a figura 3.25 é possível reparar como, por exemplo, o número 28 destacado em rosa na imagem integral é resultante da soma dos números 3, 2, 7, 1, 5, 1, 5, 1 e 3 presentes na imagem original, ou seja, a soma de todos os números acima e a esquerda da posição do *pixel* de referência, incluindo o *pixel*.

3	2	7	2	3
1	5	1	3	4
5	1	3	5	1
4	3	2	1	6
2	4	1	4	8

Imagem Original

3	5	12	14	17
4	11	19	24	31
9	17	28	38	46
13	24	37	48	62
15	30	44	59	81

Imagem Integral

Figura 3.25 – Criação de Imagem Integral - Fonte: (SZELISKI, 2011)

Assim como no método SIFT, o primeiro passo a ser dado é a determinação dos pontos de interesses, pontos-chave (*keypoints*). Para isso, utiliza-se uma matriz Hessiana, ou seja, uma matriz quadrada cujos componentes são derivadas parciais de segunda ordem de uma função multivariável (equação (3.6)).

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y \partial y} \end{bmatrix} \quad (3.6)$$

Para garantir adaptação para qualquer escala, a matriz Hessiana passa por um kernel gaussiano, tomando uma nova forma, representada na equação (3.7), onde $L_{xx}(x,\sigma)$ é a convolução da derivada de segundo grau da Gaussiana com a imagem Integral mencionada anteriormente (BAY; TUYTELAARS; VAN GOOL, 2006).

$$\mathcal{H}(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (3.7)$$

A partir deste ponto tem-se uma matriz capaz de descrever a taxa de variação da intensidade dos *pixels* da imagem em função das variáveis x e y . Como é desejado destacar os pontos de interesse, é calculado o determinante da matriz Hessiana (equação (3.8)), onde D_{xx} , D_{yy} e D_{xy} são aproximações das convoluções e w é um peso atribuído (BAY; TUYTELAARS; VAN GOOL, 2006).

$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (3.8)$$

Tem-se como resultado valores diversos, sendo detectados os valores de máximo encontrados. Nesse ponto iniciou-se a coleta dos pontos de interesse, porém pode-se encontrar estes pontos em escalas diferentes. Assim como no SIFT, implementa-se um espaço escalar com estrutura piramidal de imagens. No entanto, no caso do SURF, as dimensões da imagem não será alterada, serão utilizados diferentes escalas de máscaras gaussianas, ampliando-as entre oitavas (figura 3.26) (BAY; ESS et al., 2008). Assim como no SIFT, não há limites de oitavas, pode-se prosseguir criando novas desde que o sistema sustente o processamento completo. A cada oitava criada mais pontos de interesse (*keypoints*) são coletados.

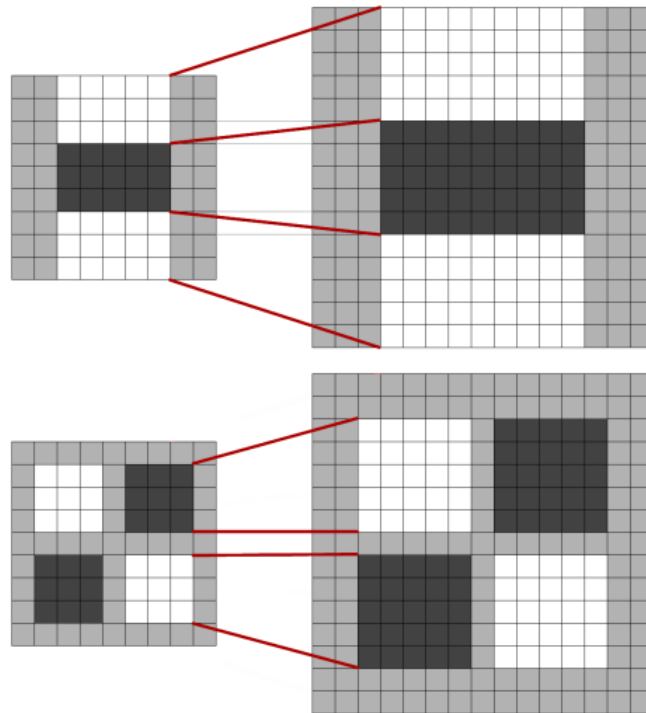


Figura 3.26 – Criação de Diferentes Escalas de Máscaras Gaussianas - Fonte: (BAY; ESS et al., 2008)

Para assegurar a invariabilidade a rotação, calcula-se a *wavelet* Haar das direções de x e y com uma vizinhança circular de raio $6s$, onde s diz respeito a escala na qual o ponto de interesse está localizado. As respostas da *wavelet* são calculadas e ponderadas com uma gaussiana centrada no ponto de interesse, as respostas mostraram sua força horizontal no eixo das abscissas e sua força vertical nos eixos das ordenadas. É escolhida uma janela específica para concatenar as respostas, geralmente $\frac{\pi}{3}$. Todas as respostas são somadas enquanto a janela circula e o vetor mais longo gerado define a orientação do ponto de interesse (figura 3.27) (BAY; ESS et al., 2008).

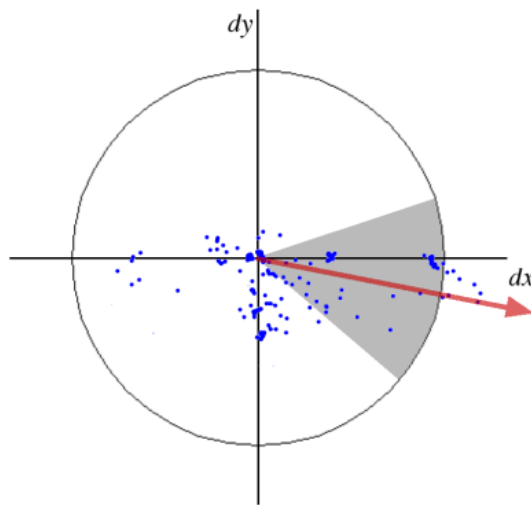


Figura 3.27 – Wavelet de Vizinhança de Raio 6s - Fonte: (BAY; ESS et al., 2008)

Por fim, para extrair os descritores, devem-se criar quadrados com lados de tamanho $20s$, onde s diz respeito a escala como mencionado anteriormente, que sejam centralizados nos seus pontos de interesse, além de serem orientados conforme as orientações definidas anteriormente. A figura 3.28 demonstra tais quadrados. Esses quadrados são então divididos em quadrados menores, 4×4 e para cada ponto da sub-região é calculada uma *wavelet* Haar nas direções horizontais e verticais. As respostas das *wavelets* na direção horizontal (dx) e vertical (dy) são somadas para cada sub-região e são armazenadas. Para trazer mais informações, pode-se utilizar o valor absoluto das respostas ($|dx|$) e ($|dy|$). Sendo assim, cada sub-região terá um vetor de descritores de quatro dimensões, $V = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Unindo os vetores das 4×4 sub-regiões, resultará em descritores de dimensão 64.



Figura 3.28 – Quadrados de lado $20s$ Orientados Pelos Pontos de Interesse - Fonte: (BAY; TUYTELAARS; VAN GOOL, 2006)

3.3.1.3 Oriented FAST and Rotated BRIEF - ORB

Oriented FAST and Rotated BRIEF, ou ORB, é um método de processamento de imagens que tem como propósito ser eficiente e robusto para detecção de pontos de interesse e descrição de características de imagens. Com o objetivo de obter um desempenho rápido e confiável na extração e correspondência de características em imagens, Ethan Rublee, Vincent Rabaud, Kurt Konolige e Gary Bradski, desenvolvedores do método, decidiram combinar de uma forma específica o detector FAST (Features from Accelerated Segment Test) e o descritor BRIEF (Binary Robust Independent Elementary Features) (RUBLEE et al., 2011).

FAST é um detector proposto por Rosten e Drummond (ROSTEN; DRUMMOND, 2006) para identificação de pontos de interesse em uma imagem. Os autores visavam possibilitar o uso de detectores em tempo real para aplicações onde os recursos computacionais fossem poucos. Para isso, era necessário que o detector fosse rápido e pouco custoso.

O algoritmo de Rosten e Drummond consiste em realizar comparações de intensidades entre um *pixel* escolhido para teste e alguns outros a sua volta. Primeiramente seleciona-se o *pixel* P a ser analisado para determinar se é ou não ponto de interesse. Define-se um Limiar de variação de intensidade L . Em seguida encontram-se os *pixels* com os quais o *pixel* questionado será comparado. Essa escolha é a partir de um Círculo de Bresenham (SINGH, T., 2018) criado com centro no *pixel* questionado. Cria-se o círculo com um raio R de *pixels* e número K de *pixels* de perímetro. Na figura 3.29 é possível visualizar um exemplo de Círculo de Bresenham com $R = 3$ e $K = 16$.



Figura 3.29 – Círculo de Bresenham com $R = 3$ e $K = 16$ - Fonte: (LEE; FULLER; MCCOWN, 2020)

Iniciam-se, então, as comparações de intensidade dos *pixels*. Os K *pixels* serão comparados com o questionado e serão classificados em três categorias:

- Mais Escuro se $I_K \leq I_P - L$
- Similar se $I_P - L < I_K < I_P + L$
- Mais Claro se $I_P + L \leq I_K$

Onde I_K é a intensidade de um *pixel* dentre os K escolhidos, I_P é a intensidade do *pixel* questionado P .

É definida uma quantidade limiar de *pixels* N para definir se o *pixel* questionado é ponto de interesse ou não. Caso haja pelo menos N *pixels* consecutivos no perímetro do Círculo de Bresenham que pertençam as categorias Mais Claro e Mais Escuro, então o *pixel* P será definido como ponto de interesse.

O processo é feito para todos os *pixels* da imagem e é criada uma lista de pontos-chave. Para acelerar o processo, pode-se comparar a intensidade de apenas alguns *pixels*. Para exemplificar, considere que $K = 16$ e $N = 12$, então pode-se comparar apenas as intensidades I_1, I_5, I_9 e I_{13} com a intensidade I_P do *pixel* questionado. Caso não haja pelo menos três dos quatro pontos nas categorias Mais Claro e Mais Escuro, não será possível haver mais de 12 *pixels* consecutivos nas categorias, logo o ponto pode ser descartado. Caso contrário, torna-se necessário conferir os *pixels* restantes para definir o ponto (ROSTEN; PORTER; DRUMMOND, 2008).

Esse algoritmo é capaz de gerar diversos ponto, tendo muitas vezes pontos muito próximos uns aos outros. Para evitar quaisquer problemas com tais pontos, suprime-se alguns pontos menos relevantes. Pode-se utilizar qualquer função heurística capaz de identificar o ponto menos relevante. A equação (3.9) traz um exemplo de função.

$$V = \max \begin{cases} \sum I_K - I_P & , \text{ para a categoria Mais Claro} \\ \sum I_P - I_K & , \text{ para a categoria Mais Escuro} \end{cases} \quad (3.9)$$

Ruble et al. (2011) utilizam esse processo descrito, escolhendo o raio $R = 9$, considerado por eles como um tamanho com boa resposta. Adicionalmente, como o método FAST não ordena os pontos-chave, os autores decidiram incrementar o método com um algoritmo de ordenação, o método de Harris e Stephens (1988). Esse método tem como resultado um score, chamado de score de Harris, e a ordenação é realizada de forma a organizar os scores em ordem decrescente. O score é calculado multiplicando os autovalores da matriz de Harris obtida a partir da aplicação de uma mascara na imagem original. A matriz de Harris é quadrada e de dimensão 2 e é calculada uma matriz para cada *pixel*. Calcula-se a matriz a partir das derivadas parciais de segunda ordem das intensidades da imagem em relação às coordenadas x e y . A equação (3.10) mostra a estrutura dessa matriz.

$$M = \begin{bmatrix} A & B \\ B & C \end{bmatrix} = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \quad (3.10)$$

Para aprimorar a seleção de pontos, os autores do método sugerem reduzir a quantidade limiar de *pixels* N para gerar mais pontos e após ordená-los com o método de Harris faz uma seleção dos melhores pontos. Por fim, criam-se diversas escalas da imagens original

em forma de pirâmide e o processo completo é realizado com todas as imagens de todas as escalas, buscando mais pontos-chave e a invariabilidade do método quanto a diferentes escalas.

A partir deste ponto, todos os pontos-chave estão determinados e é necessários orientá-los. Para isso, utilizou-se o método Centroide de Intensidade (ROSIN, 1999), criado por Paul R. Rosin. Neste método, Rosin calcula a posição do centroide de intensidade dos *pixels* da região analisada, direção para qual o ponto-chave estará orientado. A equação (3.11) mostra o momento geométrico para o calculo do centroide. Já a equação (3.12) mostra o calculo do centroide baseado nos momentos calculados (ROSIN, 1999).

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (3.11)$$

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3.12)$$

Representa-se a orientação do ponto-chave por um vetor que tem origem no ponto-chave e atinge o centroide. Para se calcular o ângulo do vetor utiliza-se a equação (3.13) (ROSIN, 1999).

$$\phi = \begin{cases} \tan^{-1}\left(\frac{m_{01}}{m_{10}}\right) & , \text{ para a categoria Mais Claro} \\ \tan^{-1}\left(\frac{m_{01}}{m_{10}}\right) + 180^\circ & , \text{ para a categoria Mais Escuro} \end{cases} \quad (3.13)$$

Em seguida, o próximo passo é definir os descritores e para isso os autores utilizam o método BRIEF, desenvolvido por Calonder, Lepetit, Strecha e Fua (CALONDER et al., 2010). Esse descritor realiza uma descrição em forma de sequência de bits construído a partir de um conjunto de testes binários (RUBLEE et al., 2011). A imagem tem sua intensidade suavizada e é realizada uma comparação entre as intensidades de dois *pixels* que estão ao redor do ponto de interesse e uma decisão binária é tomada. A equação (3.14) mostra o descritor binário utilizado, onde τ é o descritor, $p(x)$ é a intensidade do *pixel* no ponto x e $p(y)$ é a intensidade do *pixel* no ponto y (RUBLEE et al., 2011).

$$\tau(p; x,y) := \begin{cases} 1 & , p(x) < p(y) \\ 0 & , p(x) \geq p(y) \end{cases} \quad (3.14)$$

A partir desses testes binários é criado o vetor de características para cada p :

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (3.15)$$

Para concluir, os autores de ORB decidem atribuir rotação aos descritores encontrados com BRIEF. Uma matriz de testes binários nas posições (x_i, y_i) na imagem é criada:

$$S = \begin{pmatrix} x_1 & , & \dots & , & x_n \\ y_1 & , & \dots & , & y_n \end{pmatrix} \quad (3.16)$$

Usando o ângulo ϕ de orientação dos pontos-chave e a matriz de rotação para esse ângulo (R_ϕ) específico, pode-se deduzir a expressão de rotação da matriz S como:

$$S_\phi = R_\phi S \quad (3.17)$$

Dessa forma, o vetor de características é modificado como mostra a equação (3.18)

$$g_n(p, \phi) := f_n(p) | (x_i, y_i) \in S_\phi \quad (3.18)$$

3.3.1.4 Binary Robust Invariant Scalable Keypoints - BRISK

De forma semelhante ao SIFT, SURF e ORB, o BRISK é um algoritmo de detecção e descrição de pontos-chave em imagens. Leutenegger, Chli e Siegart (LEUTENEGGER; CHLI; SIEGWART, 2011) projetaram o algoritmo de forma a ser rápido, computacionalmente eficiente e robusto em relação a transformações geométricas e fotométricas.

O BRISK se destaca por apresentar a possibilidade de determinar a quantidade específica de pontos-chave detectados e possui um mecanismo que permite selecionar as escalas utilizadas para se encaixar melhor às condições das imagens. Além destes pontos, o algoritmo permite modificar o quão complexo serão os descritores gerados para os pontos-chave.

Para iniciar, o algoritmo de Leutenegger, Chli e Siegart (LEUTENEGGER; CHLI; SIEGWART, 2011) utiliza o método FAST para detectar os pontos-chave em toda a imagem, mesmo detector utilizado no método ORB na secção 3.3.1.3. O FAST é aplicado ao plano da imagem e a um espaço escalar com o objetivo de gerar invariância escalar ao modelo em ambos os métodos, porém o espaço escalar é organizado de forma diferente.

Na estrutura do BRISK, O espaço escalar também se dá em formato de pirâmide e apresenta um número n de oitavas (*octaves*) chamadas de c_i e um número n de intra-oitavas (*intra-octaves*) chamadas de d_i , em que $i = 0, 1, \dots, n - 1$. As oitavas são formadas ao se amostrar a oitava anterior pela metade (1/2). As intra-oitavas seguem o mesmo padrão, exceto pela primeira intra-oitava que é amostrada em 1/1,5 vezes (LEUTENEGGER; CHLI; SIEGWART, 2011). Pode-se visualizar tais detalhes na figura 3.30.

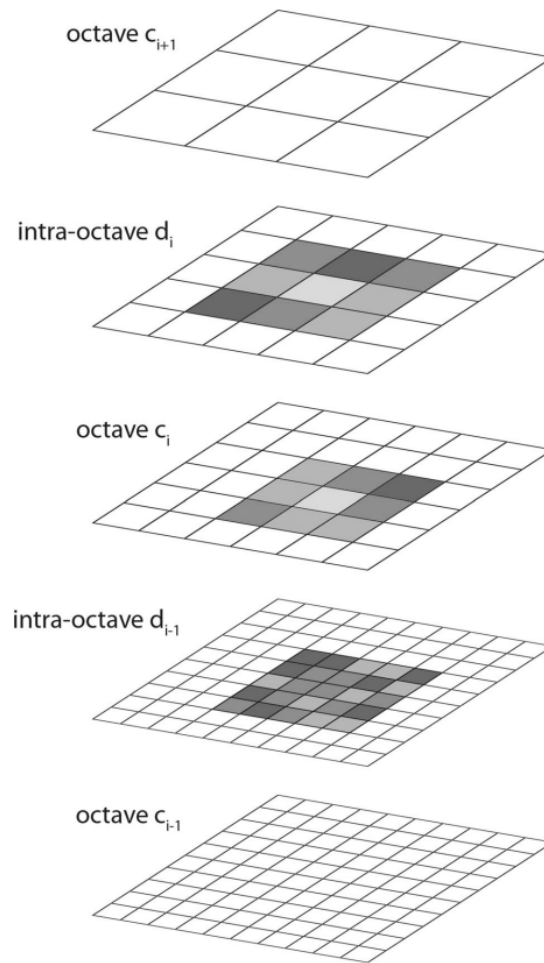


Figura 3.30 – Espaço de Escala BRISK - Fonte: (AZIMI et al., 2019)

Leutenegger, Chli e Siewart (2011) definem as características do FAST para o método como $R = 3$, $K = 16$ e $N = 9$, sendo que deve-se utilizar o mesmo limiar L para todas as camadas. O processo de determinação dos pontos-chave começa com a aplicação do FAST em todas as camadas do espaço de escalas. Para cada camada a pontuação encontrada para o ponto de interesse deve ser superior às pontuações dos seus 8 vizinhos mais próximos, caso contrário, esse ponto é descartado. Os pontos restantes agora são comparados com os seus equivalentes nas camadas adjacentes, imediatamente acima e imediatamente abaixo, caso o ponto não apresente pontuação maior que as camadas adjacente, ele é descartado. O valor do score s para cada ponto restante é calculado como o limiar máximo que ainda considera o ponto como um canto. Para a oitava c_0 há alterações, visto que é a primeira camada e não há camada inferior a ela. Para ela considera-se as características $R = 1$, $K = 8$ e $N = 5$ com o mesmo limiar L das camadas anteriores (LEUTENEGGER; CHLI; SIEWART, 2011).

Para refinar o processo limitando a complexidade, ajusta-se uma função quadrática 2D no sentido dos mínimos quadrados para as três camadas de análise, a camada do ponto de referencia, a camada diretamente superior e a camada diretamente inferior. As três pontuações refinadas e suas escalas são utilizadas para construir uma parábola ao longo do eixo de

escalas, sendo essa a estimativa final do score para o ponto de interesse (LEUTENEGGER; CHLI; SIEGWART, 2011). Por fim, as coordenadas são interpoladas e assim encontra-se a posição e score final para o ponto de interesse (figura 3.31).

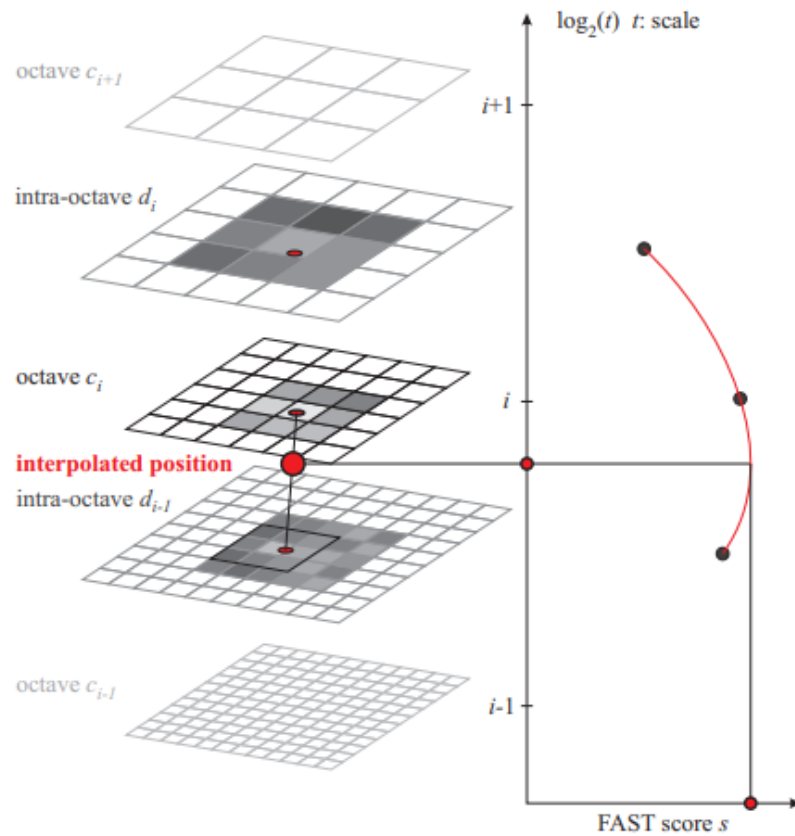


Figura 3.31 – Espaço de Escala BRISK com Respetiva Parábola Encontrada - Fonte: (LEUTENEGGER; CHLI; SIEGWART, 2011)

Em uma próxima etapa deve-se atribuir orientação aos pontos de interesse para que o método se torne invariável à rotação. Esse passo será realizado junto com a construção dos descritores.

Um padrão é criado envolvendo 60 pontos de amostragem, pontos esses que se encontram distribuídos em quatro círculos concêntricos ao redor do ponto de interesse (figura 3.32 letra a). A intensidade de cada ponto é obtida aplicando suavização gaussiana para evitar o aparecimento de efeitos de *aliasing*, de forma que o desvio padrão é proporcional à distância entre o ponto e o centro do círculo (ponto de interesse) (FAN; WANG; WU et al., 2015).

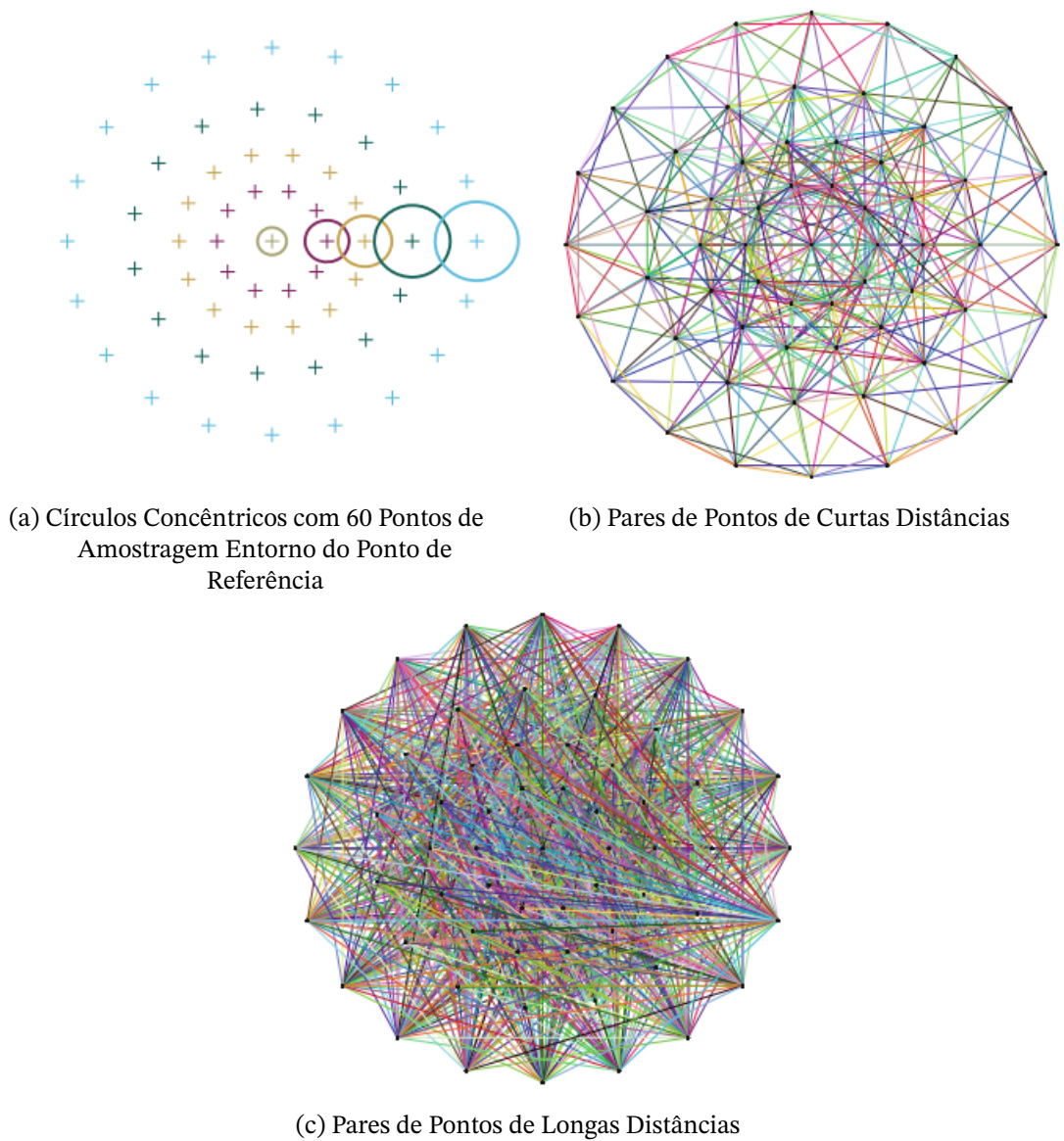


Figura 3.32 – Padrões Utilizados para Determinar os Descritores - Fonte: (FAN; WANG; WU et al., 2015)

Analisando os pontos, é possível calcular o total de combinações de pontos 2 a 2, isto é, $C_2^{60} = \frac{60!}{2!(60-2)!} = \frac{60 \cdot 59 \cdot 58!}{2 \cdot 58!} = \frac{60 \cdot 59}{2} = 1770$ pares de pontos. Essa lista de pares pode ser dividida em duas categorias, pares de curta distância \mathcal{C} (figura 3.32 letra b) e pares de longa distância \mathcal{L} (figura 3.32 letra c). A equação (3.19) mostra a divisão categorias matematicamente, onde δ é a escala onde está o ponto de interesse e $\delta_{max} = 9.75\delta$, $\delta_{min} = 13.67\delta$ (FAN; WANG; WU et al., 2015).

$$\begin{cases} \mathcal{C} = (p_i, p_j) \mid \|p_j - p_i\| < \delta_{max} \\ \mathcal{L} = (p_i, p_j) \mid \|p_j - p_i\| > \delta_{min} \end{cases} \quad (3.19)$$

Com o objetivo de tornar o descritor mais discriminativo e robusto, é interessante

foca-lo em variações de intensidades mais significativas, dessa forma os gradientes de curta distância serão ignorados, baseando os descritores apenas nas longas distâncias. Sendo assim, a orientação do ponto de interesse é definido como a orientação média dos gradientes de pares de longa distância (equação (3.20)) (LEUTENEGGER; CHLI; SIEGWART, 2011).

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{n} \sum_{(p_i, p_j) \in \mathcal{L}} g(p_i, p_j) \quad (3.20)$$

Onde, $g(p_i, p_j)$ é dado por:

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \quad (3.21)$$

Por fim, define-se o ângulo de rotação do descritor em torno do ponto de interesse como $\alpha = \arctan2(g_y, g_x)$. Um vetor descritor binário é criado a partir de comparações das intensidades de curta distância do pares de pontos. Os padrões de amostragem e os limiares de distância permitem gerar sequências binárias de comprimento 512 (LEUTENEGGER; CHLI; SIEGWART, 2011).

3.3.1.5 Homografia

Quando se tenta unir imagens, muitas vezes esbarra-se em um mesmo problema, variação de posição entre as imagens. Seja por defeito no mecanismo, seja por defeito no projétil ou por qualquer motivo, é possível que as imagens estejam tortas uma em relação a outra e é necessário corrigir tal problema antes de gerar uma imagem final, para que a imagem final seja o mais fiel possível ao projétil. Com tal problema em mente, será abordado o conceito de homografia, método capaz de auxiliar com tal problema.

A homografia é um conceito matemático muito utilizado no processamento de imagens onde, para a matemática diz-se plano, para o processamento de imagens diz-se imagem.

Segundo Hartley e Zisserman (2003), a homografia é uma transformação projetiva planar que consegue mapear pontos de um determinado plano para outro plano. Pádua et al. (2020) esclarece que uma transformação projetiva é o equivalente a uma transformação linear invertível, ou seja, é uma transformação que leva pontos de um plano a outro de forma a manter duas propriedades importante, preservação de adição e preservação de escalar, sendo possível mapear os pontos de volta ao plano original.

A figura 3.33 mostra o ponto x pertencente ao plano π que foi mapeado para o plano π' como ponto x' . Pode-se descrever a transformação linear do plano onde x está contido como $x'_n = Hx_n$, sendo H chamado de matriz de homografia que levará todos os pontos do plano x para o plano x' .

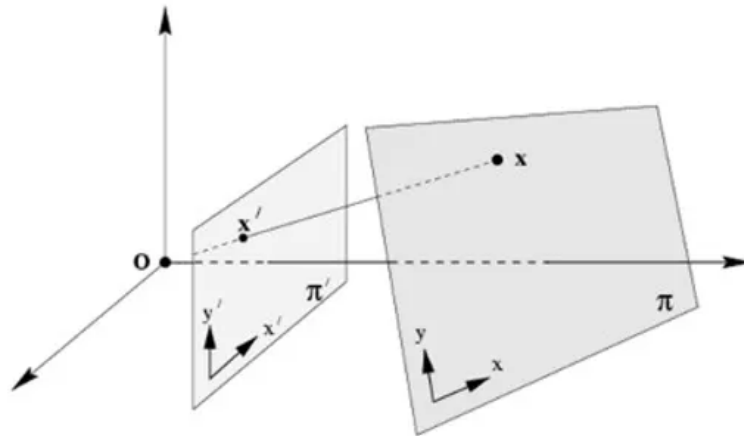


Figura 3.33 – Mapeamento entre Planos - Fonte: (HARTLEY; ZISSERMAN, 2003)

A partir deste ponto, é necessário estimar a matriz H . Considerando o espaço \mathbb{R}^3 e definindo os pontos no plano π e no plano π' como $p_n = (x_n, y_n, z_n, w_n)^T$ e $p'_n = (x'_n, y'_n, z'_n, w'_n)^T$, respectivamente, tem-se a transformação linear descrita anteriormente como $p'_n = Hp_n$ que pode ser expandida como na equação (3.22). Na expressão definida tem-se (x'_n, y'_n, z'_n) e (x_n, y_n, z_n) que representam as coordenadas cartesianas dos pontos nos planos π' e π , e acrescenta-se o termo w'_n e w_n que são coordenadas homogêneas, coordenadas que empregam ao sistema a ideia de perspectiva.

$$p'_n = Hp_n \Rightarrow \begin{bmatrix} x'_n \\ y'_n \\ z'_n \\ w'_n \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \\ w_n \end{bmatrix} \quad (3.22)$$

Como o intuito da homografia neste projeto é trabalhar com imagens, os pontos aplicados apresentam $z_n = 0$ e $z'_n = 0$. Dessa forma, o sistema criado pode ser reduzido como apresentado na equação (3.23).

$$p'_n = Hp_n \Rightarrow \begin{bmatrix} x'_n \\ y'_n \\ 0 \\ w'_n \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ 0 \\ w_n \end{bmatrix} \Rightarrow \begin{bmatrix} x'_n \\ y'_n \\ w'_n \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ w_n \end{bmatrix} \quad (3.23)$$

Com este sistema definido já é possível realizar a homografia, porém o sistema considera planos em mesma escala, logo é preciso adaptar a escala dependendo da transformação desejada. Para isso utiliza-se o fator de escala homogêneo λ . O sistema então passa a ser representado pela equação (3.24), onde $\lambda \neq 0$ e caso os planos já estejam na mesma escala basta considerar $\lambda = 1$ (SANTOS, M. C. dos; ROCHA, A., 2012).

$$\lambda p'_n = H p_n \quad (3.24)$$

O fator de escala nem sempre é conhecido e por tanto é interessante encontrar uma forma de representar o sistema sem depender deste fator. Para isso é realizado o produto vetorial em ambos os lados da equação com p'_n :

$$p'_n \times (\lambda p'_n) = p'_n \times (H p_n) \Rightarrow \lambda (p'_n \times p'_n) = p'_n \times (H p_n) \Rightarrow 0_3 = p'_n \times (H p_n) \quad (3.25)$$

onde $0_3 = (0,0,0)^T$.

Para realizar o produto vetorial, pode-se adaptar o termo $H p_i$. Considerando que se podem representar as linhas da matriz H como $h^{iT} = (h_{i1}, h_{i2}, h_{i3})$, onde i é o número da linha, tem-se a adaptação da seguinte forma:

$$H p_i = \begin{bmatrix} h^{1T} p_n \\ h^{2T} p_n \\ h^{3T} p_n \end{bmatrix} \quad (3.26)$$

Dessa forma, o produto vetorial resulta em:

$$p'_n \times (H p_n) = \begin{bmatrix} w'_n h^{2T} p_n - y'_n h^{3T} p_n \\ x'_n h^{3T} p_n - w'_n h^{1T} p_n \\ y'_n h^{1T} p_n - x'_n h^{2T} p_n \end{bmatrix} = 0_3 \quad (3.27)$$

Considerando a propriedade da transposição da multiplicação $(A^T B)^T = B^T A$, então $(h^{iT} p_n)^T = p_n^T h^i$. Observa-se que, como h^{iT} tem dimensão 1×3 e p_n tem dimensão 3×1 , então a multiplicação deles resultará em um escalar. Sabe-se que $e^T = e$, sendo e um escalar. Dessa forma, conclui-se que $(h^{iT} p_n)^T = h^{iT} p_n \Rightarrow h^{iT} p_n = p_n^T h^i$. Pode-se então reescrever a equação (3.27) no sistema a seguir.

$$\begin{cases} w'_n p_n^T h^2 - y'_n p_n^T h^3 = 0 \\ x'_n p_n^T h^3 - w'_n p_n^T h^1 = 0 \\ y'_n p_n^T h^1 - x'_n p_n^T h^2 = 0 \end{cases} \Rightarrow \begin{cases} 0 \cdot x'_n p_n^T h^1 + w'_n p_n^T h^2 - y'_n p_n^T h^3 = 0 \\ -w'_n p_n^T h^1 + 0 \cdot y'_n p_n^T h^2 + x'_n p_n^T h^3 = 0 \\ y'_n p_n^T h^1 - x'_n p_n^T h^2 + 0 \cdot w'_n p_n^T h^3 = 0 \end{cases} \quad (3.28)$$

Transformando o sistema de volta em uma matriz, tem-se:

$$\begin{bmatrix} 0_3^T + w'_n p_n^T h^2 - y'_n p_n^T h^3 \\ -w'_n p_n^T h^1 + 0_3^T + x'_n p_n^T h^3 \\ y'_n p_n^T h^1 - x'_n p_n^T h^2 + 0_3^T \end{bmatrix} = 0_3 \Rightarrow \begin{bmatrix} 0_3^T & w'_n p_n^T & -y'_n p_n^T \\ -w'_n p_n^T & 0_3^T & x'_n p_n^T \\ y'_n p_n^T & -x'_n p_n^T & 0_3^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0_3 \quad (3.29)$$

Ao analisar a matriz resultante na equação (3.29) percebe-se que a terceira linha é uma combinação linear das duas primeiras linhas. Ao se multiplicar tanto a primeira linha como a segunda por -1 e somá-las resultará na linha 3. Dessa forma, a terceira linha não é linearmente independente, não sendo necessária sua representação visto que não agregará informação nova. Logo, a equação (3.29) pode ser reescrita como:

$$\begin{bmatrix} 0_3^T & w'_n p_n^T & -y'_n p_n^T \\ -w'_n p_n^T & 0_3^T & x'_n p_n^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0_3 \quad (3.30)$$

A equação (3.30) representa a relação dos elementos da matriz de homografia H com um único par de pontos correspondentes entre os planos analisados. Pode-se escalonar essa representação para todos os n pares de pontos como mostra a equação (3.31).

$$\begin{bmatrix} 0_3^T & w'_1 p_1^T & -y'_1 p_1^T \\ -w'_1 p_1^T & 0_3^T & x'_1 p_1^T \\ 0_3^T & w'_2 p_2^T & -y'_2 p_2^T \\ -w'_2 p_2^T & 0_3^T & x'_2 p_2^T \\ \vdots & \vdots & \vdots \\ 0_3^T & w'_n p_n^T & -y'_n p_n^T \\ -w'_n p_n^T & 0_3^T & x'_n p_n^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0_3 \quad (3.31)$$

Expandindo p_n^T , h^i e 0_3^T tem-se:

$$\begin{bmatrix} 0 & 0 & 0 & w'_1 x_1 & w'_1 y_1 & w'_1 w_1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 w_1 \\ -w'_1 x_1 & -w'_1 y_1 & -w'_1 w_1 & 0 & 0 & 0 & x'_1 x_1 & x'_1 y_1 & x'_1 w_1 \\ 0 & 0 & 0 & w'_2 x_2 & w'_2 y_2 & w'_2 w_2 & -y'_2 x_2 & -y'_2 y_2 & -y'_2 w_2 \\ -w'_2 x_2 & -w'_2 y_2 & -w'_2 w_2 & 0 & 0 & 0 & x'_2 x_2 & x'_2 y_2 & x'_2 w_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & w'_n x_n & w'_n y_n & w'_n w_n & -y'_n x_n & -y'_n y_n & -y'_n w_n \\ -w'_n x_n & -w'_n y_n & -w'_n w_n & 0 & 0 & 0 & x'_n x_n & x'_n y_n & x'_n w_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.32)$$

Segundo Hartley e Zisserman (2003), uma matriz de transformação entre dois pontos homogêneos também é considerada homogênea, isto é, ela apresenta oito elementos independentes referentes a transformação linear e um elemento homogêneo referente ao espaço projetivo. Com essa definição se diz que a matriz H apresenta oito graus de liberdade.

Ao analisar a equação (3.32), nota-se que as incógnitas são apenas os elementos da matriz H ($h_{ij} \mid j, i \in \{1,2,3\}$). Sendo assim, há nove incógnitas, sendo oito independentes e uma referente ao espaço projetivo. Dessa forma, apenas oito incógnitas precisam ser encontradas, a incógnita restante será responsável pelo refinamento da transformação considerando o espaço projetivo.

Com esse conceito em mente, compreende-se que é apenas necessário que a matriz a esquerda na equação (3.32) contenha oito linhas para que se encontre a matriz de homografia H . Observa-se que cada par de pontos gera duas linhas na matriz, isto é, precisa-se apenas de quatro pares de pontos para se encontrar a matriz de homografia H .

3.3.2 Medidas de Similaridade e Distâncias para Comparação de Características

No processamento de imagens, muitas vezes, tem-se a necessidade de comparar matematicamente vetores espaciais de diversas dimensões com o intuito de concluir possíveis similaridades entre características questionadas. No caso do presente projeto, as similaridades que se deseja obter estão relacionadas com a união das imagens e com a determinação de pontos em comum entre as imagens, processo extremamente importante neste projeto.

Existem diversos métodos para realizar tais comparações, porém, devido ao escopo do projeto, foram escolhidas algumas formas específicas de comparação, sendo elas:

- Distância Euclidiana
- Distância de Hamming
- Distância de Manhattan
- Distância de Minkowski

Como mencionado anteriormente, imagens são sinais bidimensionais espaciais, ou seja, os métodos mencionados podem ser relacionados com processamento de imagens e de sinais unidimensionais. Em seção futura será mencionada a forma como tais métodos são úteis para o processamento de sinais unidimensionais que serão realizados.

3.3.2.1 Distância Euclidiana

A distância Euclidiana, inicialmente, trata-se de uma distância no espaço chamado de Euclidiano, ou seja, a distância entre dois pontos em um espaço tridimensional (O'NEILL, 2006). É possível notar que, quanto maior a distância entre dois pontos no espaço, mais diferentes eles se tornam, quanto mais próximos, mais similares são suas características.

Com o passar do tempo notou-se a possibilidade de utilizar tal cálculo para distâncias entre pontos em n dimensões. Dessa forma, é possível comparar a similaridade entre vetores de quaisquer tamanhos, desde que os vetores comparados apresentem a mesma dimensão (O'NEILL, 2006).

Considere a figura 3.34, nela é possível observar os pontos p e q .

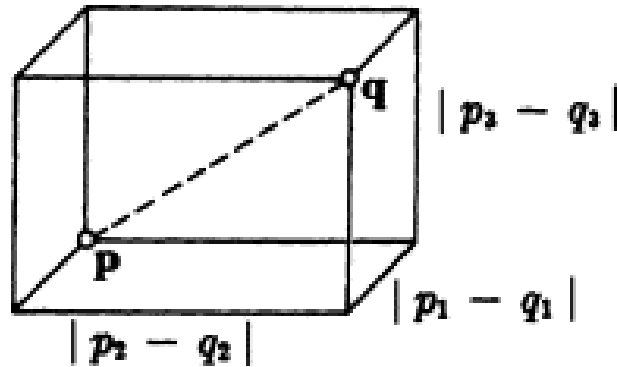


Figura 3.34 – Distância Euclidiana - Fonte: (O'NEILL, 2006)

A distância entre os pontos p e q pode ser escrita como $d(p,q) = \|p - q\|$. Sabendo que $p - q = (p_1 - q_1, p_2 - q_2, p_3 - q_3)$, deduz-se a equação (3.33) para a distância Euclidiana (O'NEILL, 2006)].

$$d(p,q) = \|p - q\| = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (3.33)$$

Levando a equação (3.33) para o espaço n dimensional, a distância euclidiana pode ser escrita como mostra a equação (3.34) (O'NEILL, 2006).

$$d(p,q) = \|p - q\| = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (3.34)$$

3.3.2.2 Distância de Hamming

Distância de Hamming é definida como o número de posições em que dois vetores se diferenciam (THEODORIDIS; KOUTROUMBAS, 2006). Em outras palavras, Hamming mede o número necessário de alterações a serem feitas em um vetor para que ele se torne idêntico ao vetor com o qual foi comparado. Para que tais comparações sejam possíveis, é necessário que os vetores comparados apresentem a mesma dimensão.

Ishengoma define a distância de Hamming pela equação (3.35) (ISHENGOMA, 2014).

$$\text{Distância de Hamming} = \sum_{j=1}^N x_j(\text{XOR})y_j \quad (3.35)$$

A figura 3.35 exemplifica uma comparação entre dois vetores por distância de Hamming. Os vetores são nomeados de A e B , ambos com dimensão 10. Ao serem comparados, a distância de Hamming resultou no valor 3.

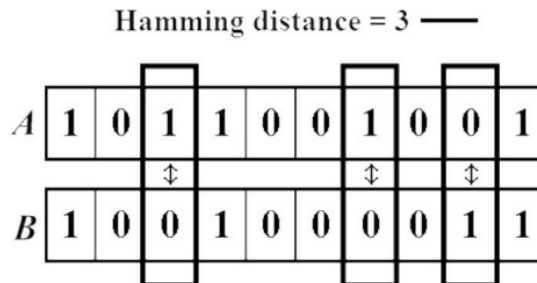


Figura 3.35 – Exemplo Distância de Hamming Igual a 3 - Fonte: (ISHENGOMA, 2014)

3.3.2.3 Distância de Manhattan

A distância de Manhattan se assemelha à distância Euclidiana, uma vez que ambas tratam da diferença entre os termos correspondentes no vetores comparados. Segundo Szabo (2015), a distância de Manhattan entre dois vetores pode ser descrita como a norma L1 da diferença entre os vetores.

Boyd e Vandenberghe descrevem a normal L1 como a soma dos valores absolutos dos componentes do vetor (BOYD; VANDENBERGHE, 2004). A equação (3.36) demonstra a norma L1.

$$\|x\|_1 = |x_1| + |x_2| + \dots + |x_n| \quad (3.36)$$

Portanto, pode-se escrever a distância de Manhattan como é descrito na equação (3.37).

$$\text{Distância de Manhattan} = \sum_{i=1}^n |x_i - y_i| = |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \quad (3.37)$$

3.3.2.4 Distância de Minkowski

Segundo Sharma (2020), a distância de Minkowski nada mais é do que uma forma generalizada da distância Euclidiana e da distância de Manhattan. A equação (3.38) mostra como Sharma descreve matematicamente a distância de Minkowski.

$$\text{Distância de Minkowski} = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} = (|x_1 - y_1|^p + |x_2 - y_2|^p + \dots + |x_n - y_n|^p)^{1/p} \quad (3.38)$$

Na equação (3.38), p representa a ordem da norma aplicada. É possível reparar que caso tenha-se norma igual a 1 ($p = 1$), a distância de Minkowski representará a distância de Manhattan e quando a norma for igual a 2 ($p = 2$), a distância de Minkowski se igualará a distância Euclidiana.

3.3.3 Processamento de Sinais

Processamento de sinais pode ser descrito como uma ciência de analisar processos físicos segundo So e Chan (1999). Ele está presente em diversas áreas como medicina, sistemas de controle, telecomunicações, processamento de áudio, de imagem e de vídeos, entre outros (ROCHA, A. F. da et al., s.d.).

Visão computacional e processamento de sinais se relacionam no meio computacional, concentrando-se na análise de dados visuais. A relação entre essas duas esferas científicas resulta principalmente em desenvolvimento de algoritmos e técnicas para adquirir, processar, analisar e compreender imagens e vídeos.

As etapas básicas do processamento de sinais no escopo da visão computacional podem ser vistas na figura 3.36. Essa etapas dizem respeito ao processamento de sinais analógicos por meios digitais, porém o processo para sinais digitais é o mesmo partindo do ponto de digitalização do sinal.

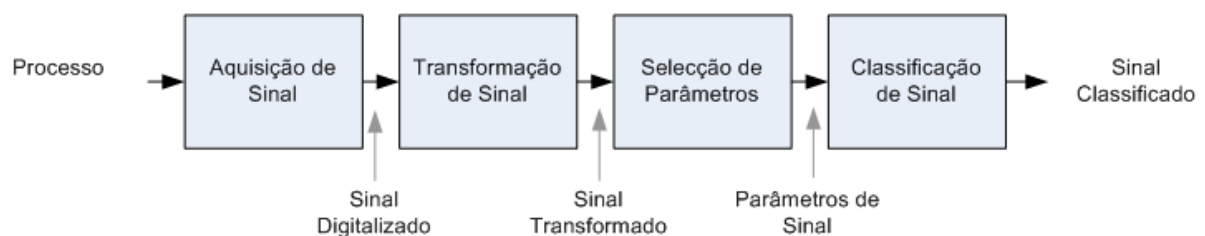


Figura 3.36 – Etapas do Processamento de Sinais - Fonte: (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006)

A aquisição de sinal pode ser de forma digital ou analógica. Na forma analógica, geralmente os sinais são elétricos e são convertidos em digitais por meio de um conversor analógico digital (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006).

A transformação de sinais consiste na aplicação de filtros para a redução e possivelmente eliminação de ruídos, além da eliminação de informação não útil como redundâncias e informações extras (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006).

A seleção de parâmetros diz respeito à aquisição de pontos relevantes e parâmetros úteis para a classificação (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006). Aqui é recorrente a aplicação de tratamentos ao sinal para evidenciar pontos de interesse (ROCHA, A. F. da et al., s.d.).

Classificação de sinal, por fim, traz a interpretação do sinal, reconhecimento de padrões e conclusões (FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, 2006). Em geral as conclusões finais são levadas para o mundo real, fora das telas, como indicar unidades defeituosas na esteira de uma sistema de controle de qualidade, ou indicar regiões suspeitas de câncer em uma imagem de raio-X (GRANLUND; KNUTSSON, 2013).

Para chegar à etapa de classificação é necessário passar primeiramente pela etapa de seleção de parâmetros, isto é, o pré-requisito para realizar uma classificação é ter pontos de interesse, características (*features*) (GRANLUND; KNUTSSON, 2013). Existem diversas formas de se analisar e classificar tais *features*, é necessário escolher a forma mais adequada para o tipo de informação que está sendo trabalhada.

O presente trabalho necessita de processamento de sinais para conseguir realizar comparações entre os projéteis. Uma forma de representar os projéteis é por meio de sinais extraídos com base na intensidade da iluminação de acordo com as ranhuras presentes em um projétil (essa ideia será melhor explanada no capítulo seguinte). Tendo extraído os sinais dos projéteis, o ideal é conseguir compará-los sabendo que podem estar defasados e que nunca serão exatamente iguais, apenas bem próximos ou nada próximos. Por este motivo, escolheu-se trabalhar com os dois tipos de classificadores listados a seguir:

- Autocorrelação e Correlação Cruzada
- Comparação Estatística

3.3.3.1 Autocorrelação e Correlação Cruzada

A autocorrelação trabalha com sinais temporais e com a relação estabelecida entre eles. Para Menke (MENKE, 2022), a ideia da autocorrelação está na correlação existente entre duas amostras retiradas da mesma série temporal e que apresentam uma diferença temporal fixa (atraso).

A função de autocorrelação (FAC) entre dois sinais $u(t)$ e $u(t + \tau)$ é definida como (AGUIRRE, 2004):

$$r_{uu}(\tau, t) = E[u(t)u^*(t + \tau)] \quad (3.39)$$

Para efeitos práticos, pode-se considerar o processo como real, ou seja, $y^*(t) = y(t)$. Ainda, Aguirre (AGUIRRE, 2004) sugere que o processo seja considerado ergódico, isto é, é possível analisar o processo apenas uma única vez, pois ele não muda com o tempo. Dessa forma, pode-se substituir a esperança matemática pela média temporal resultando na equação (3.40).

$$r_{uu}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u(t)u(t + \tau)dt \quad (3.40)$$

Uma vez que as amostras que foram retiradas da série temporal apresentam um atraso uma em relação à outra, pode-se dizer que os sinais comparados não são iguais, ou seja, abre portas para a busca de correlação entre sinais retirados de séries diferentes, a função de correlação cruzada (FCC). Considerando dois sinais $u(t)$ e $y(t)$, a FCC pode ser escrita como mostra a seguir:

$$r_{uy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u(t)y(t + \tau)dt \quad (3.41)$$

No caso discreto pode-se escrever as funções de autocorrelação e correlação cruzada respectivamente como:

$$r_{uu}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{i=-N}^N u(i)u(i + k) \quad (3.42)$$

$$r_{uy}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{i=-N}^N u(i)y(i + k) \quad (3.43)$$

Para ambas as funções, k é considerado um número inteiro e N deve apresentar um valor elevado para que a aproximação seja válida (AGUIRRE, 2004).

Dessa forma, a correlação cruzada pode ser utilizada para medir o grau de similaridade entre dois sinais e assim podem-se extrair informações essenciais dependendo de cada aplicação. Alguns exemplos de aplicação são medições em radares, sonares, comunicações digitais, geologia e diversas áreas da ciência e da engenharia (PROAKIS, 2007).

3.3.3.2 Comparação Estatística

A comparação estatística pode ser utilizada para determinar se existem diferenças ou semelhanças significativas entre dois sinais. Qualquer tipo de sinal pode ser descrito por meio de medidas estatísticas e oito delas podem ser ditas as principais:

- Média

A média é a soma de todos os valores de um determinado conjunto dividida pela quantidade de elementos somados. Seu resultado apresenta um número que se refere ao centro do conjunto de dados usados. A equação (3.44) mostra como se calcula a média.

$$\bar{x} = \frac{\sum_{i=1}^n x(i)}{n} \quad (3.44)$$

Na equação (3.44), \bar{x} representa a média, $x(i)$ representa os termos do conjunto x e n a quantidade de elementos do conjunto.

- Variância

A variância nada mais é que a medida de quanto os valores do conjunto se afastam da média, indicando a dispersão deles no conjunto. A equação (3.45) mostra como se calcula a variância.

$$s^2 = \frac{\sum_{i=1}^n (x(i) - \bar{x})^2}{n - 1} \quad (3.45)$$

Na equação (3.45), s^2 representa a variância, $x(i)$ representa os termos do conjunto x , \bar{x} representa a média e n a quantidade de elementos do conjunto.

- Desvio Padrão

O desvio padrão, assim como a variância, calcula o quanto os valores do conjunto se afastam da média já que é tido como a raiz quadrada da variância. A equação (3.46) mostra melhor como se calcula o desvio padrão.

$$s = \sqrt{\frac{\sum_{i=1}^n (x(i) - \bar{x})^2}{n - 1}} \quad (3.46)$$

Na equação (3.46), s representa o desvio padrão, $x(i)$ representa os termos do conjunto x , \bar{x} representa a média e n a quantidade de elementos do conjunto.

- Mediana

Ao se organizar um conjunto em ordem crescente ou decrescente, a mediana se torna o valor do elemento central do conjunto para os casos onde há um número ímpar de elementos, ou se torna a média entre os dois elementos centrais para os casos onde há um número par de elementos.

$$Med(x) = \begin{cases} x(\frac{n+1}{2}) & , \text{ se } n \text{ for ímpar} \\ \frac{x(\frac{n}{2}) + x(\frac{n}{2}+1)}{2} & , \text{ se } n \text{ for par} \end{cases} \quad (3.47)$$

Na equação (3.47), $Med(x)$ representa a mediana, $x(i)$ representa os termos do conjunto x e n a quantidade de elementos do conjunto.

- Amplitude

Amplitude é a variação máxima que pode haver dentro do conjunto, ou seja, é a diferença entre o valor do maior e o valor do menor elementos do conjunto.

$$Amp(x) = x_{max}(i) - x_{min}(i) \quad (3.48)$$

Na equação (3.48), $Amp(x)$ representa a amplitude, $x(i)$ representa os termos do conjunto, onde x_{max} é o maior valor que $x(i)$ assume e x_{min} é o valor mínimo que $x(i)$ assume.

- Coeficiente de Variação

O coeficiente de variação calcula a variabilidade relativa entre desvio padrão e média.

$$CV(x) = \frac{s}{\bar{x}} \quad (3.49)$$

Na equação (3.49), $CV(x)$ representa o coeficiente de variação, \bar{x} representa a média e s representa o desvio padrão.

- Skewness

A skewness é capaz de medir o quão assimétrica é a distribuição dentro do conjunto.

$$\tilde{\mu}_3 = \frac{\sum_{i=1}^n (x(i) - \bar{x})^3}{(n - 1) \cdot s^3} \quad (3.50)$$

Na equação (3.50), $\tilde{\mu}_3$ representa a skewness, $x(i)$ representa os termos do conjunto x , \bar{x} representa a média, n a quantidade de elementos do conjunto e s é o desvio padrão.

- Curtose

A curtose indica o quão achatada foi a curva de distribuição nos elementos do conjunto em relação à distribuição normal.

$$Kurt = \frac{\sum_{i=1}^n (x(i) - \bar{x})^4}{(n - 1) \cdot s^4} - 3 \quad (3.51)$$

Na equação (3.51), $Kurt$ representa a curtose, $x(i)$ representa os termos do conjunto x , \bar{x} representa a média, n a quantidade de elementos do conjunto e s é o desvio padrão.

4 Metodologia

No capítulo atual, os materiais e metodologias aplicadas serão descritos e detalhados, evidenciando as escolhas feitas e mudanças e adaptações realizadas.

4.1 Dispositivo

Um dispositivo foi desenvolvido para que fosse possível realizar confrontos balísticos preliminares em campo. Para isso, o dispositivo deve ser pequeno, portátil, prático e de fácil uso. Com as aplicações definidas para os dispositivos, é possível definir as características estruturais, mecânicas, eletrônicas e computacionais do mesmo.

4.1.1 Captura da Imagem

Para iniciar o processo de confronto é necessário que haja uma imagem completa da face lateral, face curva, do projétil. Esse fato traz algumas complicações, primeiramente é necessário incluir no dispositivo um microscópio ou câmera capaz de registrar com detalhes as micro ranhuras. Além disso, é necessário registrar imagens do projétil em diversas posições já que o mesmo se assemelha a um tipo de cilindro, não sendo possível fotografá-lo por completo em uma só foto.

Tendo em mente a intenção de criar um projeto acessível, apresentando um baixo custo, escolhe-se duas possibilidades para a captura de imagens, uma câmera *WebCam* e um microscópio digital.

4.1.1.1 Câmera

Em busca de uma câmera ou microscópio para a obtenção das imagens, muitas pesquisas foram realizadas e deparou-se com a utilização de *WebCams* como microscópios (SATURNO, 2018). Na fonte mencionada, essa característica surge quando se inverte a lente da *WebCam*, ou seja, as suas faces trocam de lado, a face virada para o objeto fica em frente ao sensor da câmera e a face que estava em frente ao sensor fica em frente ao objeto.

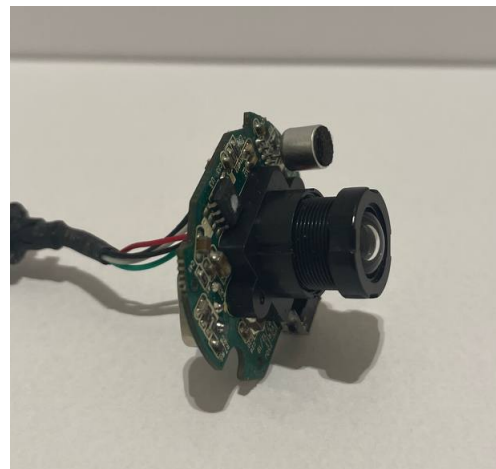
Câmeras são consideradas instrumentos de projeção que utilizam o efeito da câmera escura de orifício onde no lugar do orifício há uma lente convergente. Como discutido na seção 3.1.1, lentes convergentes convergem os raios de luz, ou seja, uma imagem grande se torna uma imagem pequena que, no caso das câmeras, é registrada por sensores fotossensíveis. Como foi explicitado na seção 3.1.1, é possível inverter o efeito da lente ao invertê-la, uma imagem que antes era reduzida agora é ampliada na mesma proporção.

Dessa forma, com apenas a inversão das lentes de uma câmera é possível ampliar imagens como microscópios digitais fazem, porém sem possibilidades de refino e mudança significativa de ampliação. É necessário saber a posição exata para fixar a lente para que a ampliação apresente tamanho suficiente, sendo possível adaptar algum mecanismo de movimentação para auxiliar no foco.

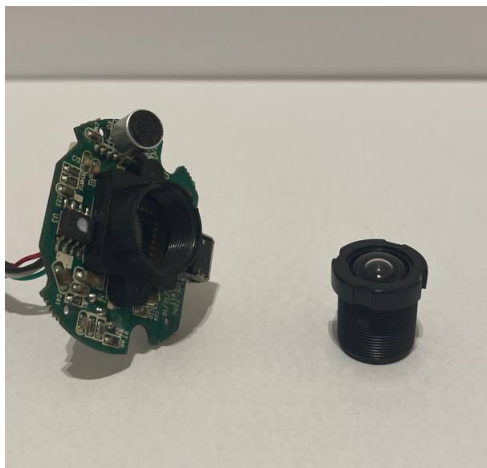
Com esse pensamento, escolheu-se a WebCam A4Tech PK-750MJ, que além ser uma câmera com uma lente removível, é um dispositivo com conectividade a computadores que possibilita a captura de imagens de forma computadorizada. É possível vê-la nas figura 4.37.



(a) WebCam Fechada - Fonte: (BUMAGA-S, 2012)



(b) WebCam Aberta - Fonte: Autor



(c) WebCam Aberta com a Lente Retirada - Fonte: Autor



(d) WebCam Aberta Modificada para Microscópio - Fonte: Autor

Figura 4.37 – Imagens da WebCam Escolhida

A lente da WebCam foi invertida e fixada provisoriamente com fita isolante, bloqueando a entrada de luz pela lateral e deixando a lente fixa o suficiente para testes iniciais. Testou-se a qualidade da câmera ao se inverter a lente e constatou-se uma boa qualidade de imagem, porém é necessário colocar o objeto na posição perfeita de foco para que algo

seja visível com qualidade. Para melhor avaliar a câmera, faz-se necessário compreender o tamanho da ampliação.

Como as características da lente da *WebCam* são desconhecidas, é necessário utilizar uma imagem com ampliação tendo o conhecimento das dimensões reais do objeto ampliado. A partir deste ponto, com uma regra de três simples (equação (4.1)), é possível determinar a ampliação do microscópio.

$$\frac{\text{Dimensão Real Medida}}{1} = \frac{\text{Dimensão Ampliada Estimada}}{\text{Ampliação Estimada}} \quad (4.1)$$

Para a equação (4.1), "Dimensão Real Medida" será o tamanho original de uma das dimensões do objeto e o "1" mostra que a dimensão real não apresenta ampliação. Já a "Dimensão Ampliada Estimada" será o tamanho estimado na imagem capturada, utilizando a mesma dimensão medida anteriormente para realizar a estimação. E por fim, a "Ampliação Estimada" será a incógnita a se descobrir, mostrando o quando a lente ampliou o objeto.

Realizando a conta, encontra-se o valor aproximado de 100 vezes de ampliação. A ampliação é considerável, podendo ou não viabilizar os confrontos balísticos que se desejam realizar. Portanto, é necessário testar com um projétil real.

Foi liberado o acesso a dois projéteis os quais foram utilizados para testar a câmera de todas as formas possíveis. Em primeiro teste as imagens ficaram bem nítidas, porém obteve-se uma ampliação muito maior do que o necessário, não sendo funcional para o projeto. A câmera foi modificada, a lente foi afastada do sensor fotossensível, várias tentativas foram aplicadas, porém a câmera se mostrou bem sensível e pouco flexível, inviabilizando o prosseguimento do projeto com ela.

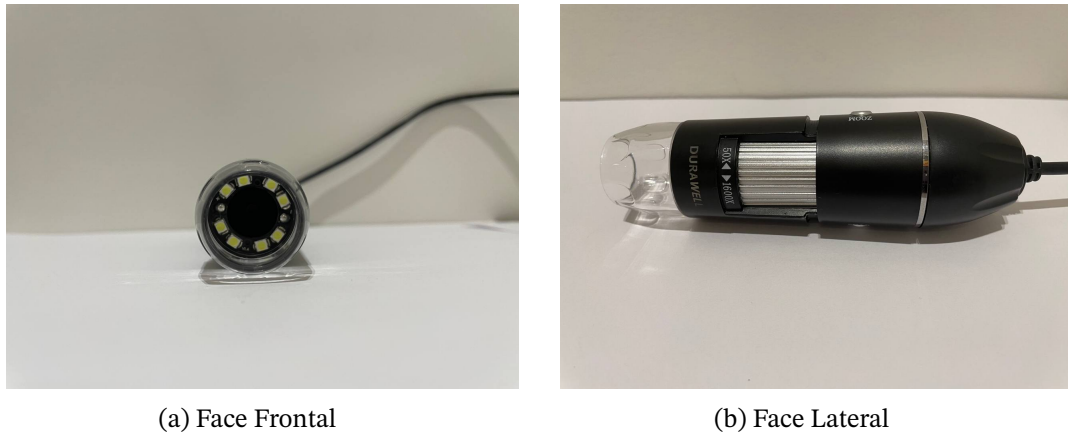
4.1.1.2 Microscópio Digital

Como segunda opção, havia sido definido o microscópio como solução para a captura das fotos. Como ponto de partida para encontrar um microscópio, utilizou-se o aumento calculado para a câmera (*WebCam*), de 100 vezes. Com base ele, estimou-se que era necessário encontrar algum dispositivo que apresenta pelo menos cerca de 5 a 20 vezes menos ampliação. Outro pré-requisito estabelecido como base no sistema proposto foi a possibilidade de controlar o microscópio por conexão USB de forma a facilitar a comunicação geral.

Diversas pesquisas foram realizadas e, com o intuito de continuar a perseguir um dos pré-requisitos estabelecidos no início do projeto, escolheu-se um microscópio com preço relativamente baixo dentre as possibilidades. Levou-se em consideração o nível de aumento requerido, escolhendo um dispositivo que oferece a possibilidade de regular o aumento, sendo que esse aumento pode variar facilmente de 2 a 100 vezes, mesmo que o dispositivo indique uma escala diferente. Outro ponto importante são as dimensões, como o dispositivo

foi pensado para ser portátil, é necessário que o microscópio tenha pequeno porte e seja leve, além de poder ser alimentado exclusivamente pelo computador.

A figura 4.38 mostra o microscópio escolhido, um dispositivo genérico criado para uso com placas de circuito impresso e afins.



(a) Face Frontal

(b) Face Lateral

Figura 4.38 – Imagens do Microscópio Escolhido - Marca PalyShop Eletrônicos e Modelo Play16 -
Fonte: Autor

Alguns testes foram realizados quanto ao aumento da imagem, a facilidade de uso do dispositivo, para regular e realizar todos os ajustes necessários. O microscópio apresenta uma ajuste manual para o aumento da imagem, visto na figura 4.38 letra b. Ele apresenta um CD para instalar um software capaz de manipular as imagens enquanto utiliza o dispositivo, porém é possível utilizá-lo apenas como uma câmera portátil acoplada ao computador.

Após todos os teste, o dispositivo foi aprovado para uso no projeto.

4.1.1.3 Mecanismo de Rotação do Projétil

Pela necessidade de capturar imagens de diversas posições do projétil, surgem duas opções de imediato, pode-se circular o microscópio em torno do projétil capturando as imagens aos poucos, ou pode-se rotacionar o projétil em torno de seu próprio eixo.

Por fins de praticidade, estabilidade e facilidade, escolheu-se fixar o projétil ao eixo de um motor para que fosse rotacionado em torno de seu próprio eixo enquanto o microscópio captura as imagens.

Para uma boa qualidade de imagem é preciso que as capturas sejam bem ampliadas, logo ocupam pequena área, assim, o motor utilizado precisa ser capaz de girar pequenos graus com uma boa precisão. Para tal feito, escolheu-se um motor de passo capaz de realizar 32 passos a cada 360° e que tem uma redução de 64 para 1, ou seja, ao final o motor é capaz de assumir 2048 (32×64) posições diferentes no eixo de saída, o que é equivalente a aproximadamente 0.176° de giro entre cada posição, trazendo excelente precisão ao dispositivo.

O motor de passo escolhido foi o motor 28byj-48 (KIATRONICS, s.d.) e ele foi adquirido já com uma placa *driver* (placa ULN2003) para realizar seu controle com facilidade, sendo que essa placa necessita de um microcontrolador para receber os comandos que serão efetuados no motor. O motor e a placa de *driver* podem ser vistos na figura 4.39.

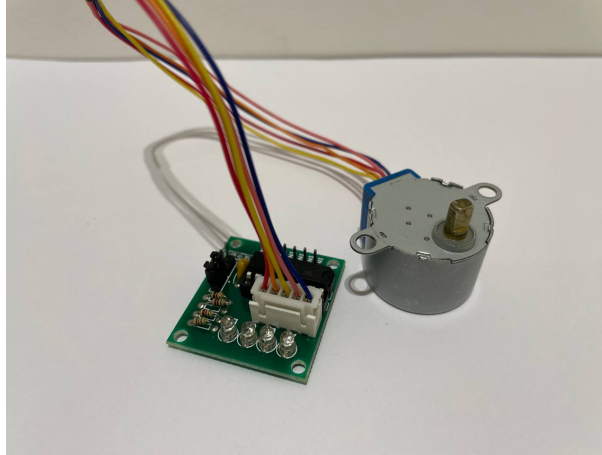


Figura 4.39 – Motor de Passo e sua placa de driver - Fonte: Autor

O motor 28byj-48 apresenta tensão de operação 5 volts, não necessitando de fonte externa para alimentá-lo. Ainda, ele apresenta torque de 34.3 mN.m (KIATRONICS, s.d.), e sabendo as características físicas do objeto que será rotacionado, é possível calcular a aceleração que o motor vai desempenhar, sendo possível avaliar se o motor conseguirá mover o objeto desejado.

A ideia desenvolvida e que será melhor abordada em seção futura, é a utilização de um mandril para fixar o projétil e rotacioná-lo. Para determinar se o motor é capaz de mover o mandril junto com o projétil, será considerado o mandril junto ao projétil como um cilindro de 6 cm de diâmetro, 10 cm de altura, sendo feito de aço maciço, isto é, com seu volume completamente preenchido.

A equação (4.2) mostra como é calculado o torque do motor, sendo que T é o torque, a é a aceleração e I é o momento de inércia. Este último está descrito na equação (4.3), onde M é a massa do objeto e R seu raio. A partir destas equações e sabendo as características do objeto que será rotacionado, é possível calcular a aceleração do objeto.

$$T = I \cdot a \quad (4.2)$$

$$I = \frac{1}{2} M \cdot R^2 \quad (4.3)$$

Utilizando o valor de $7860 \text{ Kg}/\text{m}^3$ como a densidade do aço que constitui o mandril, e sabendo suas dimensões, calcula-se uma massa de 4,4447 Kg para o mandril. Conhecendo este valor de massa e o raio do cilindro, calcula-se o momento de inércia igual a 0,002

$Kg \cdot m^2$. Com este cálculo feito e sabendo o torque que o motor apresenta, pode-se calcular a aceleração do mesmo, que resultou em $17,149 \text{ rad/s}^2$. Essa aceleração é condizente com o esperado e mostra que rapidamente o motor atingirá sua velocidade constante de rotação. Caso a massa do mandril seja menor, a aceleração será maior, chegando-se mais rápido ainda à velocidade constantes. Sendo assim, pode-se dizer que o motor escolhido não terá problemas para realizar a tarefa proposta.

4.1.1.4 Controlador

Para que o sistema funcione, é necessário que haja um controlador para que seja possível controlar o mecanismos de rotação ao mesmo tempo que as fotos serão capturas, além de controlar a iluminação e quaisquer aspectos necessários.

Alguns pré-requisitos foram estabelecidos para a escolha da placa. É necessário poder controlar e alimentar o motor com a placa para que não necessite de fonte externa, apenas o computador. Da mesma maneira, é interessante poder controlar e alimentar a iluminação com a placa. A placa deve ser de pequeno porte, além de o preço também ser um fator a se preocupar, visto o pré-requisito de projeto de baixo custo.

Foi realizada uma análise, alguns controladores foram cogitados, entre eles a ESP32 que apresenta conexão *wifi*, conexão *bluetooth*, portas analógica, portas digitais, necessita apenas de alimentação via USB, apresenta portas o suficiente, tem baixo custo e porte pequeno. Outro controlador cogitado foi o Arduíno da classe UNO, classe concorrente em preço e tamanho ao ESP32. O UNO apresenta portas suficientes, sendo algumas analógicas e outras digitais, ele apresenta alimentação via USB como suficiente, além de ser de porte pequeno e baixo custo.

Para a escolha, além dos aspectos mencionados, foi levado em consideração a facilidade de acesso ao controlador, com esse motivo em peso, foi escolhido o Arduíno UNO, o qual pode ser visto na figura 4.40.

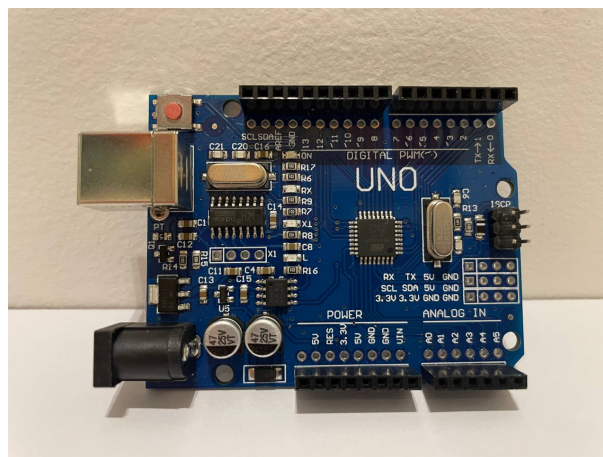


Figura 4.40 – Arduíno UNO - Fonte: Autor

4.1.2 Estrutura

Com o sistema de captura de imagens e o mecanismo de rotação do projétil definidos, é preciso montar um suporte para que os componentes sejam fixados e assim seja possível realizar a captura das imagens da forma desejada.

Para a montagem do sistema existem duas possibilidades, procurar peças prontas para utilizar e adaptar ao projeto, ou criar suas próprias peças de forma a poder modifica-las quantas vezes necessárias, criar peças com encaixes específicos e próprios para o contexto do seu uso. Fica evidente que, em geral, encontrar peças prontas e adaptá-las é um processo mais fácil, rápido e menos custoso, porém não foi a escolha.

No cenário atual, o contato com a manufatura aditiva está se difundido, esta tecnologia está se tornando mais acessível. Com isso em mente, a estratégia tomada foi modelar em 3D cada peça necessária para a estrutura e imprimi-las em uma impressora 3D. Assim, foi possível atingir com mais facilidade e precisão a estrutura desejada, além de ter sido possível realizar pequenas modificações no projeto para melhorá-lo com o tempo e de acordo com os testes realizados.

4.1.2.1 Fixação do Projétil

Escolheu-se começar a organizar a estrutura do comparador a partir do motor de passo. Sendo um elemento central do sistema, para onde todos os outros componentes estarão focados, é interessante que seja fixado primeiro e, posteriormente, as outras peças serão modeladas e acrescentadas conforme for mais conveniente.

Inicia-se, então, pela fixação do motor de passo. O suporte do motor deve seguir alguns parâmetros, ele deve ser robusto de forma a fixar bem o motor evitando uma folga que possa gerar um jogo mecânico na estrutura, isto é, possa haver uma movimentação do motor dentro do suporte, mesmo que quase imperceptível. Para atingir este objetivo, foram escolhidas duas maneiras simultâneas de fixar o motor, por encaixe e por parafusos. O suporte foi modelado para que o motor encaixe nele de forma justa e ainda possa ser fixado com parafusos nas laterais. Ainda, foram acrescentados furos para parafusar o suporte em uma base e deixá-lo estável e imóvel. Um exemplar dessa peça pode ser visualizado na figura 4.41.

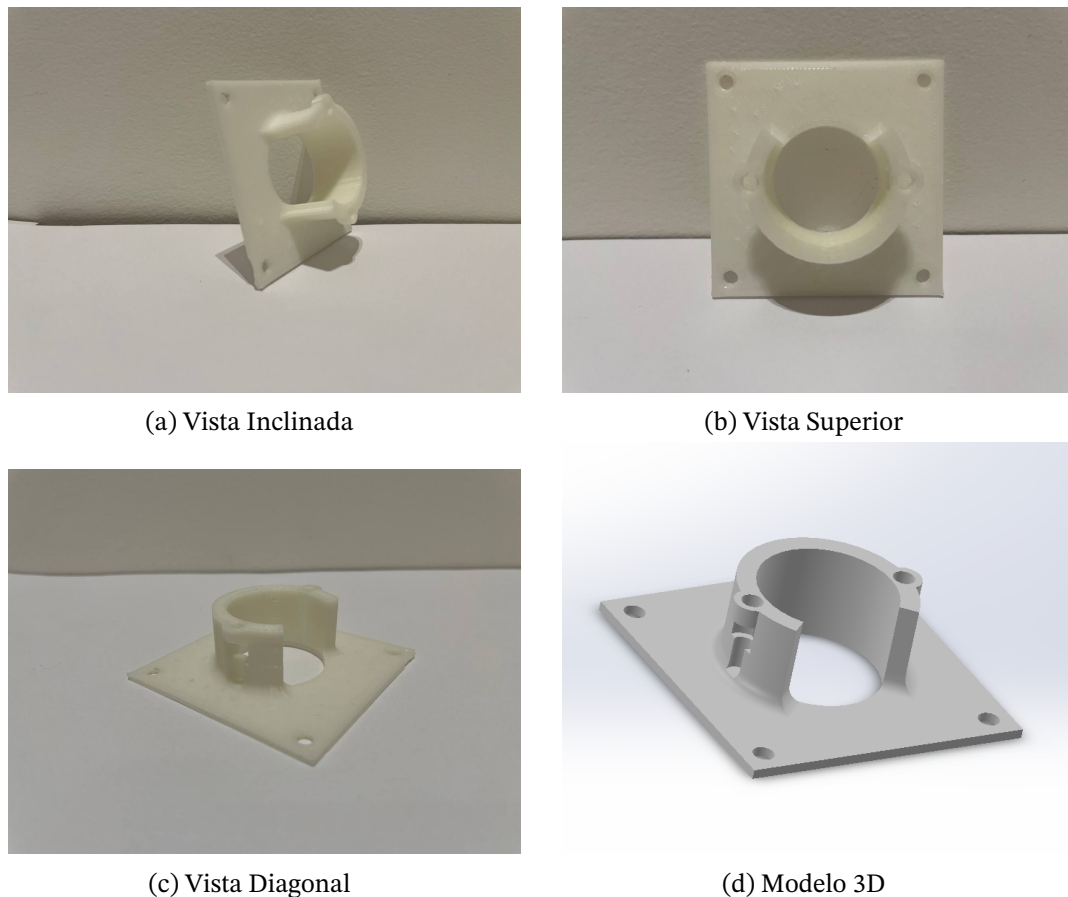


Figura 4.41 – Suporte do Motor de Passo em Algumas Vistas e seu Modelo 3D - Fonte: Autor

O próximo passo poderia ser a fixação do microscópio, porém o mesmo depende da posição em que o projétil se encontra para que fique alinhado corretamente. Sendo assim, é necessário, primeiramente, definir a fixação para os projéteis.

Um ponto importante na fixação dos projéteis é a variabilidade de modelos e tamanhos de armas e consecutivamente de modelos e tamanhos dos projéteis. Para isso, uma solução prática foi encontrada, a utilização de uma espécie de mandril, objeto usado em tornos para fixar o material que será torneado. Com um mandril, é possível ajustar a abertura do suporte para fixar tamanhos variados de projéteis. O problema que se encontra no caminho é a utilização de roscas e engrenagens no mecanismo. A resolução de uma impressão 3D pode chegar à qualidade suficiente para que roscas e engrenagens funcionem bem, porém o acesso a uma impressora com tal resolução demanda muitos recursos.

Por esse motivo, no lugar de utilizar o Modelo 3D do mandril, escolheu-se utilizar um mandril metálico. O Mandril escolhido apresenta uma variação no bocal de 1.5 a 13mm, não possibilitando a utilização de todos os tamanhos de projéteis, alguns modelos ultrapassam tal medida (([AMMUNITION, 2023](#))). Como a maioria do projéteis estão dentro do intervalo de abertura do mandril, este modelo foi suficiente para os testes. A massa do mandril é de aproximadamente 300 gramas e suas dimensões são de 5 cm de diâmetro e 8 cm de altura,

sendo possível rotacioná-lo com o motor escolhido baseado no cálculos feito anteriormente para o motor. O mandril pode ser visto na figura 4.42.

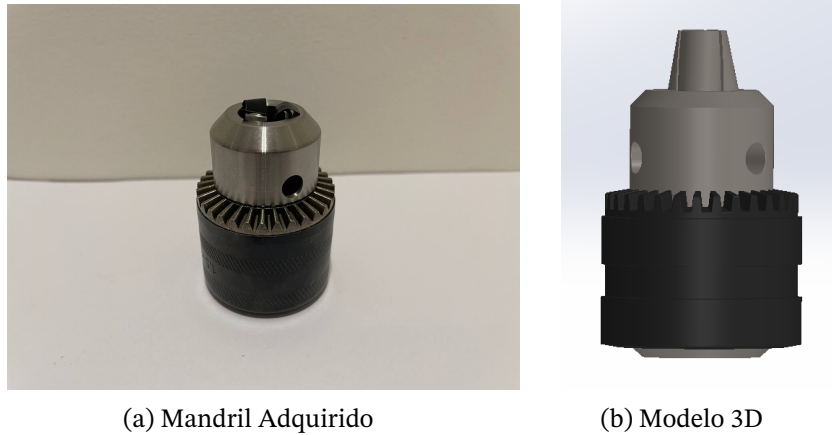


Figura 4.42 – Mandril Adquirido e seu Modelo 3D - Fonte: Autor

Com essas peças definidas, necessita-se conectar o mandril ao motor. Por ser feito de metal, o mandril é consideravelmente mais pesado que o resto das peças, pesando cerca de 300g, como já mencionado. Dessa forma, é necessário criar uma conexão entre o mandril e o motor de forma a apoiar o peso do mandril na base e não no eixo do motor, usando o motor apenas para rodar o mandril, sem impor muito esforço ao eixo do motor, evitando sua inclinação, desalinhamento e possível quebra.

A solução encontrada foi a utilização de um rolamento para apoiar o mandril e conseguir movimentá-lo sem problemas. Para isso é necessário a modelagem de uma peça que se prende junto à base do motor como apoio para o rolamento. Escolheu-se o rolamento modelo 6004-2RS com base em seu tamanho e custo. A peça de apoio foi modelada e impressa em 3D e pode ser vista junta ao rolamento na figura 4.43.

Com o suporte do rolamento definido, pode-se conectar o mandril e o eixo do motor ao rolamento. O mandril apresenta um furo roscado na parte inferior, facilitando prendê-lo utilizando um parafuso. O eixo do motor apresenta chanfros, então é facilmente encaixado em furos apropriados para ele. Já o rolamento não apresenta forma fácil de fixação, principalmente pelo fato de que sua parte interna deve ficar completamente solta com relação a parte externa.

Sendo assim, foi modelada uma peça para que fosse fixada na parte interna do rolamento de forma a entrar justa, com um furo na base no formato chanfrado do eixo do motor para acoplar ao motor, e com um furo na parte superior com formato da cabeça de um parafuso sextavado que foi adquirido para fixar o mandril. Essa peça pode ser vista na figura 4.44.

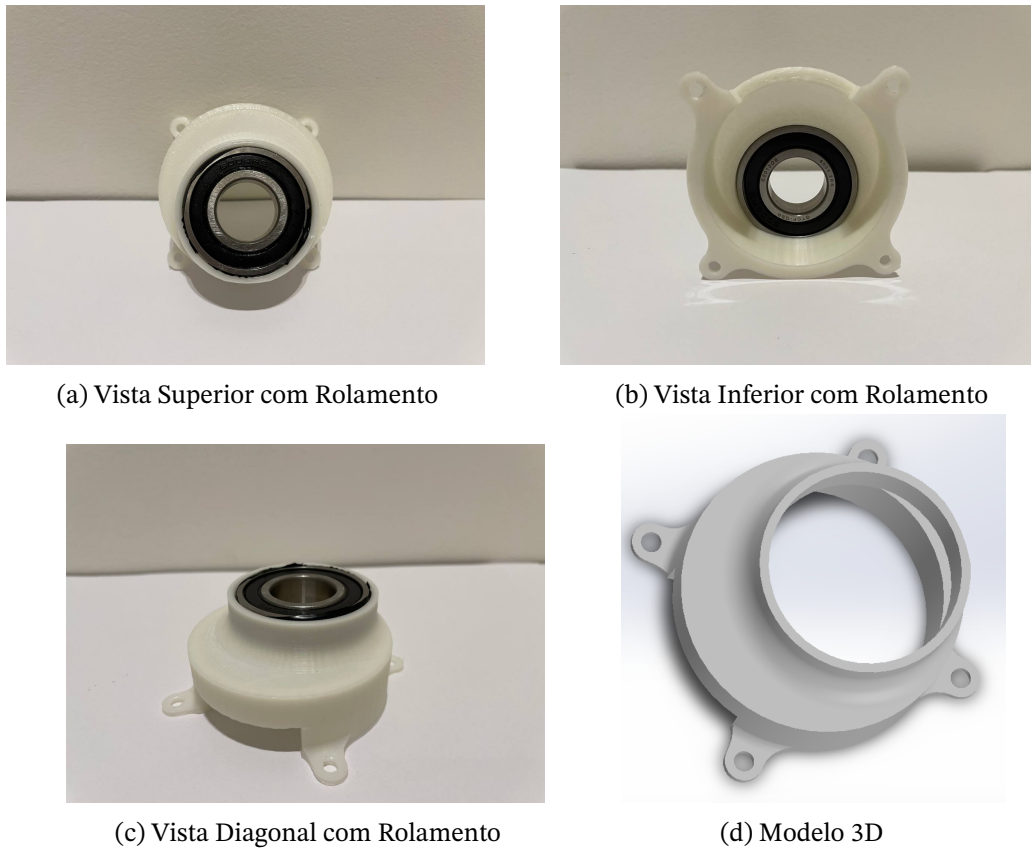


Figura 4.43 – Suporte do Rolamento em Algumas Vistas e seu Modelo 3D - Fonte: Autor

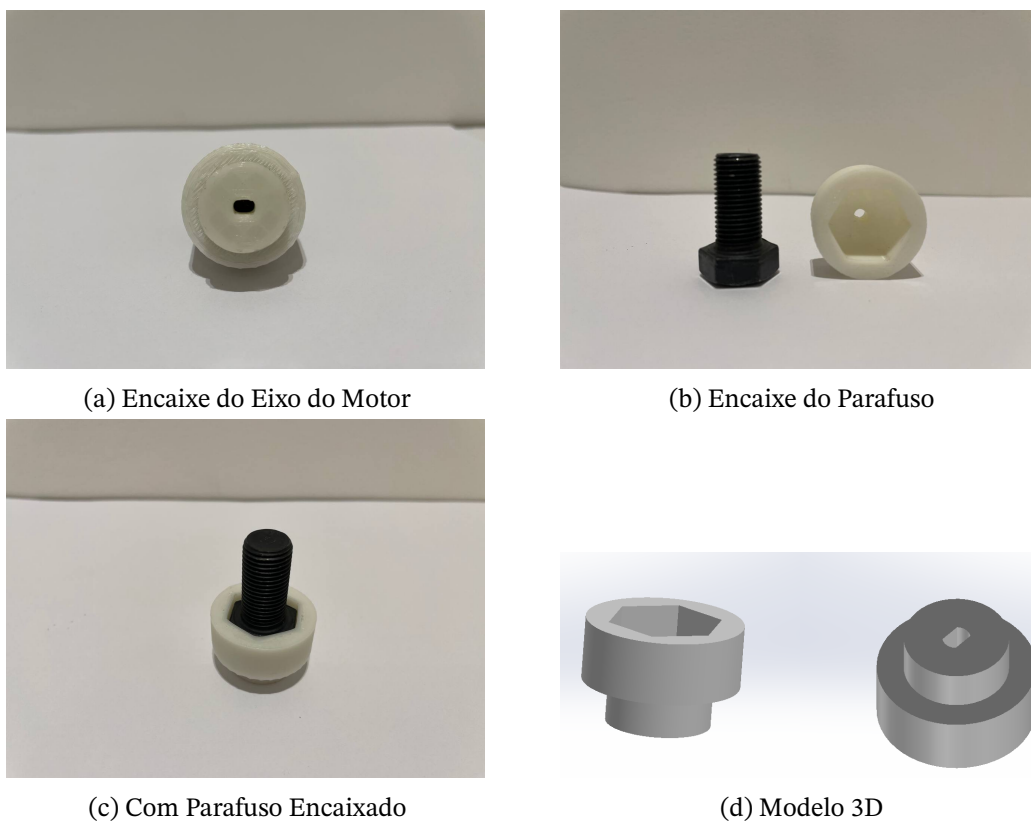


Figura 4.44 – Suporte de Conexão entre Mandril, Rolamento e Motor e seu Modelo 3D - Fonte: Autor

A partir deste ponto, pode-se montar toda a estrutura para girar o projétil (4.45).

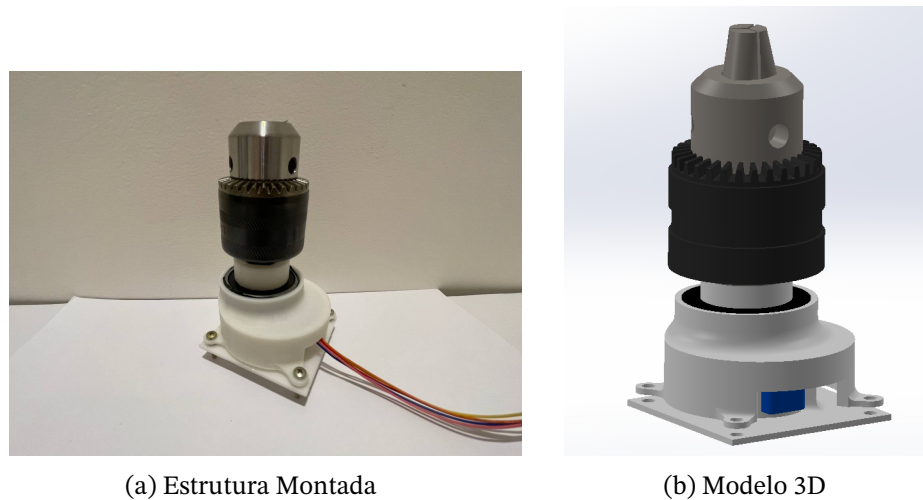


Figura 4.45 – Estrutura Montada para Rotar o Projétil - Fonte: Autor

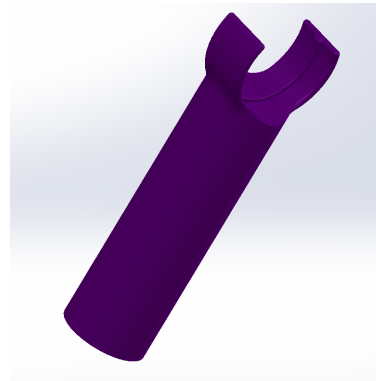
4.1.2.2 Fixação do Microscópio

O próximo passo é criar um suporte para o microscópio, de forma que ele possa ser regulado com relação ao projétil. O microscópio escolhido apresenta regulação de ampliação, ou seja, é possível regular a posição do foco da lente de aumento, sendo necessário aproximar ou afastar a amostra para colocá-la no foco. Ainda, é interessante poder ajustar a altura do microscópio visto que os diversos modelos de projétil apresentam comprimentos variáveis e podem fazer necessário focar em alguma parte específica de cada projétil.

Dessa forma, iniciou-se pelo suporte que sustenta o microscópio. O microscópio apresenta uma espécie de vale circunscrito próximo à região central de seu comprimento. Esse vale é utilizado para prender um suporte próprio, porém este suporte próprio não apresenta as características necessárias para o projeto, então foi desconsiderado. A forma com que tal suporte encaixa no microscópio foi replicada e um modelo foi criado para encaixar na mesma posição. Foi colocada uma haste logo embaixo do encaixe com intuito de usá-la para ajustar a altura. Então outra peça foi modelada e nela foram colocados espaços para fixá-la à base, espaço para o encaixe do microscópio deslizar para ajustá-lo e um espaço para colocar um parafuso junto a uma porca que fazem o papel de travar a peça que segura o microscópio na altura desejada. A figura 4.46 mostra ambas as peças mencionadas.



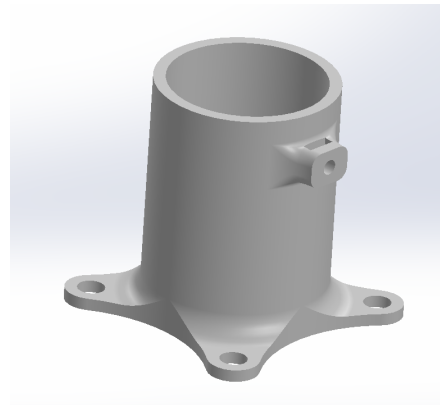
(a) Encaixe no Microscópio



(b) Modelo 3D do Encaixe no Microscópio



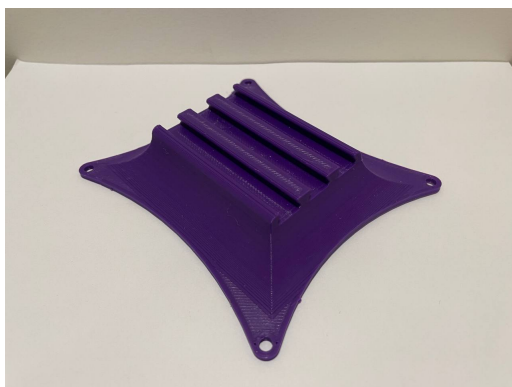
(c) Base de Ajuste de Altura



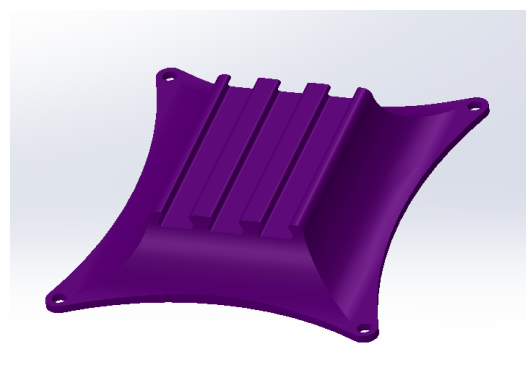
(d) Modelo 3D da Base de Ajuste de Altura

Figura 4.46 – Suporte do Microscópio e seu Modelo 3D - Fonte: Autor

Após determinar a sustentação do microscópio e o ajuste da sua altura, é necessário determinar como será feito o ajuste de aproximação entre microscópio de projétil. Como já há movimentação do microscópio, decide-se criar uma base para o suporte dele para que seja possível movê-lo. Então criou-se uma base com um trilho com cavas para servir de guia para a movimentação do microscópio. Nele há três cavas para prover o máximo de estabilidade ao conjunto. A figura 4.47 mostra essa peça.



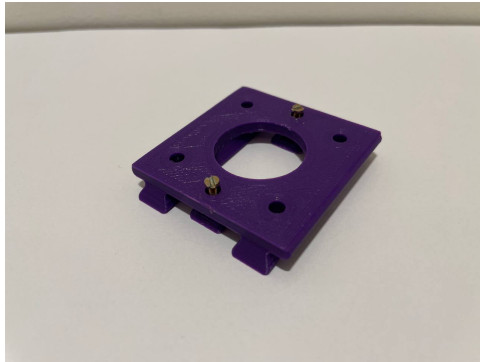
(a) Trilho



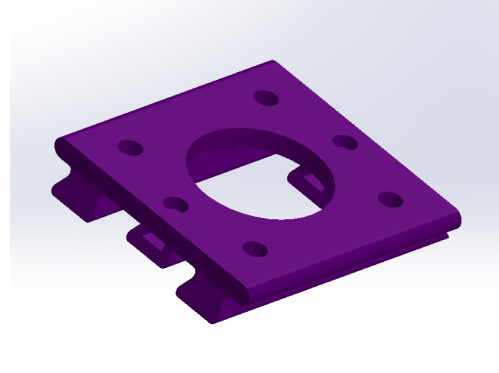
(b) Modelo 3D

Figura 4.47 – Trilho para ajustar a Distância do Microscópio ao Projétil - Fonte: Autor

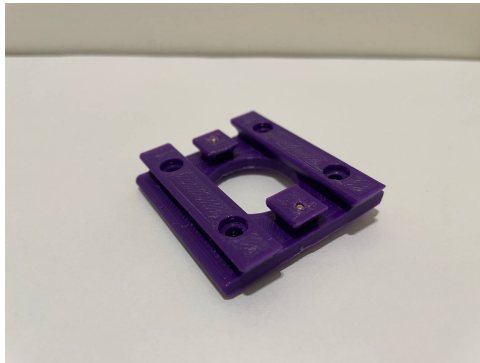
Outra base que encaixa no trilho e pode percorrê-lo foi modelada, essa com furos para prender o suporte do microscópio e movimentá-lo junto. No trilho foram feitos encaixes para dois parafusos que, junto com porcas, fazem o travamento entre os dois trilhos para fixar o microscópio. Essa peça está visível na figura 4.48.



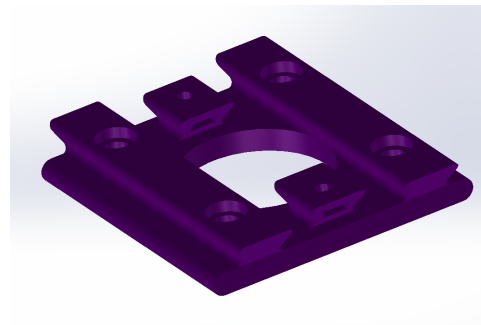
(a) Vista Inclinada Superior



(b) Modelo 3D da Vista Inclinada Superior



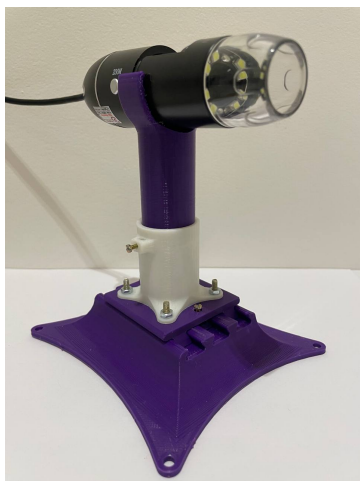
(c) Vista Inclinada Inferior



(d) Modelo 3D Vista Inclinada Inferior

Figura 4.48 – Suporte do Microscópio e seu Modelo 3D - Fonte: Autor

A partir deste ponto, pode-se montar toda a estrutura para posicionar o microscópio (4.49).



(a) Estrutura Montada



(b) Modelo 3D

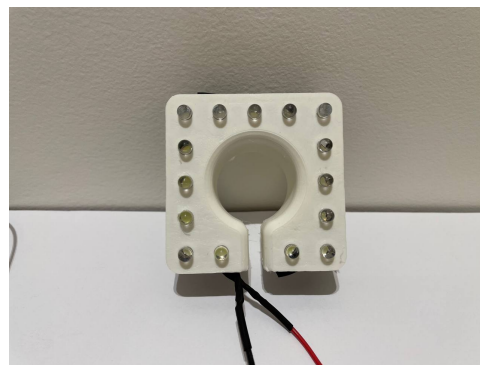
Figura 4.49 – Estrutura Montada para Posicionar o Microscópio - Fonte: Autor

4.1.2.3 Iluminação

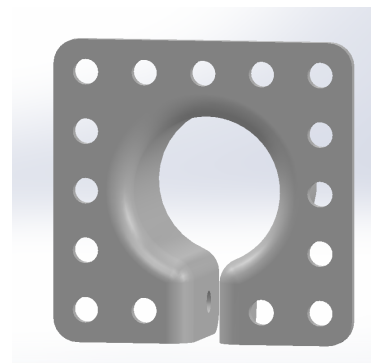
A última necessidade é iluminar o projétil utilizando a ideia de campos escuros vista no seção 3.1.2, isto é, iluminar o projétil de forma que a luz que atinja as lentes do microscópio sejam unicamente provenientes do reflexão no projétil. Necessita-se ainda, que a luz seja uniformemente distribuída para que as fotos apareçam com a iluminação mais homogênea possível.

Para isso, decidiu-se criar 3 pontos de iluminação, um ponto em volta do microscópio e mais dois ao lado, um à esquerda e outro à direita. Escolheu-se LEDs como fonte de luz por serem pequenos, de baixo custo, geram pouco calor e são fáceis de controlar com o Arduino. Os LEDs escolhidos foram os brancos de alto brilho para se ter a melhor iluminação possível, sendo possível ajustar a intensidade dos mesmos.

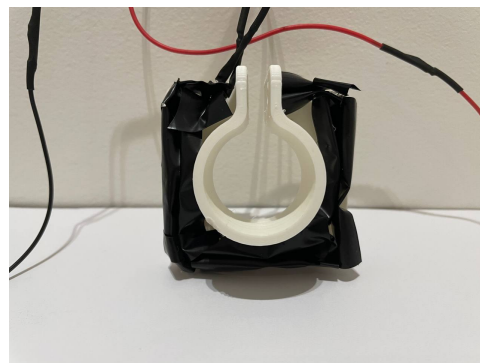
O ponto de iluminação em volta do microscópio necessita estar fixado ao microscópio para que esteja centralizado e não afete a captura das imagens. Para isso, foi modelado um suporte com uma abertura central e inferior, de forma que fosse possível encaixar no microscópio e ajustar a posição. Foi colocado um espaço para passar um parafuso na parte de trás, assim é possível apertar a peça e deixá-la fixa para que não mude de posição durante o processo e assim as imagens mantenham a mesma configuração de iluminação do início ao fim. Essa peça pode ser vista da figura 4.50.



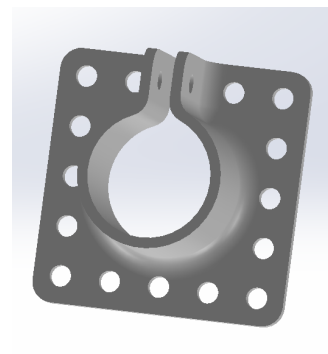
(a) Vista Inclinada Frontal



(b) Modelo 3D da Vista Inclinada Frontal



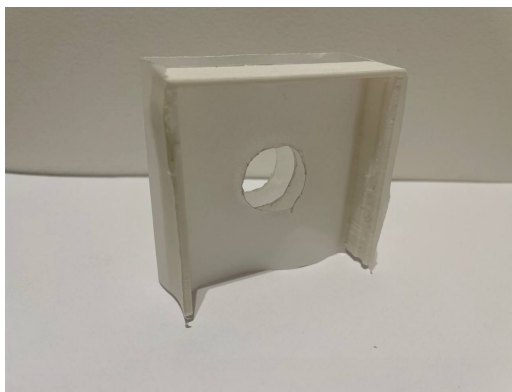
(c) Vista Inclinada Dorsal



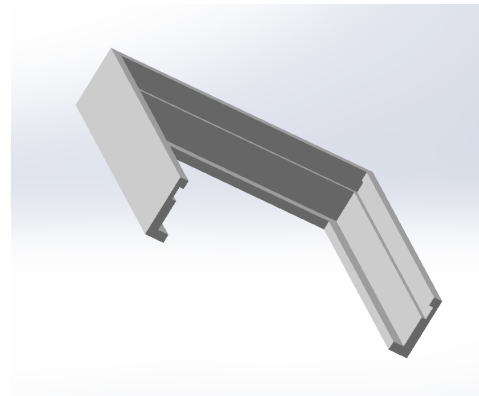
(d) Modelo 3D Vista Inclinada Dorsal

Figura 4.50 – Suporte para LEDs para o Microscópio e seu Modelo 3D - Fonte: Autor

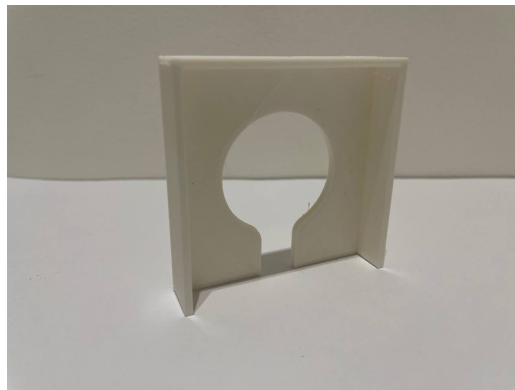
A iluminação será conduzida com LEDs, ou seja, haverá focos de luz fortes em direção ao projétil, o que pode provocar grandes reflexões capazes de ofuscar a lente do microscópio, prejudicando a imagem final. Para que isso seja evitado e para que a luz seja dispersada de forma mais homogênea possível, foi desenvolvida uma peça para encaixar em frente aos LEDs na qual será colocado um papel difusor, podendo colocar mais ou menos camadas de acordo com o necessário. Ainda, como forma de proteger a parte de trás dos LEDs, evitando qualquer problema como curto-circuito, quebra de solda e afins, foi modelada uma peça que encaixa na parte de trás do suporte dos LEDs. Ambas as peças podem ser vistas na figura 4.51.



(a) Suporte do Difusor com o Papel Difusor



(b) Modelo 3D do Suporte do Difusor



(c) Capa



(d) Modelo 3D da Capa

Figura 4.51 – Capa e Difusor para o Suporte de LEDs do Microscópio e seu Modelo 3D - Fonte: Autor

Para a iluminação lateral, é necessário que os suportes sejam idênticos e iluminem da forma mais semelhante possível. É interessante que se possa mover tais suportes para ajustar a posição da luz dependendo do projétil utilizado, tanto na questão da altura como na seção de iluminação. Para isso, foi aproveitada a estrutura da base utilizada no suporte do microscópio e nesse caso ela foi fixada. Então, foi criada uma placa de LEDs e um cilindro com um corte onde a placa de LEDs encaixa. O cilindro encaixa no suporte fixado e pode subir, descer e rotacionar para ajustar a iluminação. Essas duas novas peças estão visíveis na figura 4.52.

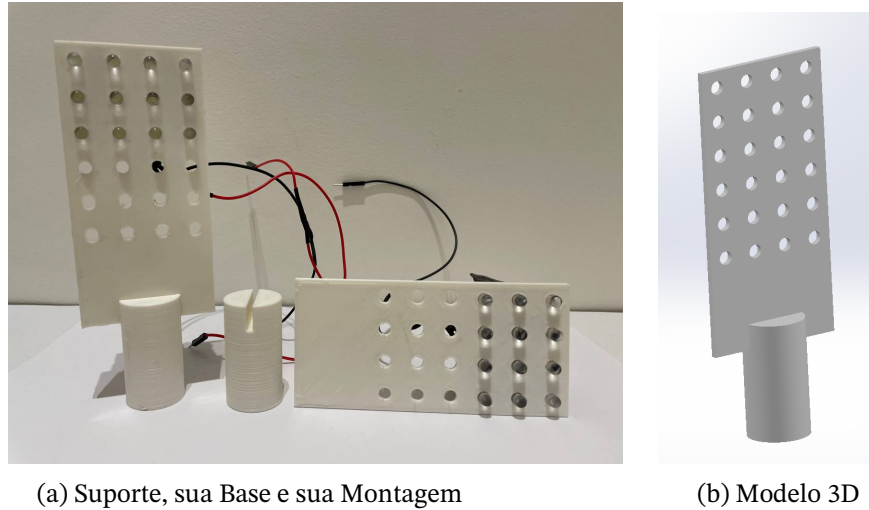


Figura 4.52 – Suporte para LEDs para a Lateral e seu Modelo 3D - Fonte: Autor

Da mesma forma que foi necessário difundir a luz da placa de iluminação do microscópio, é necessário difundir a luz das placas laterais. Para isso, foi modelada uma casca que pode ser encaixada por cima na placa e que, ao mesmo tempo que permite colocar as camadas de papel difusor que forem necessárias, ainda contém uma capa para proteger os LEDs por trás assim foi feito anteriormente. A figura 4.53 mostra tal peça.

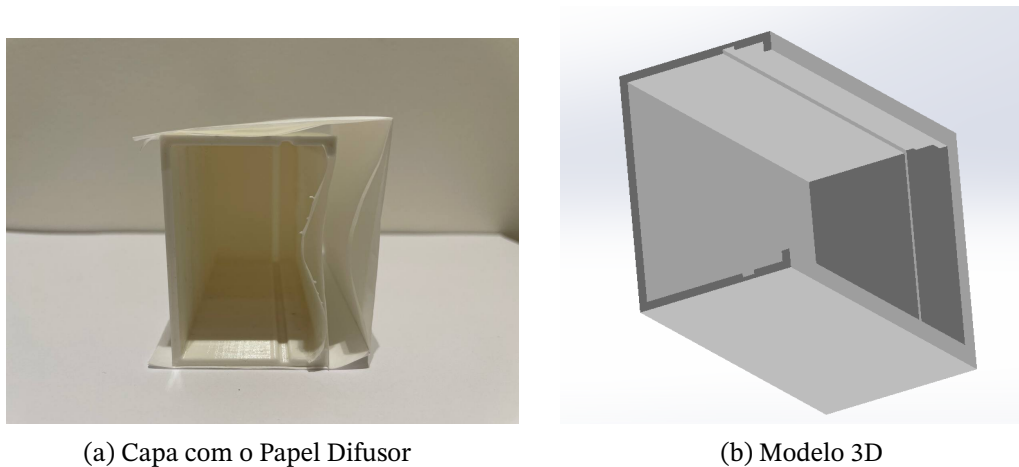
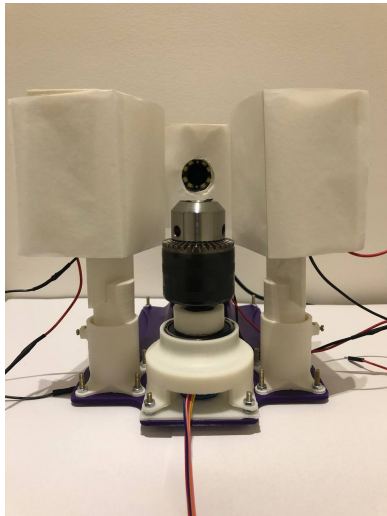
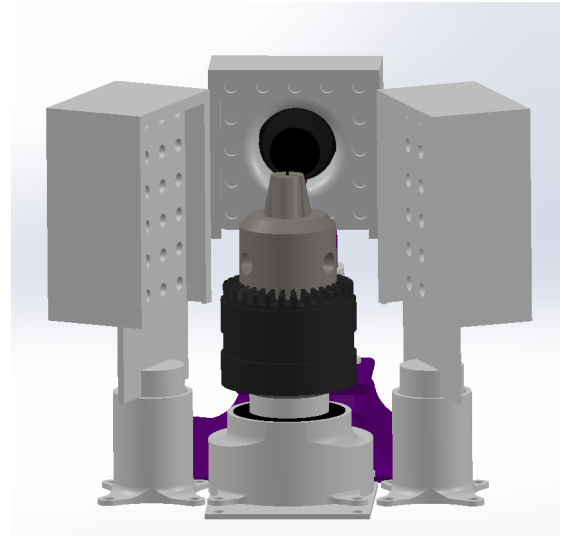


Figura 4.53 – Capa para o Suporte de LEDs da Lateral e seu Modelo 3D - Fonte: Autor

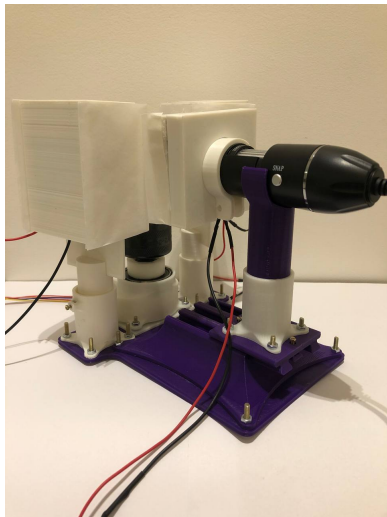
A partir deste ponto, pode-se montar toda a estrutura completa do Comparador (4.49).



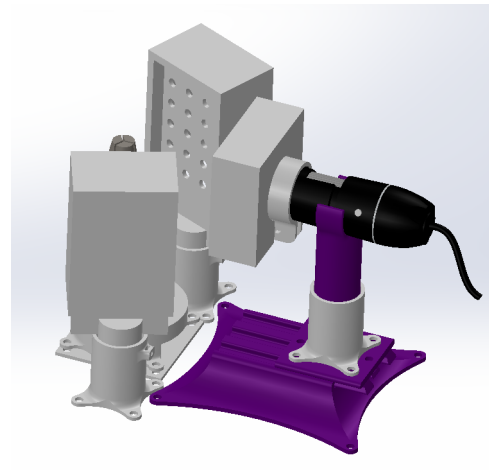
(a) Estrutura Completa Vista Frontal



(b) Modelo 3D da Estrutura Completa Vista Frontal



(c) Estrutura Completa Vista Dorsal/Lateral



(d) Modelo 3D da Estrutura Completa Vista Dorsal/Lateral

Figura 4.54 – Estrutura Completa Montada - Fonte: Autor

4.2 Software

Junto ao dispositivo desenvolvido foi necessário desenvolver um código para realizar a comparação balística. Esse código apresentou pré-requisitos para seu desenvolvimento, deveria ser leve e rápido para que possa rodar em campo e realizar pré-análises em cena fresca.

4.2.1 Acionamento do Motor e Captura de Imagens

Para iniciar todo o processo, primeiramente é necessário capturar as imagens para, posteriormente, trabalhar com delas. Para isso foi escolhido o Arduino UNO e é necessário programá-lo para controlar o motor e os LEDs ao mesmo tempo em que se controla o

microscópio para capturar as imagens. Com o Arduino não seria possível capturar as imagens, portanto a melhor solução encontrada foi capturar as imagens diretamente no computador.

Portanto, decidiu-se programar o Arduino em *python*, abrindo a possibilidade de programar o microscópio simultaneamente. Uma forma viável de realizar essa tarefa foi fazer o *upload* de um código existente na biblioteca de exemplos da IDE (Integrated Development Environment) do Arduino capaz de interpretar códigos em *python* enviados para a placa via terminal.

A partir deste ponto, pode-se utilizar apenas linguagem python em algum editor de texto e compilar o código com um terminal. Então aqui foi criado um código capaz de acionar o motor, ligar os LEDs na intensidade correta (baseado em testes anteriores) e capturar as imagens na posição correta.

O motor de passo 28byj-48 é capaz de assumir 2048 posições diferentes dentro de uma rotação de 360° , como visto na seção 4.1.1.3. Logo, a cada movimento o motor rotaciona $0,1758^\circ$. Com essa informação pode-se controlar exatamente quantos passos o motor irá rodar a cada foto, basta definir qual será essa quantidade.

Baseado no formato aproximadamente circular do projétil, como é possível visualizar na figura 4.55, a imagem capturada contém partes mais distorcidas (representadas pelas partes em vermelho na figura 4.55) devido à curva da superfície do projétil. Uma possível solução para esse problema seria capturar imagens muito próximas ao projétil, com muita ampliação, de forma que a imagem contenha apenas partes pertencentes à área verde da figura 4.55, porém nessa solução perde-se informação vertical, a área capturada é menor e por mais que continue conseguindo juntar imagens e formando um campo horizontal completo, a altura do campo será consideravelmente menor. Por tanto, a solução mais condizente com a situação é fotografar a seção visível completa do projétil e depois retirar as bordas distorcidas mencionadas de forma digital, por meio de código.

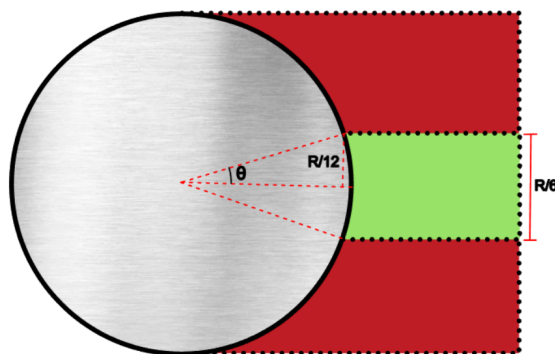


Figura 4.55 – Seção de Visibilidade do Projétil com Medidas - Fonte: Autor

Uma vez definido esse método, é necessário tomar cuidado com a movimentação do motor, para que de uma foto para a outra não haja perda de informação na horizontal. É

preciso que ângulo de movimento seja pequeno para que ainda sobre informação antiga e tenha informação nova na imagem cortada final.

Considerando que a seção verde represente $\frac{R}{6}$ e que o motor deve girar aproximadamente o equivalente a metade dessa distância, então o motor deve percorrer o valor do ângulo θ (Figura 4.55). O cateto oposto ao ângulo θ tem medida de $\frac{R}{12}$ e a hipotenusa mede R , logo, pelas relações trigonométricas, $\theta = 2,39^\circ$.

Sabendo-se disso e de que a cada movimentação o motor de passo gira $0,1758^\circ$, então, por regra de três, para rodar $2,39^\circ$ o motor precisa ser movimentado em 13,6 passos. Para manter a precaução, prefere-se aproximar pelo valor mais baixo, isto é, 13 passos. Dessa forma, a cada 13 movimentações é capturado um *frame*. Dessa forma, sendo ao todo 2048 passos para completar 360° , precisam ser capturadas 157,54 imagens do projétil. Para que não falte nenhuma parte da face lateral, escolheu-se capturar 158 imagens, mesmo que a última movimentação passe dos 360° , podendo-se cortar o excesso digitalmente.

4.2.2 União das Imagens

Após a captura das fotos todo o trabalho físico, que envolve hardware, se encerrou, a partir deste ponto haverá apenas processamento digital até que se chegue a algum resultado.

O primeiro passo a ser dado em seguida é a união das 158 imagens formando a face lateral completa do projétil. Para isso, é necessário identificar pontos específicos entre duas imagens consecutivas que são equivalentes, sendo possível cortar as partes extras das imagens e uni-las corretamente. Nesse processo, muitas vezes, entra a necessidade de corrigir o posicionamento das imagens dado pequenas variações do dispositivo durante o escaneamento do projétil.

4.2.2.1 Determinação de Pontos de Referência

Os pontos de referência são pontos em todas as imagens que apresentam seu equivalente na imagem consecutiva. A tarefa de encontrar tais pontos não é facilmente resolvida de forma manual, então é necessário criar um software que analise as imagens e devolva tais pontos.

Para isso, é necessário determinar a movimentação entre as imagens, isto é, determinar o deslocamento dos *pixels* em comum entre as imagens (SILVINO JUNIOR, 2010). Na figura 4.56, é possível ver que em ambas as imagens tem-se estruturas iguais, porém em posições diferentes. É necessário identificar as regiões semelhantes entre ambas as imagens para que seja possível retirar tais partes.



Figura 4.56 – Duas Imagens com Pontos em Comum - Fonte: Autor

Como discutido na seção 3.3.1, existem diversas formas de se processar imagens, sendo algumas delas formas de identificação de *keypoints*, ou seja, pontos de referência ou interesse que estão presentes na imagem e descrevem um objeto específico, sendo mais facilmente encontrados em outras imagens que contenham os mesmos objetos.

Na seção 3.3.1 foram explorados quatro métodos de seleção de *keypoints*, sendo cada um bem detalhado. Todos são capazes de selecionar pontos específicos que apresentam correspondentes entre as imagens e foram selecionados pois todos apresentam características de invariabilidade, principalmente à rotação, a iluminação e a escala. Essa invariabilidades são importantes visto que o dispositivo, por mais bem construído que seja, sempre apresentará alguma oscilação no eixo do motor podendo haver rotação na imagem, mesmo que mínima, apresentará alguma oscilação de iluminação ou de reflexão dependendo também do material com o qual o projétil foi feito. Sendo assim, é importante garantir que o dispositivo funcionará independente destas variáveis que são muito complicadas de controlar.

Os quatro métodos foram testados para seleção de *keypoints*. Os pontos selecionados foram avaliados conforme a necessidade do projeto, isto é, era necessário uma quantidade relativa de pontos que estivessem dentro de um limiar aceitável de precisão. Com base nisso, escolheu-se o melhor método para continuar o processo de união das imagens.

Seguindo o processo, primeiramente é necessário conseguir comparar duas fotos. Esse processo é realizado por meio de cálculo de medidas de similaridades entre os descritores dos *keypoints* encontrados. Algumas formas de realizar esse cálculo foram descritas na seção 3.3.2. Com as medidas calculadas, cria-se uma lista com as distâncias entre todos os pares comparados, sendo que quanto menor o valor calculado, mais próximo de o par estar correto.

Dessa forma, pode-se descrever o método utilizado como, primeiramente é necessário comparar as duas primeiras imagens e encontrar os pontos de interesse equivalentes entre elas. Faz-se uma rápida avaliação e, em teoria, basta selecionar as melhores equivalências encontradas, as que apresentam mais certeza de compatibilidade, cortar uma faixa a direita da primeira imagem com o corte passando em cima da coluna determinada pelos *keypoints* e cortar uma faixa a esquerda da segunda imagem logo ao lado da coluna equivalente à da

primeira imagem. Em seguida basta unir as imagens cortadas para gerar a imagem final, como é mostrado na figura 4.57. Após essa junção, utiliza-se a nova imagem gerada junto com a próxima foto para fazer a detecção de *keypoints*, corte e união das imagens, fazendo tal processo com todas as 158 imagens até obter como imagem final a face lateral completa do projétil.

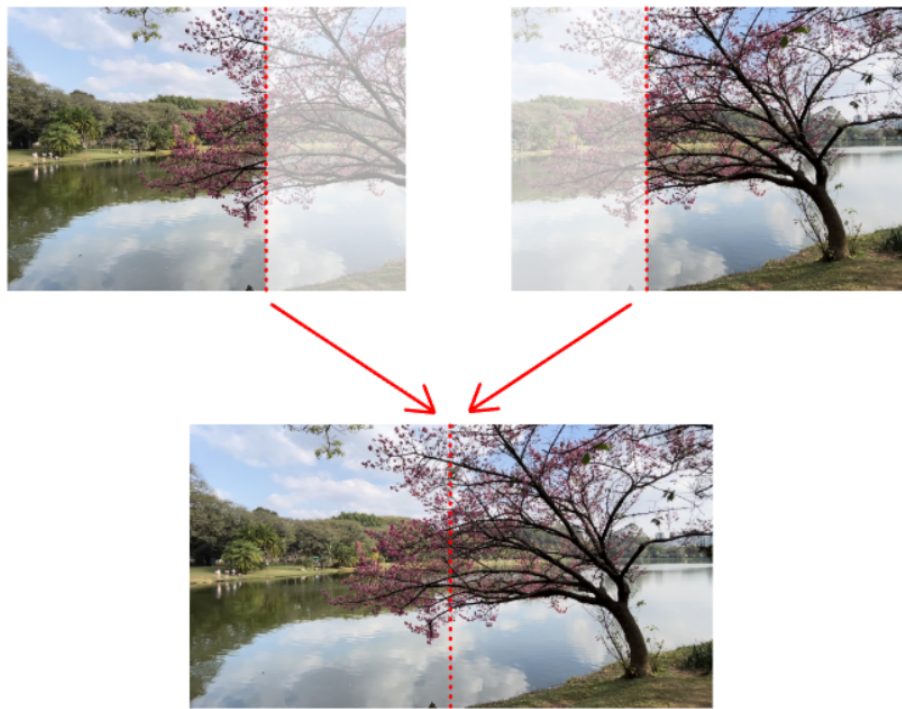


Figura 4.57 – Corte e União das Imagens - Fonte: Autor

Espera-se que esse processo seja simples e fácil, porém há alguns obstáculos no caminho, tornando necessário realizar algumas tarefas no meio do processo. O primeiro obstáculo encontrado é a variação de inclinação das imagens. Como o dispositivo é físico, ele apresenta variabilidades, apresenta folgas inevitáveis, falhas nos LEDs, e várias outros detalhes que aparecem com o tempo inevitavelmente e isso prejudicaria completamente a análise das imagens. Para evitar tal desfecho, o software foi pensado para ser robusto a tais variações.

Os métodos escolhidos para testar, como mencionado, são robustos às principais variações, detectando pontos de interesse da mesma forma, porém a união das imagens não apresenta tal robustez. Por isso, é necessário corrigir quaisquer inclinações que as imagens apresentem para que, na união, não apresentem erros. A correção dessas inclinações será discutida na seção seguinte (4.2.2.2).

Outro obstáculo observado entre as imagens foi erro de seleção de *keypoints*. Mesmo com a boa precisão do método utilizado, é inevitável que algum ponto de interesse seja detectado fora da posição correta. Esse problema foi contornado com um algoritmo de

verificação da variação de posição. Como a inclinação que o projétil pode sofrer é pequena, a variação no *pixel* correspondente com relação à linha em que ele se encontra (variação no eixo y) não será expressiva. A partir de testes experimentais, determinou-se um limiar de variação de mais ou menos 5 *pixels* no eixo y . Caso uma variação maior fosse detectada, o *keypoint* era eliminado da lista.

O último problema expressivo enfrentado foram erros ao final do processo. Como foi explicado anteriormente, não é possível cobrir exatamente 360° de rotação do projétil com o ângulo escolhido para cada rotação, ou seja, faltará parte da face ou repetirá um pedaço ao fim. Escolheu-se anteriormente repetir o final e consertar isso digitalmente. Por esse motivo, ao final do processo, quando tem-se uma imagem quase completa da face lateral do projétil e o programa ainda fará comparação com as imagens finais da lista, em algum momento uma das imagens terá presente ao mesmo tempo partes do início e do fim da imagem quase completa, o que gera confusão ao algoritmo.

Para evitar tal problema, foi adotada uma espécie de *kernel* para as análises de semelhança entre as imagens. Para cada comparação, a lista de pontos selecionados só poderia conter pontos que estivessem dentro de uma região de tamanho fixo sempre ao lado direito da imagem. Essa região fixa apresenta as mesmas dimensões das fotos capturadas após o corte das distorções, ou seja, a área efetiva de comparação será sempre a mesma e em nenhum momento haverá conflito com o início da imagem. A figura 4.58 mostra a área efetiva comparada com a altura original da imagem e com a largura L das imagens cortadas inicialmente.



Figura 4.58 – Área de Comparação Efetiva - Fonte: Autor

4.2.2.2 Correção de Inclinação

As folgas e quaisquer avarias que venham a ocorrer com o dispositivo fazem com que as imagens apresentem movimentações do projétil em outras direções além da direção de

rotação do projétil. Esse efeito prejudica a união simplificada das imagens que foi descrita anteriormente.

Para resolver esse problema, utilizou-se a teoria da homografia, discutida na seção 3.3.1.5. Lá é mostrado como achar a relação entre dois planos que apresentam posições diferentes, mas pontos equivalentes. Com a mesma lógica vista lá, será determinada a relação de homografia entre as imagens consecutivas capturadas.

Como visto na seção 3.3.1.5, a quantidade ideal de pares de pontos do plano a se conhecer é quatro, quantidade essa suficiente para determinar a matriz de homografia, responsável por identificar a relação de distorção de uma imagem com a outra.

Com base nisso, foi feito um algoritmo que, a partir dos *keypoints* detectados anteriormente, varre a lista e seleciona os pontos equivalentes aos vértices do maior quadrilátero possível de se desenhar com os *keypoints* encontrados. Ou seja, é encontrado o ponto mais a esquerda e acima, o ponto mais a direita e acima, o ponto mais a esquerda e abaixo e o ponto mais a direita e abaixo possível. Um exemplo é mostrado na figura 4.59.



Figura 4.59 – Quatro Pontos Escolhidos Para a Homografia - Fonte: Autor

A partir destes pontos, é possível determinar a matriz de homografia que relaciona as imagens. Considerando a imagem a esquerda, que sempre será a última imagem resultante de uma união, como referência, a imagem nova sempre será a imagem considerada distorcida e que precisa ser reconfigurada. Utilizando a matriz de homografia, é possível encontrar a posição em que cada ponto da imagem distorcida deveria estar para assumir a mesma posição da imagem de referência e assim gerar uma nova imagem corrigida.

Com nova imagem gerada, aplica-se o detector de *keypoints* novamente para garantir que as posições detectadas estão de acordo com o esperado e o processo segue como mencionado anteriormente para evitar os problemas mencionados. Essa sequência de detecção, homografia, nova detecção, corte e união das imagens é repetida para todas as 158 imagens até que se obtenha a imagem completa da face lateral do projétil.

Como mencionado, não é possível fotografar perfeitamente os 360° de rotação do projétil e ao final da imagem gerada haverá repetição do início. Esse problema foi corrigido ao cortar a imagem final para ter o tamanho equivalente aos 360°, fazendo ainda com que seja garantido que todas as imagens geradas de projéteis tenham as mesmas dimensões.

4.2.3 Extração de Sinais para Comparação

Para conseguir comparar as imagens de dois projéteis, é necessário extrair informações específicas que identifiquem individualmente cada projétil. Para isso, decidiu-se utilizar sinais para realizar as comparações.

Existem diversas formas para se extrair sinais de imagens, cada uma com recomendações conforme pretende-se utilizar o sinal. É necessário avaliar as necessidades do projeto para se escolher a forma como extrair sinais das imagens.

No atual projeto, as imagens são padronizadas e apresentam picos de luminosidade de acordo com o relevo da superfície dos projéteis, ou seja, é possível utilizar tais características como informações específicas e individuais visto projéteis que são disparados de armas diferentes apresentam relevos diferentes na superfície.

Sendo assim, definiu-se extrair o sinal das intensidades dos *pixels* que representam o quão alto ou baixo é o relevo referente àquele *pixel* específico. Para isso, a imagem deve estar em escala de cinzas para ter apenas um valor de intensidade por *pixel*. Dependendo do descritor utilizado para encontrar os *keypoints*, a imagem da face lateral pode ser criada já em escala de cinzas, o que pode facilitar no passo de extrair sinal, porém a conversão para escala de cinzas é simples e rápida, pode-se fazê-la no processo de extrair o sinal.

O sinal, em uma imagem de projétil, pode ser extraído por meio de uma linha de *pixels*, sendo fiel às ranhuras em função da intensidade de iluminação que varia de acordo com o relevo. Porém, essa linha pode trazer erros de interpretação devido a ruídos que eventualmente aparecem causado por ferrugem, lascas de superfícies que tiveram contato, quaisquer sujeiras, entre outros. Logo, é necessário utilizar mais linhas para conseguir extrair um sinal adequado para as comparações.

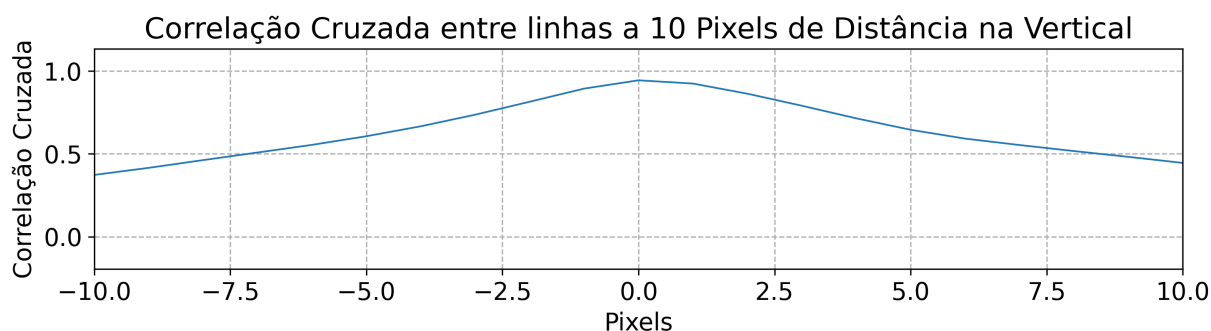
Um projétil apresenta ranhuras inclinadas em sua superfície, ou seja, o sinal baseado na intensidade da iluminação acaba variando de acordo com a extensão vertical do projétil. Pode-se notar que o ângulo de inclinação das ranhuras dos projéteis não é alto e isso se deve ao passo longo que as ranhuras apresentam no cano da arma. O ângulo de um projétil de exemplo foi calculado como pode-se ver na figura 4.60, e resultou em $\alpha = 6,34^\circ$.



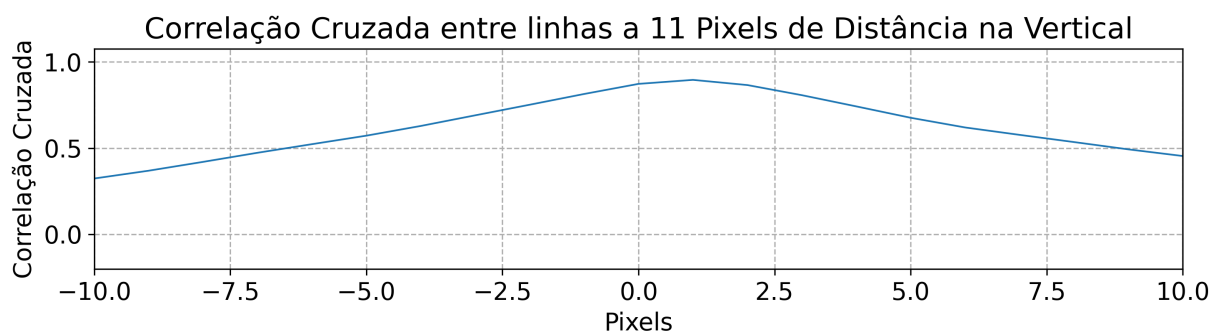
Figura 4.60 – Ângulo de Inclinação da Ranhura de Um Projétil - Fonte: Autor

Por meio de medidas físicas do projétil e da imagem capturada, calculou-se a relação entre *pixel* e comprimento real e assim descobriu-se que cada *pixel* da imagem representa 0,0167mm do projétil. Sabendo-se do ângulo de inclinação das ranhuras, calculou-se que a cada 0,1503mm percorridos verticalmente na ranhura, a ranhura desloca para o lado 0,0167mm. Ou seja, para cada nove *pixels* deslocados na vertical, um *pixel* é deslocado na horizontal. Não é necessário calcular o deslocamento em milímetros para descobrir o deslocamento em *pixels*, mas é interessante visualizar o tamanho do deslocamento efetivo no mundo real.

Como forma de confirmar a conta acima, dado que o ângulo calculado partiu de medidas manuais, foi realizada a correlação cruzada entre linhas consecutivas e observou-se que após a 10ª linhas houve um deslocamento do sinal em 1 *pixel*, ou seja, confirmou-se que a cada 10 *pixels* na vertical, a ranhura caminha 1 *pixel* na horizontal. Esse resultado pode ser visto na figura 4.61.



(a) Correlação Cruzada no 10º Pixel



(b) Correlação Cruzada no 11º Pixel

Figura 4.61 – Deslocamento Horizontal de 1 *pixel* Após o Deslocamento Vertical de 10 *Pixels* - Fonte Autor

Sendo assim, um trecho de dez linhas de *pixels* mantém as informações, começando a alterar levemente após esse espaço devido a limitação do *pixel* em representar apenas uma intensidade de iluminação. Dessa forma, com o intuito de conseguir um sinal adequado para a comparação decidiu-se utilizar uma faixa de dez linhas da imagem para gerar um sinal mais refinado. Como em geral há a repetição da informação desejada e a informação não desejada é esporádica, escolheu-se a média como filtro para eliminar o ruído presente nas linhas, visto que a média levará o valor final do sinal mais próximo aos valores que mais se repetem.

Nesse ponto, tem-se a abordagem definida para extrair o sinal de um projétil. Para melhorar os resultados, é possível retirar mais de um sinal da mesma imagem, em posições variadas, sendo possível fazer análises com esses sinais e comparar os resultados antes de tomar uma decisão.

4.2.4 Comparação de Sinais

A comparação de sinais é o processo que irá avaliar a semelhança ou a diferença entre diferentes sinais extraídos das imagens. Na seção 3.3.3 foram discutidos dois métodos para comparações de sinais e pretende-se testá-los e validá-los.

4.2.4.1 Autocorrelação e Correlação Cruzada

O primeiro método de comparação de sinais escolhido para testar foi a utilização de Autocorrelação e Correlação Cruzada. A aplicação deste método é baseada na aplicação realizada por João Bosco ([SILVINO JUNIOR, 2010](#)) na tentativa de comparar projéteis de arma de fogo.

Como comentado na seção [3.3.3.1](#), a autocorrelação mede a similaridade de um sinal com ele mesmo ao longo tempo, onde a comparação é feita em relação ao atraso do sinal em relação a ele mesmo. Já a correlação, mede a similaridade entre dois sinais diferentes, mostrando, da mesma forma, a similaridade entre eles de acordo com o tempo podendo identificar se os sinais são de mesma origem ou não.

Utilizando essa ideia, deseja-se comparar os sinais obtidos pelas imagens utilizando esse método para poder avaliar se os sinais pertencem a mesma origem o não, ou seja, se são projéteis expelidos pela mesma arma ou não.

Como primeiro passo, é interessante que seja calculada a autocorrelação de um sinal antes de serem comparados. Deve-se utilizar um mesmo sinal que foi extraído para que seja possível saber como o sinal se relaciona com ele mesmo antes de julgar a comparação com outros sinais.

Em seguida pode-se calcular a correlação cruzada entre dois sinais de projéteis diferentes. Esse cálculo traz um gráfico que mostra em quais posições os dois sinais se assemelharam mais e menos. Dessa forma, pode-se gerar um laudo sobre a possível compatibilidade entre eles, sabendo ainda o atraso de um sinal em relação ao outro.

A partir de testes de comparação entre projéteis de mesma arma e de projéteis de armas diferentes, imagina-se ser possível identificar diferenças suficientes para determinar quando projéteis foram expelidos pela mesma arma ou por armas diferentes.

Para realizar esse procedimento e obter um resultado mais expressivo, com mais confiança, é necessário realizar várias comparações para que não haja viés. Aqui é criado um obstáculo para a continuação do projeto, foi conseguido apenas dois projéteis para análise, os quais são de diferentes armas, ou seja, se torna impossível realizar o procedimento descrito com as presentes amostras. Por este motivo, decidiu-se trabalhar com amostras adquiridas por [Silvino Junior \(2010\)](#) em seu trabalho, as quais são amostras de dois projéteis por arma, sendo sete armas ao todo, ou seja, quatorze amostras. Estas amostras estão disponíveis no capítulo [A](#) (Anexo A) deste relatório.

4.2.4.2 Comparação Estatística

A comparação estatística nada mais é do que uma análise da semelhança entre as medidas estatísticas de cada sinal. As principais medidas estatísticas foram citadas na seção [3.3.3.2](#).

O método desenvolvido aqui começa pelo cálculo das medidas estatísticas para o par de sinais que serão comparados. Em seguida essas medidas são colocadas em um vetor em ordem específica e a partir deste ponto é possível comparar os sinais. Essa comparação é feita por meio de medidas de similaridade e distância citadas na seção 3.3.2.

Para descobrir qual a medida de similaridade mais compatível com esse método, testou-se três das quatro estudadas junto a esse método. Escolheu-se a distância euclidiana, a distância de Manhattan e a distância de Minkowski, visto que a distância de Hamming trabalha com números binários, o que não é o caso do método aplicado aqui.

Para realizar a comparação, será plotado um gráfico com os resultados de todas as distâncias calculadas. Esse gráfico será separado em duas curvas, a curva de distâncias entre projéteis de mesma arma e a curva de distâncias entre projéteis de armas diferentes. Para plotar estes gráficos, será feita uma distribuição de densidade de probabilidade, ou seja, o gráfico levará em conta a quantidade de vezes que uma certa distância se repete, mostrando o valor da distância em no eixo X e sua densidade no eixo Y. A figura 4.62 mostra um exemplo de distribuição de densidade de probabilidade em quatro curvas diferentes. O método foi escolhido pois imagina-se ser possível diferenciar a relação entre os projéteis baseado na diferença entre as duas curvas geradas.

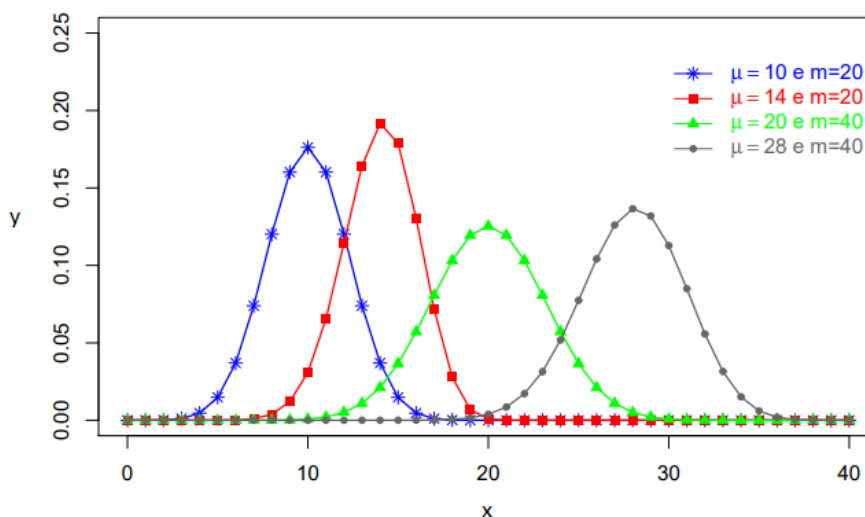


Figura 4.62 – Distribuição Em Função de Densidade de Probabilidade

5 Resultados

5.1 Captura das Imagens

5.1.1 Câmera - *WebCam*

Como citado anteriormente, decidiu-se primeiramente investir na montagem de um microscópio por meio da reversão das lentes de uma *WebCam*. Foi feito o teste da reversão das lentes e alguns resultados foram obtidos.

Primeiramente houve a impressão de falha no propósito, porém, com persistência percebeu-se que o microscópio criado é de extrema fina calibração, necessitando de algum mecanismo de ajuste fino para que o sistema funcione como o esperado.

Como forma de testar o dispositivo montado, tentou-se ajustar com as mãos o foco do microscópio e algumas imagens foram capturadas. Neste momento não havia acesso a projéteis ainda, impossibilitando testes mais precisos. Utilizou-se, então, um projétil impresso em 3D para o primeiro teste. A figura 5.63 mostra o projétil e a imagem capturada dele.

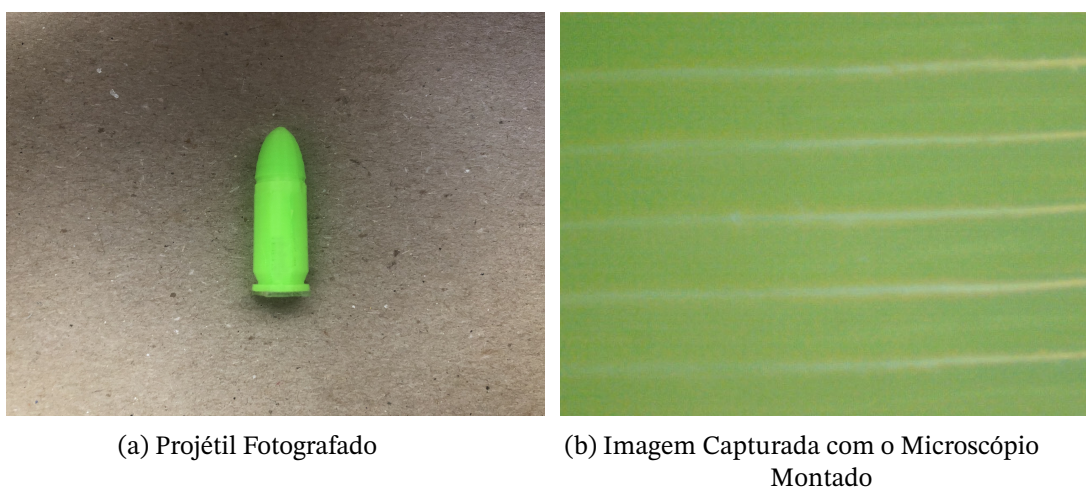


Figura 5.63 – Projétil e sua imagem capturada pelo microscópio montado - Fonte: Autor

Pela imagem é possível perceber que a imagem não de muito boa qualidade, obteve muito brilho, a luz refletiu, ficou um tanto difusa. Cogitou-se diminuir e/ou retirar completamente a fonte de luz diretamente projetada no objeto, porém com um pouco menos de luz a câmera escureceu, não sendo possível visualizar nada. A princípio concluiu-se que o microscópio não obtinha qualidade para o propósito, porém, antes de qualquer conclusão definitiva decidiu-se realizar alguns testes com outros materiais e cores diferentes.

Alguns testes foram feitos e separou-se os melhores resultados para serem evidenciados. O primeiro teste a ser evidenciado envolveu uma peça impressa em filamento preto. Por

ser compacto e de cor escura, totalmente diferente da réplica do projétil, o resultado teve grande melhora e é visível na figura 5.64.

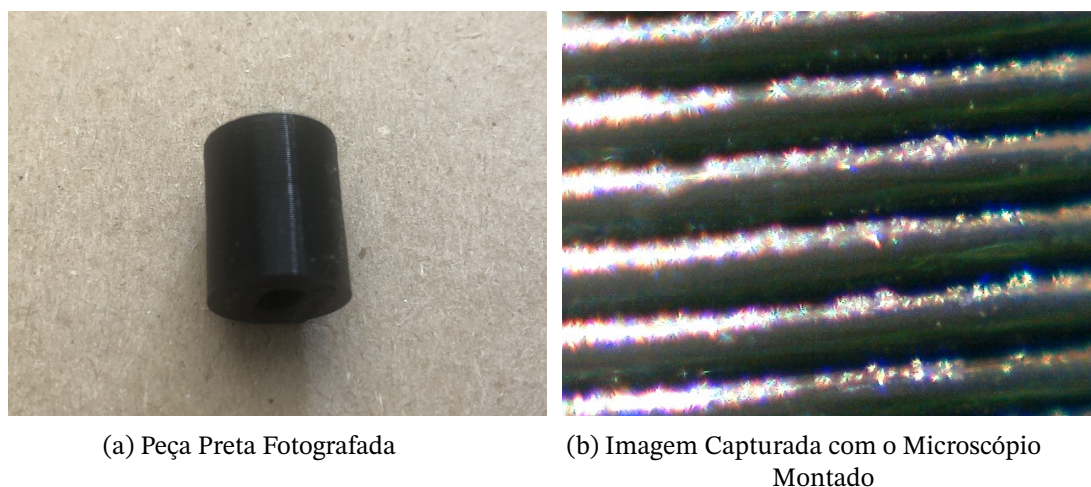


Figura 5.64 – Peça Preta e sua imagem capturada pelo microscópio montado - Fonte: Autor

Com esse teste foi possível ver claramente as camadas do filamento de impressão, sendo que as bordas da imagem ficaram ligeiramente mais desfocadas, o que não é um empecilho para o projeto visto que as imagens serão tratadas e recortadas.

Com esse teste observou-se que o problema anterior era de fato a quantidade de luz que o objeto refletia ofuscando a captura da imagem pela câmera. Mesmo que ainda refletindo luz, o plástico preto absorveu muito mais luz, o que tornou possível a captura da imagem de forma adequada.

Neste ponto, o funcionamento do microscópio havia sido comprovado, mas havia a dúvida sobre quais materiais e superfícies o microscópio funcionaria e em quais sua utilidade seria nula. Com o intuito de chegar a alguma conclusão sobre o funcionamento do microscópio, realizaram-se testes com o microscópio utilizando objetos com materiais e características físicas parecidas com um projétil. Utilizou-se uma lima (uma espécie de lixa metálica) que apresenta um cabo aderente com ranhuras dos dois lados. Na figura 5.65 é possível ver a lima, uma ampliação digital para melhor visualização de suas ranhuras e por fim uma captura de imagem do microscópio montado,

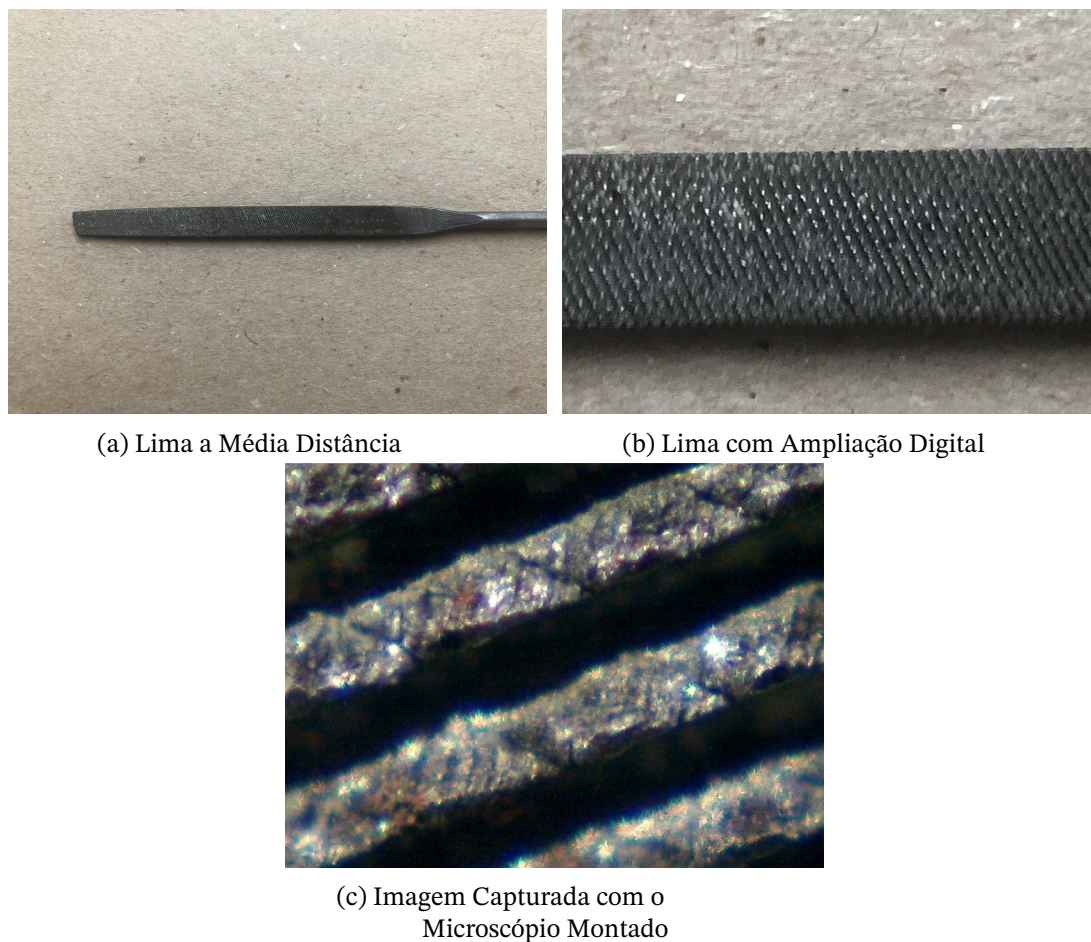


Figura 5.65 – Lima, Lima com zoom e sua imagem capturada pelo microscópio montado - Fonte: Autor

Com o novo teste e a clara visibilidade das ranhuras largas e finas da lima, constatou-se que as imagens obtidos pelo microscópio apresentavam grande chance de satisfazer o objetivo principal.

Após conseguir o acesso a projéteis, a câmera foi testada e tentativa de fotografar um projétil foi falha devido à ampliação excessiva da lente. Algumas tentativas de remontar o microscópio com a lente em posições diferentes, a distanciando do sensor fotossensível da câmera também não surtiu efeito significativo para viabilizar seu uso no projeto. Sendo assim, foi necessário partir para uma nova solução.

5.1.2 Microscópio

A tentativa com a inversão da lente foi a primeira escolha devido ao baixo custo e fácil adaptação, porém, como os resultados inviabilizaram o uso da câmera, foi necessário subir um pouco o custo e escolher um microscópio digital de pequeno porte.

Com o microscópio em mãos, testes foram feitos para validar a utilização do mesmo. Primeiramente testou-se a ampliação que era possível obter com o microscópio e a figura

5.66 mostra esse resultado.

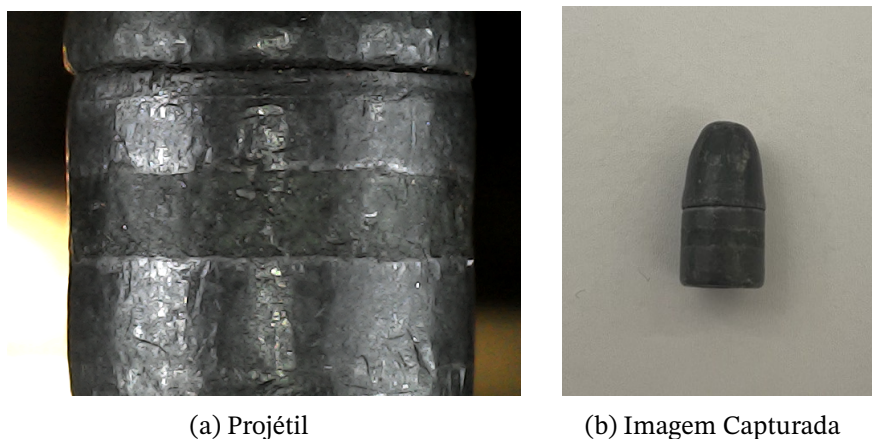


Figura 5.66 – Projétil e Primeira Imagem Capturada com o Microscópio - Fonte Autor

O primeiro projétil ao qual foi liberado o acesso era um projétil não disparado, ele teve sua cápsula retirada manualmente, logo não era esperado visualizar ranhuras nele. O interessante do primeiro teste seria identificar se era possível coletar bastante informação do projétil com o microscópio. Como é possível ver na figura 5.66 letra b, ao ajustar a ampliação e foco do microscópio foi possível visualizar com qualidade as características do projétil, a marca da cápsula, pequenas marcas na superfície e o seu relevo. Alguns ajustes foram feitos aproximando e afastando o microscópio do projétil para que fosse possível analisar como o microscópio se comportaria com outros modelos de projéteis que necessitariam de mudança de posição do microscópio para capturar uma imagem completa. As mudanças de posição foram satisfatórias, os ajustes de ampliação e foco apresentaram ótimo funcionamento.

Sendo assim, validou-se a ampliação do microscópio e pode-se seguir para os próximos testes. Pensando na estratégia de extrair sinais da imagem que é baseada na intensidade de iluminação que cada pixel da imagem apresenta, é necessário testar a intensidade dos LEDs e do difusor para que a iluminação fique difusa e não gere variações que viriam a atrapalhar o processo de comparação. Esse teste foi realizado com o projétil mostrado na figura 5.67, um projétil 9mm, e seu resultado pode ser visualizado na figura 5.68.



Figura 5.67 – Projétil Utilizado para Testes e Ajustes - Fonte: Autor

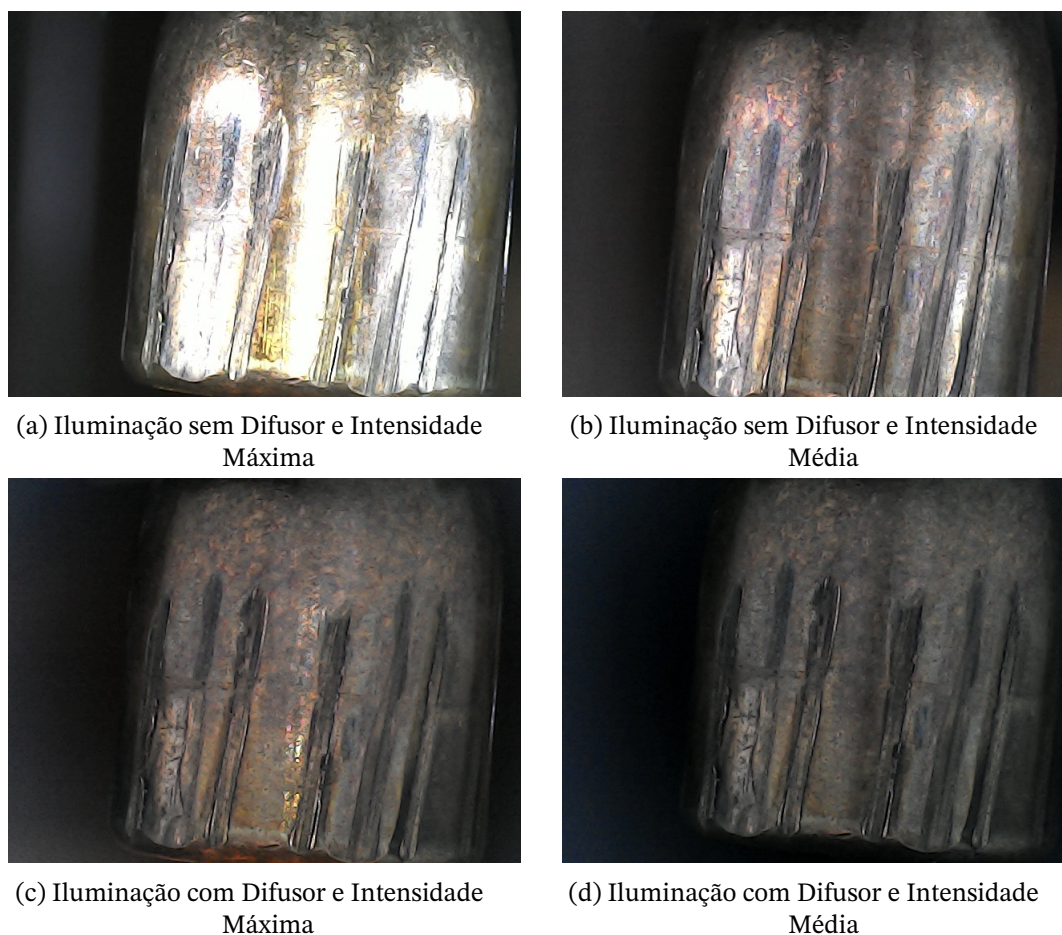


Figura 5.68 – Teste de Iluminação - Fonte Autor

A letra a da figura 5.68 mostra a imagem capturada do projétil com as luzes ligadas diretamente em frente a ele e em intensidade máxima. Percebeu-se que o projétil apresentou muito brilho ao refletir as luzes e então decidiu-se reduzir a intensidade da luz para média, o que mostra a letra b da figura 5.68. A intensidade do brilho diminuiu consideravelmente, porém ainda não era o indicado para o dispositivo, o brilho restante ainda poderia atrapalhar a comparação. Sendo assim, decidiu-se utilizar um difusor para espalhar a luz, diminuindo o brilho. A letra c da figura 5.68 mostra o resultado da aplicação de um difusor às fontes de luz. Ainda era perceptível pontos de brilho e então foram feitos vários testes alterando a intensidade dos LEDs e a quantidade de camadas de difusor, chegando ao resultado mostrado na letra d da figura 5.68.

O próximo passo é testar a montagem do dispositivo, verificar como as fotos ficam ao associar o microscópio, o projétil girando e a iluminação escolhida. Todos os ajustes foram feitos e foram capturadas todas as 158 imagens previstas para o processo. Ao analisar as imagens obtidas foi notável uma movimentação do projétil com relação a região de captura do microscópio. As letras a e b da figura 5.69 mostram essa variação onde na letra a tem-se o projétil na foto 56 e na letra b tem-se o projétil na foto 131.

A movimentação foi analisada e identificou-se um movimento de precessão da estru-

tura que segura o projétil. O movimento ocorre devido à distância entre o projétil e o eixo do motor ser relativamente grande (aproximadamente 10 cm) e a estrutura ser relativamente pesada. Considerando a distância de aproximadamente 10 cm mencionada, com uma inclinação de 5° próximo ao eixo do motor, o projétil moveria horizontalmente cerca de 8,7 mm, ou seja, o projétil utilizado (9 mm) praticamente sairia da região de captura do microscópio. Sendo assim, qualquer mínima inclinação que ocorra na estrutura próxima ao eixo gera uma grande variação na posição do projétil.

Para solucionar esse problema seria ideal modelar e construir uma estrutura mais robusta para manter o conjunto de peças o mais centralizado possível, evitando grandes variações. Dado o tempo para finalizar o projeto e todo o trabalho computacional a ser desenvolvido, não foi possível mexer na estrutura, então decidiu-se remontá-la ajustando delicadamente cada peça e cada encaixe de forma a evitar a precessão e o resultado é mostrado na figura 5.69 nas letras c e d.

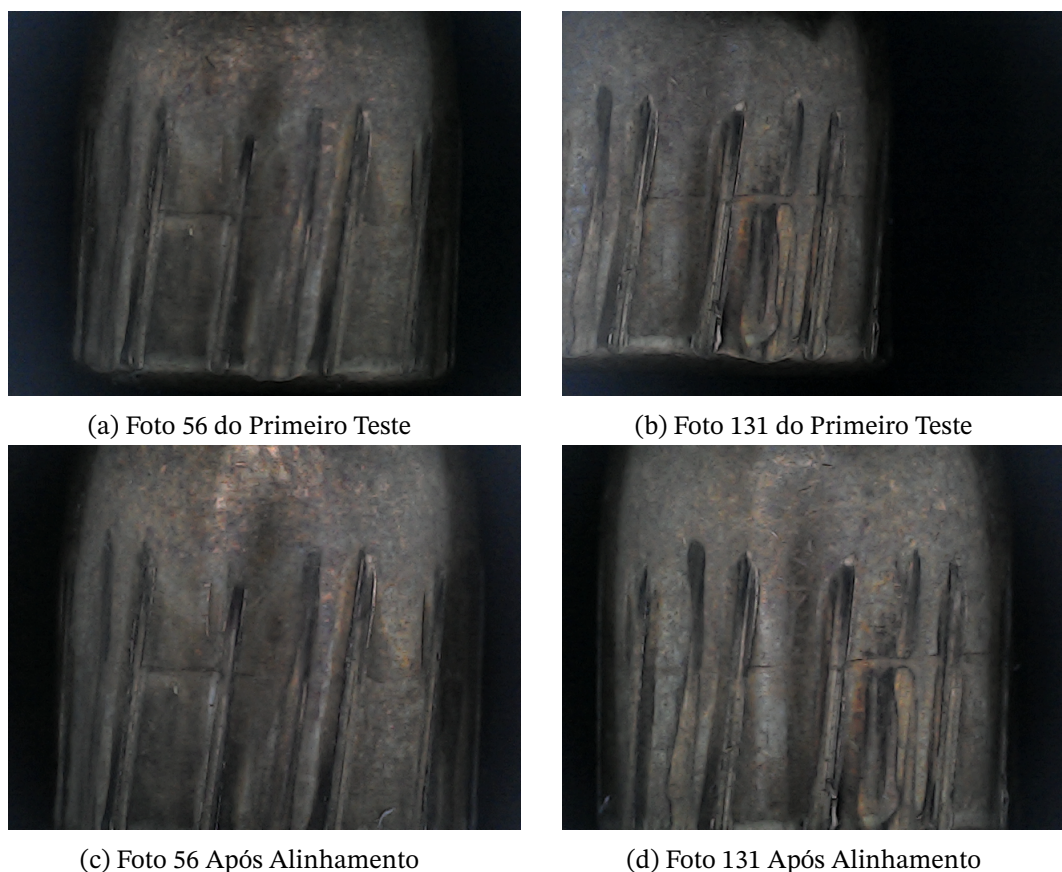


Figura 5.69 – Teste do Dispositivo Completo - Fonte Autor

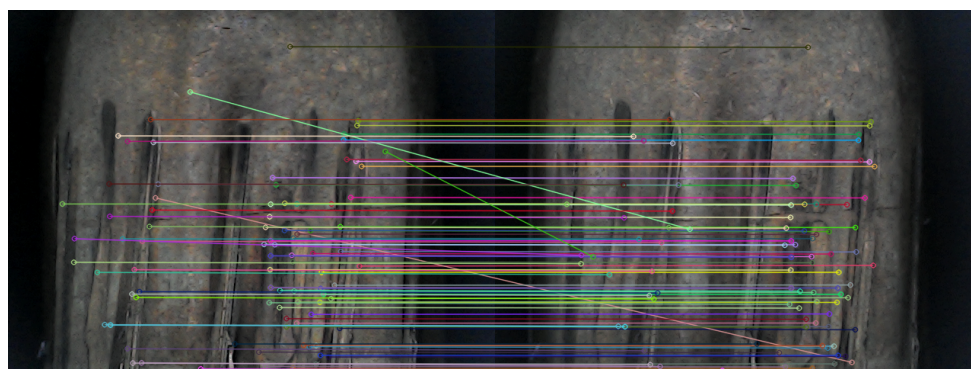
A letra c da figura 5.69 mostra a foto 56, equivalente a letra a, e a letra d mostra a foto 131, equivalente a letra b. É possível notar que houve uma leve variação da posição do projétil entre a letra c e d, porém mínima comparada com a variação entre as letras a e b. Com essa solução temporária foi possível prosseguir com o projeto e iniciar o processamento digital das imagens.

5.2 Processamento de Imagens

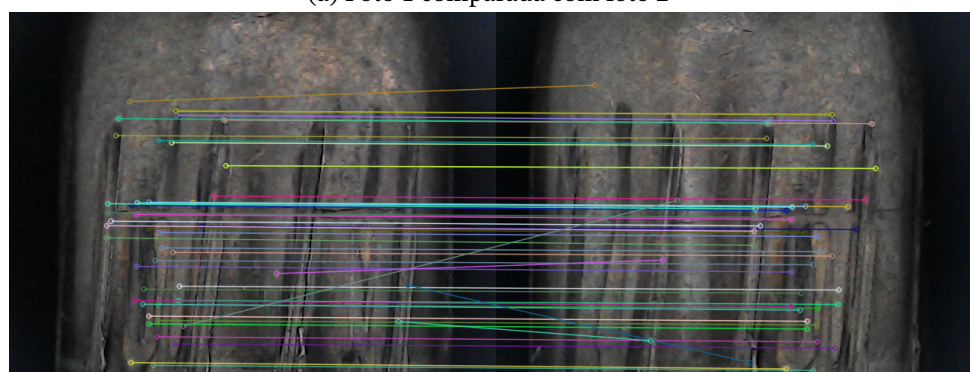
Com o sistema funcionando e com as 158 imagens fotografadas, pode-se iniciar o processamento das imagens.

5.2.1 União das Imagens

Para realizar a união das imagens foi, primeiramente, necessário definir qual dos métodos sugeridos anteriormente (seção 3.3.1) será utilizado para seguir o projeto. Para isso, foram testados os quatro métodos com dois pares de imagens. O primeiro par é composto por duas imagens consecutivas, que representa o processo que será realizado, comparando e unindo sempre imagens consecutivas. O segundo par, por outro lado, é composto por duas imagens com 19 fotos de diferença, o que representa uma rotação de aproximadamente 45° do projétil. Esse teste foi realizado com o objetivo de validar a invariabilidade a iluminação, a rotação e escala, visto que com tamanha distância entre as fotos os pontos em comum entre as duas fotos mudaram completamente de posição, sendo afetados pela iluminação não perfeita, pela pequena precessão do mandril e pela variação de escala devido à curvatura do projétil. As figuras 5.70, 5.71, 5.72 e 5.73 mostram os testes mencionados.



(a) Foto 1 comparada com foto 2

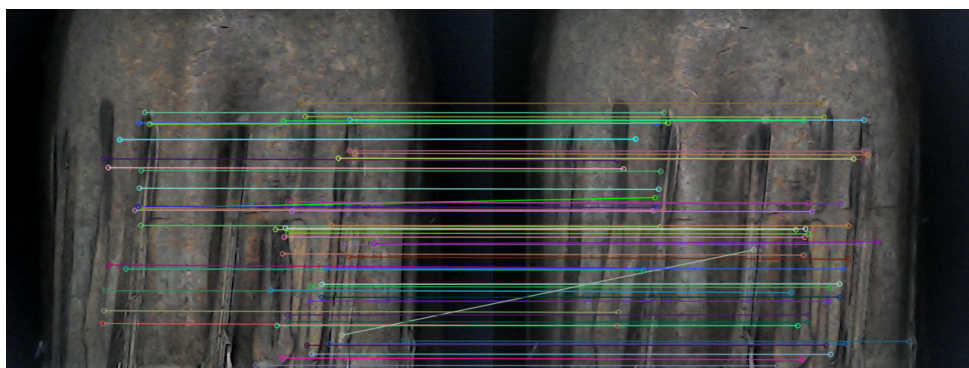


(b) Foto 1 comparada com foto 20

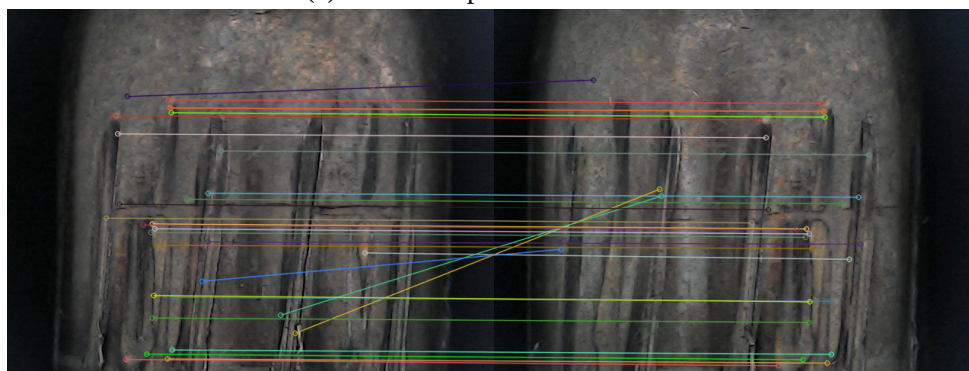
Figura 5.70 – Teste do Método SIFT - Fonte Autor

Como é possível visualizar na figura 5.70, muitos pontos em comum entre as duas imagens foram encontrados para os dois pares, ou seja, o método SIFT é bem indicado para

o escopo deste projeto. São vistos alguns erros, porém são poucos comparados à quantidade de acertos e podem ser excluídos da lista de pontos por meio do software.



(a) Foto 1 comparada com foto 2

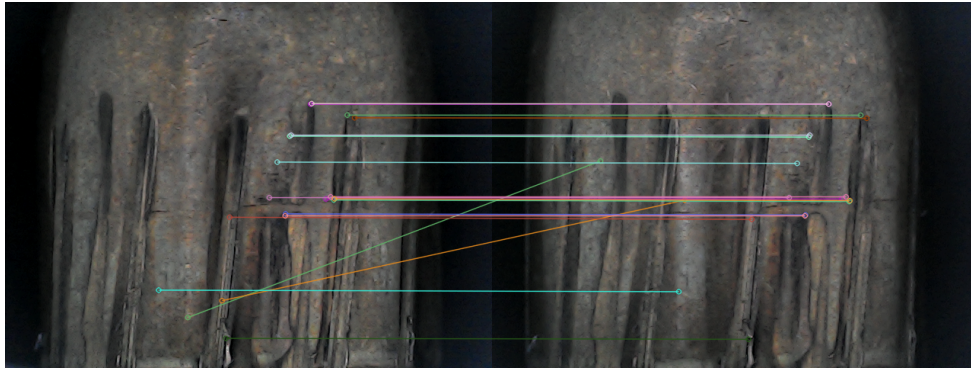


(b) Foto 1 comparada com foto 20

Figura 5.71 – Teste do Método SURF - Fonte Autor

A figura 5.71 mostra que a qualidade do SURF é semelhante a qualidade do SIFT, porém ao analisar o segundo par onde há mais distorções, o SURF apresentou menos combinações, o que pode vir a provocar erros eventualmente. Ainda assim, o SURF é indicado para o atual projeto e pode eventualmente apresentar melhores resultados comparado ao SIFT dada a proximidade dos resultados.

Os resultados do método ORB visualizados na figura 5.72 mostram uma quantidade reduzida de combinações comparado com os dois métodos anteriores e uma porcentagem alta de erros. Esses dois fatores tornam o método pouco indicado para o presente projeto, principalmente quando se tratar de situações onde as características das imagens são semelhantes ao segundo par de imagens.

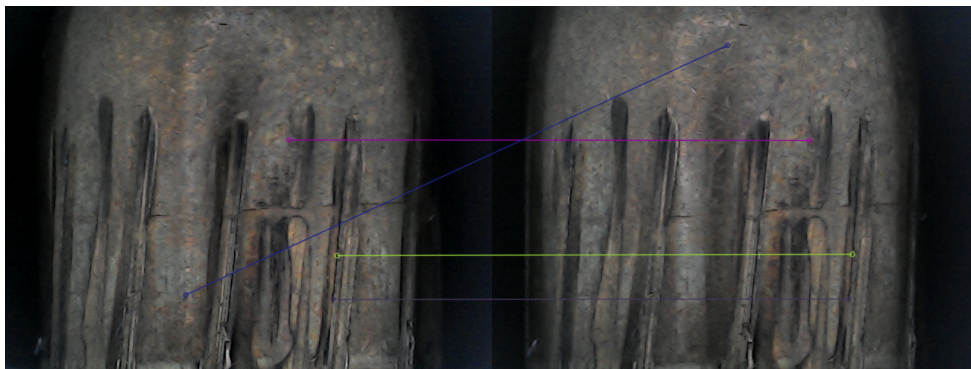


(a) Foto 1 comparada com foto 2

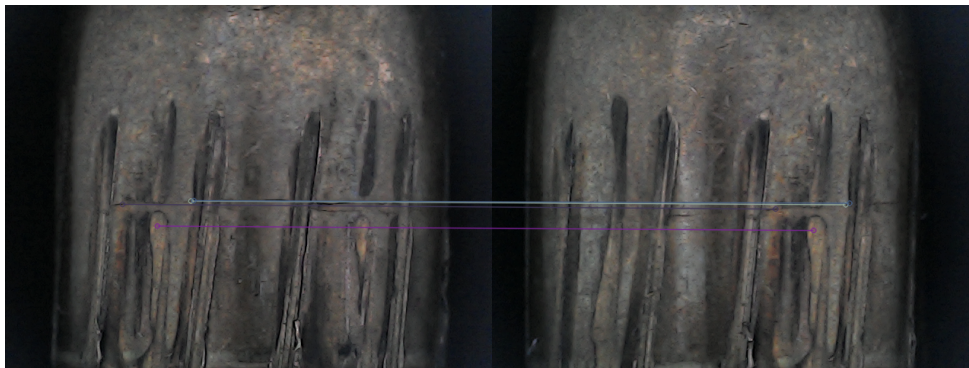


(b) Foto 1 comparada com foto 20

Figura 5.72 – Teste do Método ORB - Fonte Autor



(a) Foto 1 comparada com foto 2



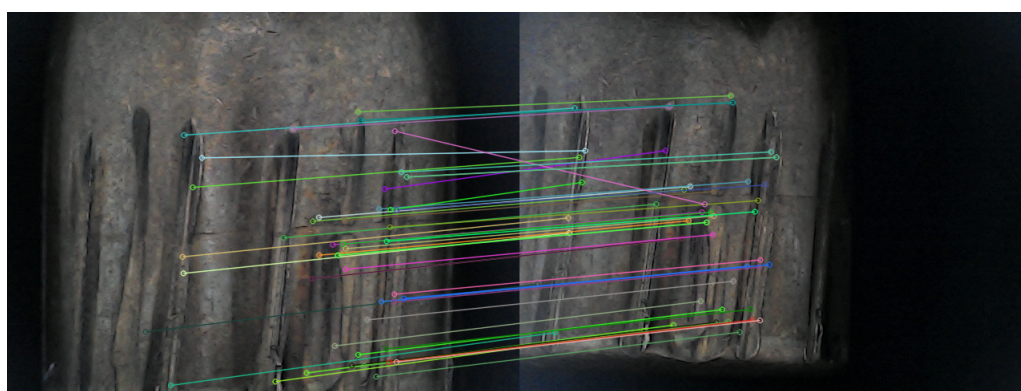
(b) Foto 1 comparada com foto 20

Figura 5.73 – Teste do Método BRISK - Fonte Autor

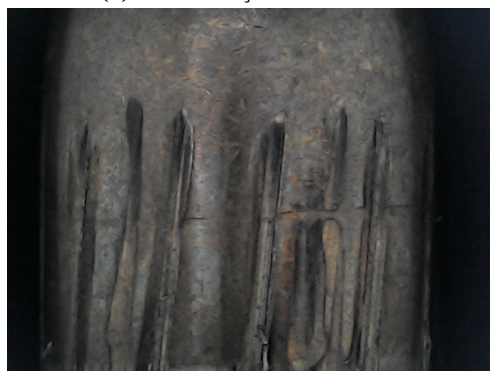
O último método testado, BRISK, tem seus resultados demonstrados na figura 5.73, onde é possível ver que praticamente não há combinações, principalmente no segundo par de imagens. Dessa forma, este método não é indicado para o projeto atual.

Após todos os testes realizados pode-se comparar os métodos e tomar uma decisão quanto ao melhor método para o presente projeto. Foi, então, escolhido trabalhar com o SIFT, visto que ele apresentou mais combinações para os dois pares de imagens e uma porcentagem baixa de erros.

O próximo passo para a união das imagens é a realização da homografia, garantindo o correto alinhamento das imagens. Utilizando a lista de pontos encontrados com o SIFT e escolhendo os quatro pontos de canto, foi possível realizar a homografia. Um exemplo pode ser visto na figura 5.74.



(a) Combinações em Fotos de Mesma Posição com Inclinações Diferentes



(b) Foto de Referência para a Homografia



(c) Foto Alinhada à Foto de Referência

Figura 5.74 – Exemplo da Homografia Realizada - Fonte Autor

Na figura 5.74 é possível ver na letra b o projétil de referência para a realização da homografia, assim como as combinações entre a imagem desalinhada e a imagem de referência na letra a. O resultado da homografia é visto na letra c da mesma figura. É possível ver que tanto a escala como a inclinação do projétil da imagem desalinhada foram corrigidas e agora é possível unir as imagens.

Com os últimos passos dados, é possível iniciar a união de imagens para se ter a face lateral do projétil planificada. Um exemplo desse processo está visível na figura 5.75.



(a) Montagem Completa



(b) Montagem com o Corte Final

Figura 5.75 – Exemplo de uma Montagem da Face Lateral Planificada de um Projétil - Fonte Autor

A letra a da figura 5.75 mostra o resultado final de uma união consecutiva das 158 fotos capturadas. É possível reparar a repetição de parte do início da imagem no seu fim e isso é devido ao ângulo escolhido de $2,39^\circ$ para rotação do motor. Com esse ângulo é impossível chegar exatamente aos 360° , sendo necessário ultrapassar esta marca. Para corrigir esse problema, foi feito um corte na imagem baseado no excesso percorrido pelo motor que é conhecido. O Corte pode ser visto na letra b da figura 5.75.

Com a face lateral completa planificada é finalizado o processamento de imagens e pode-se partir para o processamento de sinais.

5.3 Processamento de Sinais

O processamento de sinais inicia com a extração do sinal da imagem montada anteriormente. Como mencionado na seção 4.2.3, o sinal será obtido a partir da média de 10 linhas consecutivas. A posição pode depender de projétil para projétil, mas no caso do projétil de 9mm utilizado anteriormente, escolheu-se pegar o sinal das linhas 90 a 99 já que elas apresentam as ranhuras sem interferência das marcas deixadas pela cápsula, marcas estas que serão diferentes de projétil para projétil sem relação com a arma utilizada, atrapalhando a comparação. Uma comparação entre o sinal obtido apenas da linha 90 e a média entre as linhas 90 a 99 pode ser vista na figura 5.76. É possível reparar que a média realmente funcionou como filtro, há vários pontos que foram suprimidos e refinados, gerando um sinal mais fiel à estrutura das ranhuras no projétil.

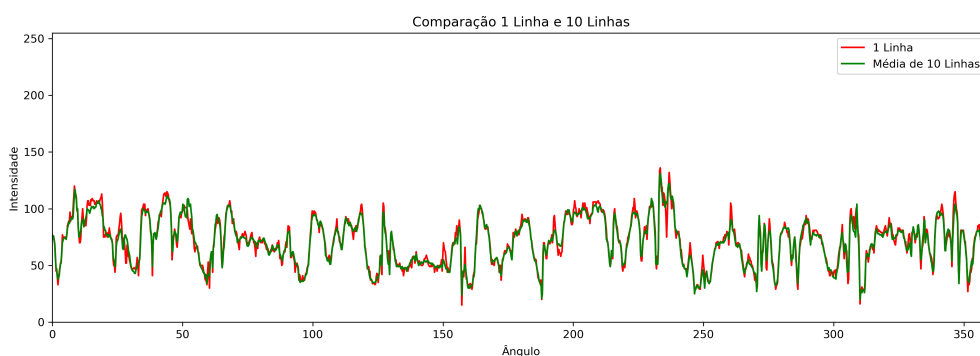


Figura 5.76 – Comparação Sinal da Linha 90 e Sinal da Média entre as Linhas de 90 a 99 - Fonte: Autor

A partir deste ponto, o próximo passo é extrair o sinal de todos os projéteis que serão testados e assim iniciar as comparações utilizando estes sinais e os dois métodos descritos anteriormente nas seções 3.3.3.1 e 3.3.3.2. Como foram conseguidos apenas dois projéteis para teste, não é possível testar com efetividade o sistema com tão pouca quantidade. Como forma de resolver tal problema, serão utilizados as imagens de face lateral de 14 projéteis, sendo que cada dois são de mesma arma, que foram acessados em um trabalho pesquisado (SILVINO JUNIOR, 2010).

5.3.1 Autocorrelação e Correlação Cruzada

Foi realizada uma autocorrelação com o sinal obtido anteriormente com o objetivo de testar a teoria estudada, que mostra que o resultado deve mostrar um pico no atraso igual a zero graus. O resultado dessa correlação é visto na figura 5.77.

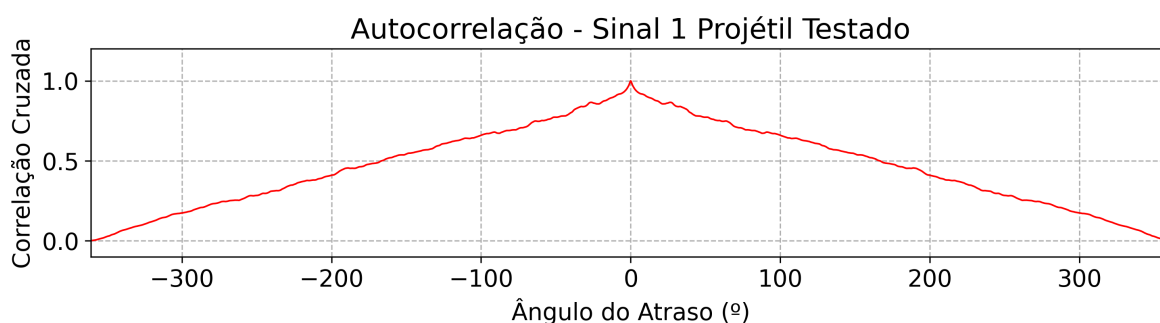
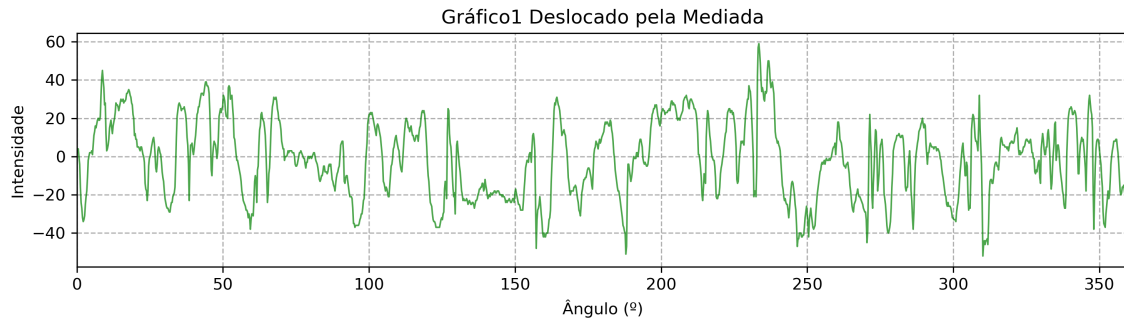


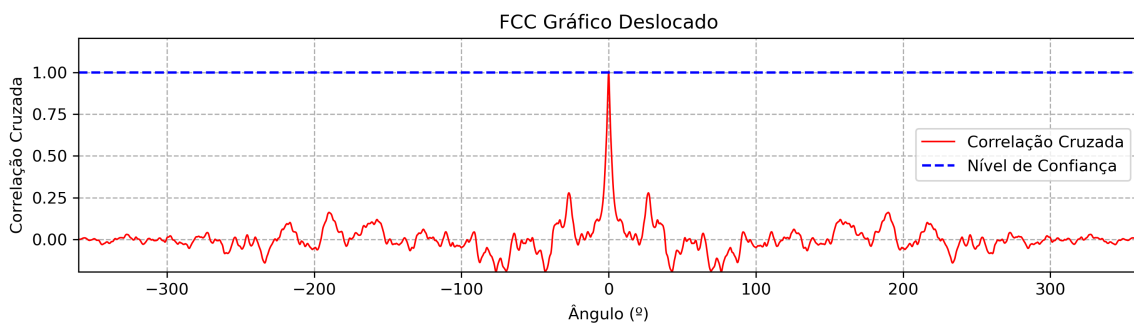
Figura 5.77 – Autocorrelação do Sinal Obtido a partir do Projétil 9mm - Fonte: Autor

O sinal observado não foi como esperado, apresentou um formato de rampa dupla ao invés de apresentar um formato de pulso. Ao investigar e estudar o sinal, ficou claro que o problema está na completa positividade do sinal, ou seja, para nenhum atraso o valor da autocorrelação será zero e quanto mais próximo do atraso zero, maior a autocorrelação. Para resolver esse problema, tomou-se a decisão de transladar o sinal verticalmente para que parte

dele apresentasse valores negativos e assim a autocorrelação mostrasse o resultado esperado. Essa ideia foi cogitada, pois o formato do sinal seria mantido, assim como as variações e diferenças entre os pontos. Para realizar a translação do sinal foi escolhido como variação a mediana do sinal, dessa forma teria-se a mesma quantidade de sinais negativos e positivos no sinal. O resultado dessa translação pode ser visto na figura 5.78.



(a) Sinal Transladado Pela Mediana



(b) Autocorrelação do Sinal Transladado

Figura 5.78 – Resultado da Translação Vertical do Sinal - Fonte Autor

É possível reparar, na letra b da figura 5.78, que há um pico no atraso igual a zero e, mesmo não sendo um pulso, é um pico discrepante com relação ao resto do gráfico, estando o resto próximo a linha do zero. Esse gráfico da autocorrelação mostra que o processo de comparar os sinais é promissor, porém é necessário comparar os demais projéteis. Para se ter mais certeza do resultado, foi plotado um nível de confiança baseado no nível de 99,7% da tabela *T-student*.

O procedimento realizado anteriormente para translação do sinal verticalmente pode ser aplicado para todos os sinais e então torna-se possível realizar todas as comparações necessárias para validar o método.

5.3.1.1 Comparações Entre Amostras de Mesma Arma

Seguindo o descrito, as amostras de mesma arma foram comparadas e os resultados são mostrados a seguir. Mesmo sabendo que são projéteis da mesma arma, será analisada a compatibilidade dos projéteis baseado no gráfico gerado.

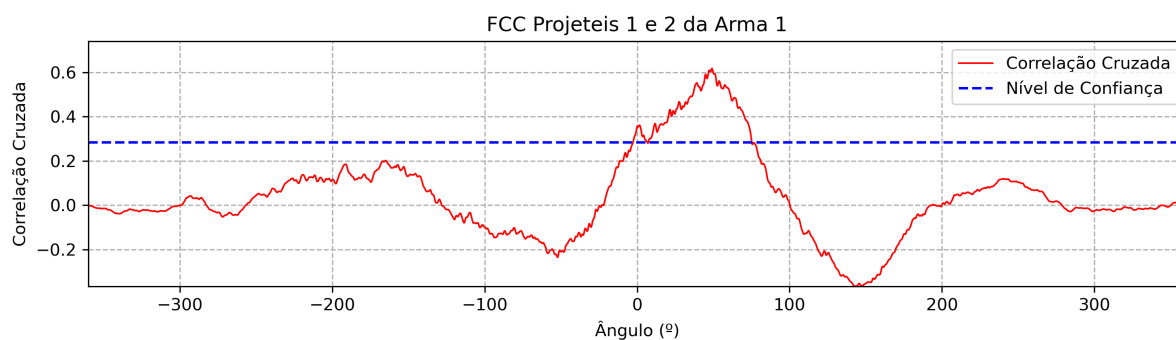


Figura 5.79 – Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 2 da Arma 1 - Fonte: Autor

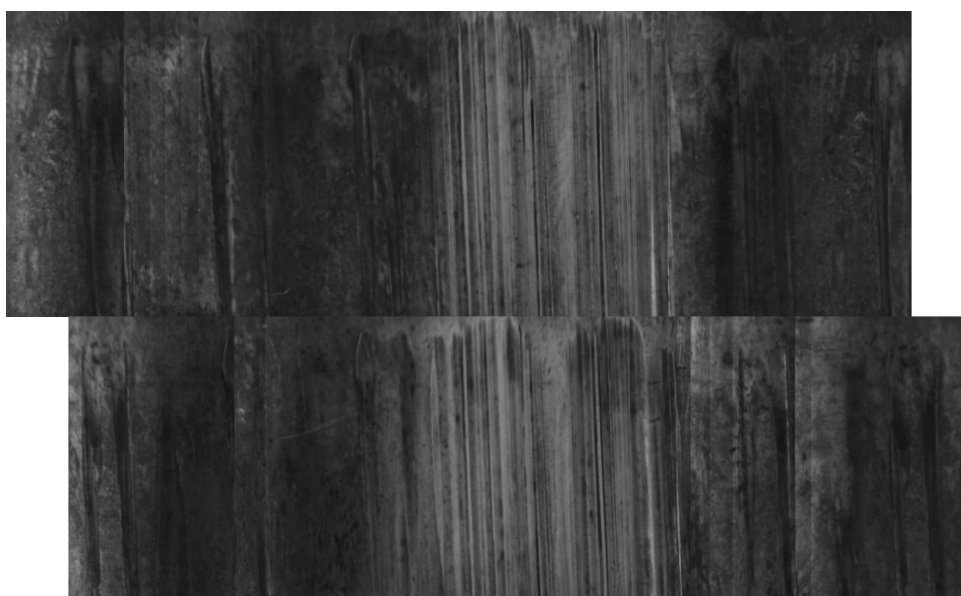


Figura 5.80 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 1 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

A figura 5.79 mostra o resultado da correlação cruzada entre os dois projéteis da arma 1. É possível ver uma parte da onda acima do nível de confiança e o restante se encontra próximo ou abaixo do zero, ou seja, a probabilidade das imagens dos projéteis se alinharem no pico da curva é considerável, mostrando que são altas as chances de os projéteis realmente terem origem de mesma arma.

A figura 5.80 mostra as imagens dos projéteis da arma 1 alinhados, sendo possível reparar como o deslocamento do pico da curva de correlação corresponde ao deslocamento necessário para alinhar as ranhuras.

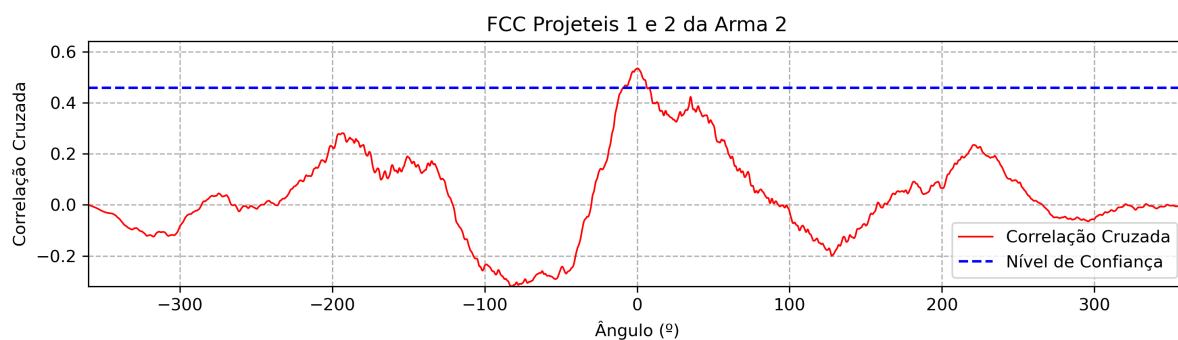


Figura 5.81 – Correlação Cruzada do Projétil 1 da Arma 2 com o Projétil 2 da Arma 2 - Fonte: Autor



Figura 5.82 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 2 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

É possível ver o resultado da correlação cruzada entre os projéteis da arma 2 na figura 5.81. Há dois picos próximos mais ressaltados, sendo que um deles passa do nível de confiança, e duas regiões expressivas acima do zero em posições semelhantes ao gráfico anterior. Como antes, desconsiderando-se o conhecimento da origem dos projéteis, é possível concluir pela análise do gráfico que existe alta chance de os projéteis terem sido expelidos pela mesma arma.

Na figura 5.82 é possível ver as imagens dos projéteis da arma 2 alinhados. O maior pico da correlação indica que as imagens estariam praticamente alinhadas, porém é possível ver que o alinhamento da imagem seguiu o segundo maior pico. Esse acontecimento se deve ao grande reflexo na borda esquerda das duas imagens, fazendo o sinal ficar muito parecido nas bordas por apresentarem as mesmas intensidades nos *pixels*.

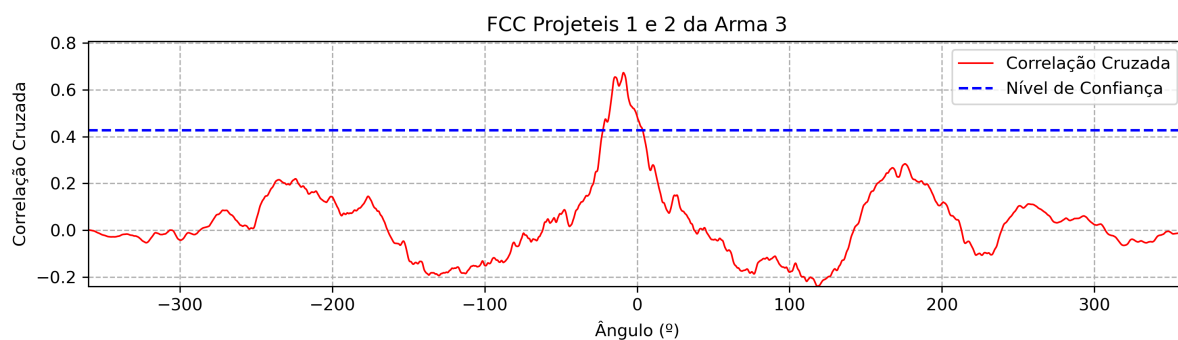


Figura 5.83 – Correlação Cruzada do Projétil 1 da Arma 3 com o Projétil 2 da Arma 3

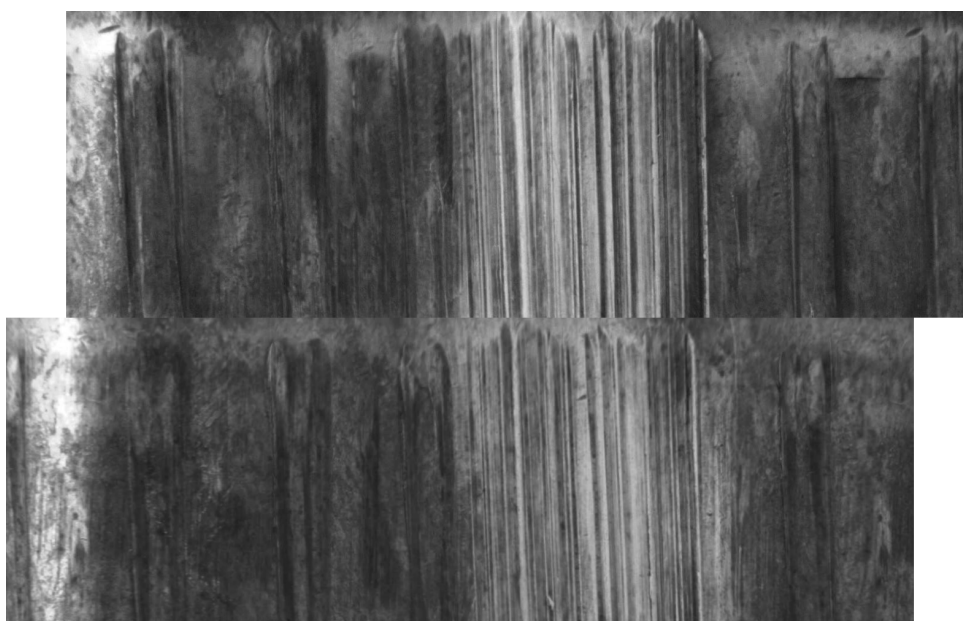


Figura 5.84 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 3 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

Na figura 5.83 visualiza-se um gráfico novamente semelhante aos anteriores, apresentando um pico superior aos demais pontos, tendo regiões acima do zero em posições similares aos gráficos anteriores. Isso mostra novamente a chance considerável de se tratar de dois disparos da mesma arma quando se desconsidera o conhecimento da origem dos projéteis.

Ao visualizar a figura 5.84, pode-se ver como o alinhamento dos projéteis expelidos pela arma 3 está seguindo o deslocamento à esquerda do pico da correlação cruzada.

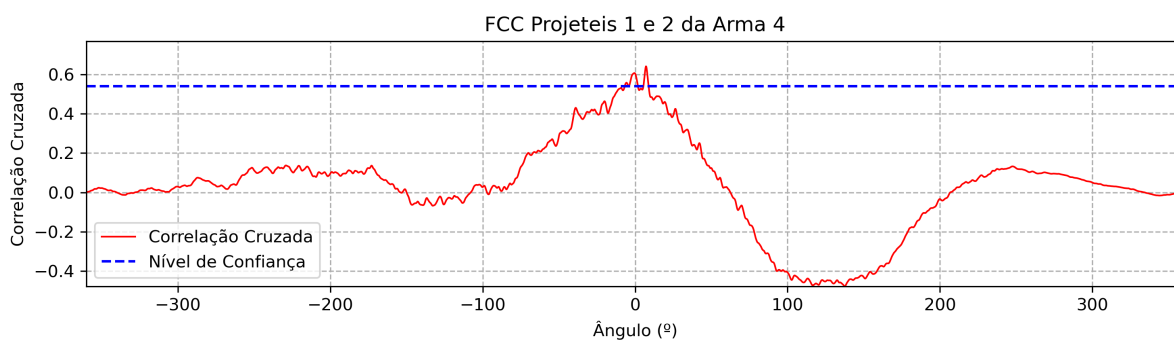


Figura 5.85 – Correlação Cruzada do Projétil 1 da Arma 4 com o Projétil 2 da Arma 4 - Fonte: Autor



Figura 5.86 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 4 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

O gráfico da correlação cruzada entre os projéteis da arma 4 mostrado na figura 5.85 mostra uma pequena diferença com relação aos outros na posição das curvas acima do zero, porém ainda apresenta o pico no centro e as duas curvas nas laterais como anteriormente. Este também é um caso onde há boa possibilidade de serem projéteis provenientes da mesma arma, quando se analisa o gráfico e desconsidera o conhecimento da origem dos projéteis.

A leve variação do pico da correlação cruzada visto no gráfico foi aplicado para alinhar as imagens dos projéteis da arma 4 como mostra a figura 5.86.

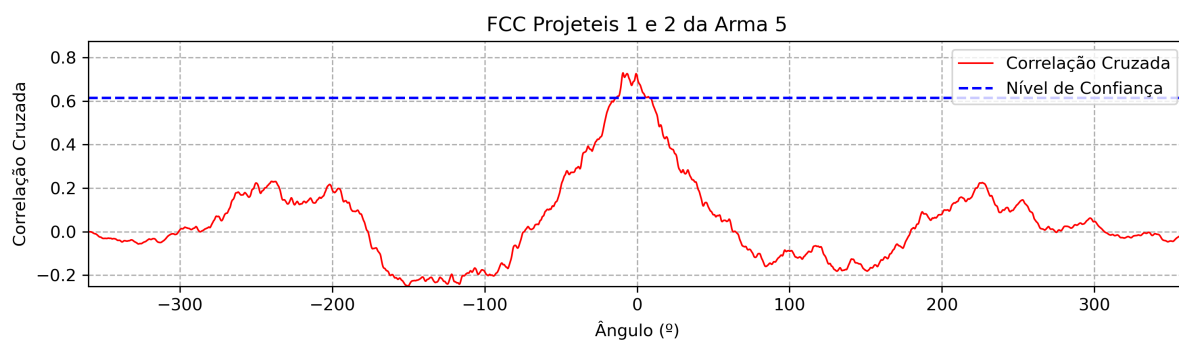


Figura 5.87 – Correlação Cruzada do Projétil 1 da Arma 5 com o Projétil 2 da Arma 5 - Fonte: Autor

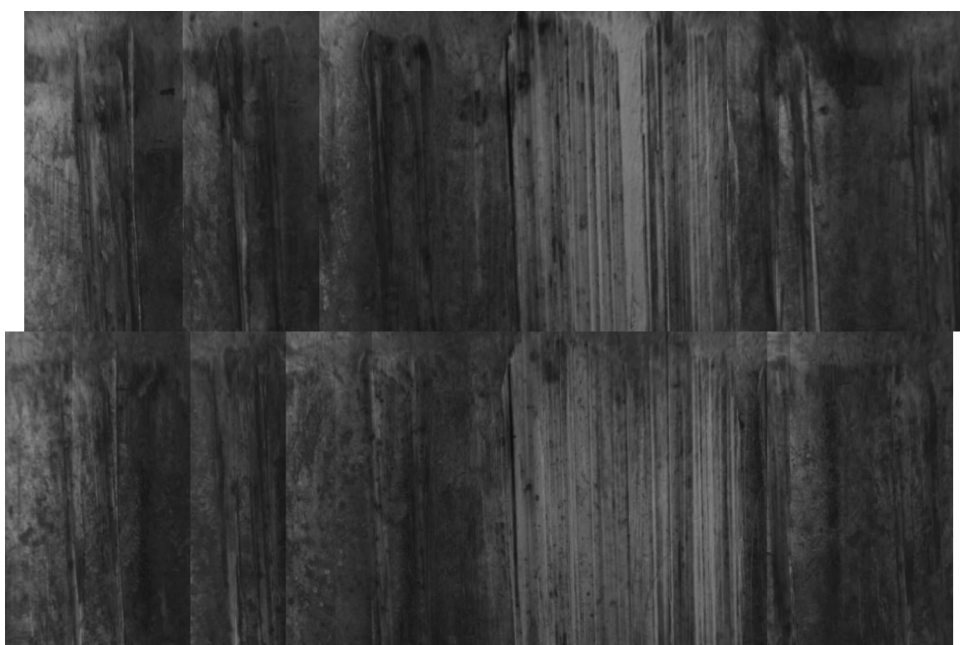


Figura 5.88 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 5 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

O pico ressaltado com duas curvas nas laterais um pouco acima do zero também são visíveis na figura 5.87, representando a correlação cruzada entre os projéteis disparados pela arma 5. Essa configuração, quando se desconsidera o conhecimento da origem dos projéteis, também permite ter certa confiança de que os projéteis analisados são de mesma arma.

A figura 5.88 mostra o pequeno alinhamento realizado à esquerda seguindo o pequeno desvio do gráfico de correlação cruzada à esquerda.

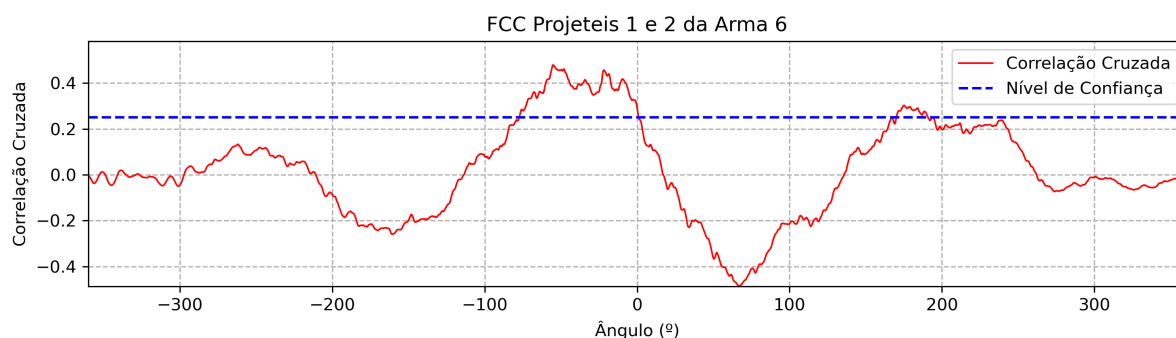


Figura 5.89 – Correlação Cruzada do Projétil 1 da Arma 6 com o Projétil 2 da Arma 6 - Fonte: Autor

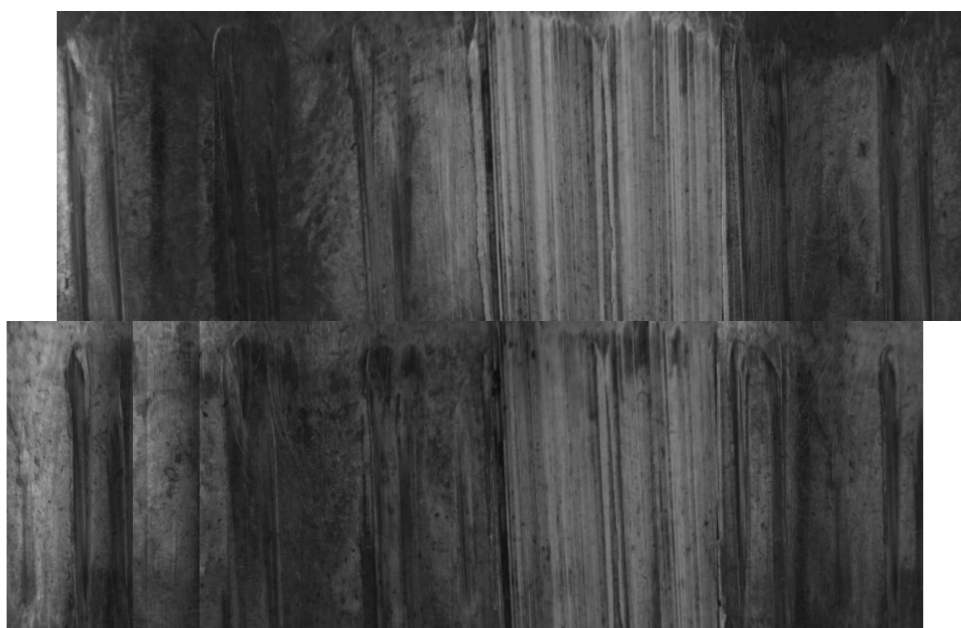


Figura 5.90 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 6 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

Ao analisar o gráfico presente na figura 5.89, percebe-se uma parte mais elevada da curva que passa do nível de confiança, uma segunda parte elevada que começa a passar do nível de confiança e uma terceira parte elevada minimamente, bem próxima ao zero. Este gráfico, ao considerar o desconhecimento da origem dos projéteis, traria certa dúvida quando aos projéteis serem ou não da mesma arma. Com não há mais fatores confundidores no gráfico, é possível concluir que há certa chance de serem projéteis de mesma arma, sendo necessário confirmar com análise humana.

Os projéteis foram alinhados e constatou-se que a parte mais alta da curva de correlação cruzada consegue representar o deslocamento entre as imagens para que fosse possível alinhá-las. O alinhamento pode ser visto na figura 5.90.

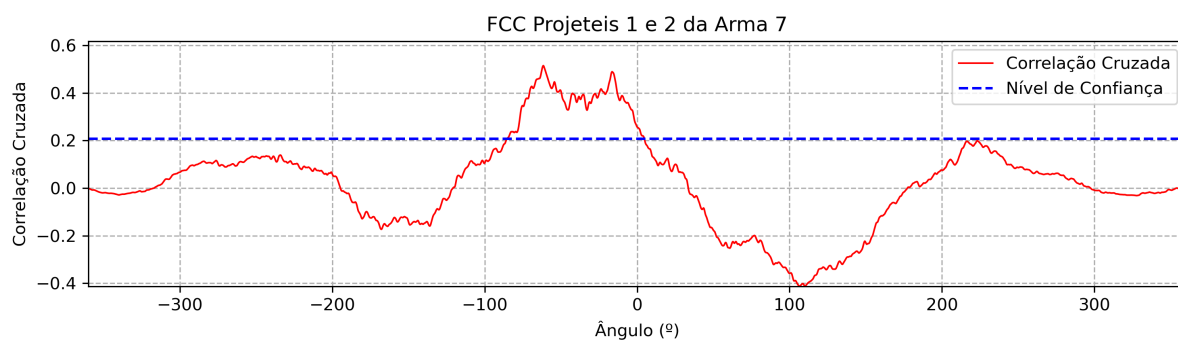


Figura 5.91 – Correlação Cruzada do Projétil 1 da Arma 7 com o Projétil 2 da Arma 7 - Fonte: Autor

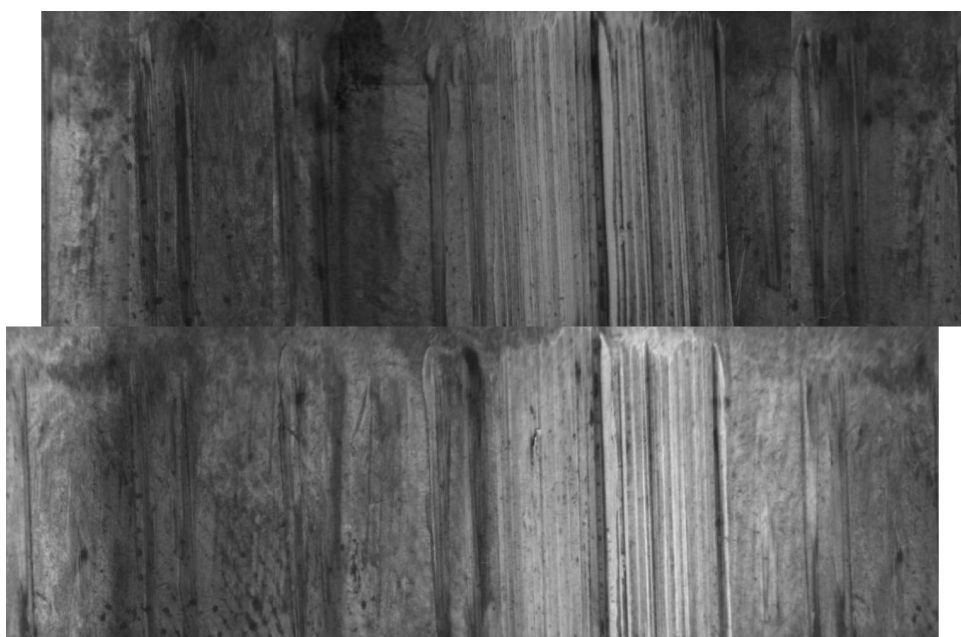


Figura 5.92 – Alinhamento do Projétil 2 (embaixo) com o Projétil 1 (em cima) da Arma 7 - Fonte: Modificado de (SILVINO JUNIOR, 2010)

A figura 5.91 mostra o gráfico de correlação entre os projéteis propélidos pela arma 7. Estrutura semelhante a mencionada anteriormente é visualizada, há uma parte elevada que passa do nível de confiança e há duas partes que passam um pouco do zero. A parte que está acima do nível de confiança apresenta dois picos semelhantes, podendo qualquer um representar o deslocamento de uma curva em relação a outra. Considerando o desconhecimento da origem dos projéteis, é razoável considerar que os projéteis podem ter origem de mesma arma.

As imagens dos projéteis da arma 7 foram alinhadas (figura 5.92) e é possível ver que o deslocamento entre as imagens está relacionado ao pico mais à direita do gráfico de correlação cruzada. Ao olhar a figura, é visível duas ranhuras mais expressivas e bem semelhantes na imagem de baixo, sendo possível que o pico à esquerda do gráfico represente um alinhamento com a ranhura errada, transladando a imagem mais à esquerda.

Após analisar todas as comparações entre projéteis de mesma arma pode-se notar que há a presença de pequenas curvas laterais acima de zero em todas as comparações aproximadamente na mesma posição. Foi investigado mais a fundo e percebeu-se que o brilho presente nas imagens tanto na borda esquerda como nas ranhuras principais se confundem já que apresentam intensidades de luz parecidas. Sendo assim, a correlação cruzada interpreta que, quando a borda esquerda de uma imagem se alinha com as ranhuras expressivas da outra, gera um semi-positivo de alinhamento, não sendo totalmente positivo pela falta de compatibilidade das demais partes das imagens. A figura 5.93 mostra esse alinhamento semi-positivo.



Figura 5.93 – Alinhamento Errado Semi-positivo - Fonte: Modificado de (SILVINO JUNIOR, 2010)

Como é possível reparar, dependendo da posição de onde forem retiradas as 10 linhas para gerar o sinal, é possível que as intensidades sejam um pouco compatíveis mesmo que com atraso entre os sinais. Esse efeito se deve a má iluminação utilizada para a captura das imagens, iluminação esta que gerou brilho excessivo em duas partes específicas de todas as imagens capturadas. Pelo problema constatado, acredita-se que basta regular a distribuição de luz corretamente e este problema observado será resolvido, porém, como as imagens são de autoria de João Bosco (SILVINO JUNIOR, 2010), não é possível melhorar as imagens para realizar uma melhor análise.

5.3.1.2 Comparações Entre Amostras de Armas Diferentes

Além de realizar testes com projéteis de mesma arma, é necessário que sejam feitos teste com projéteis de diferentes armas para que seja possível comparar os resultados e

compreender melhor os resultados. Então, aqui serão expostas apenas algumas das possíveis comparações dos projéteis dois a dois visto que são muitas possibilidades.

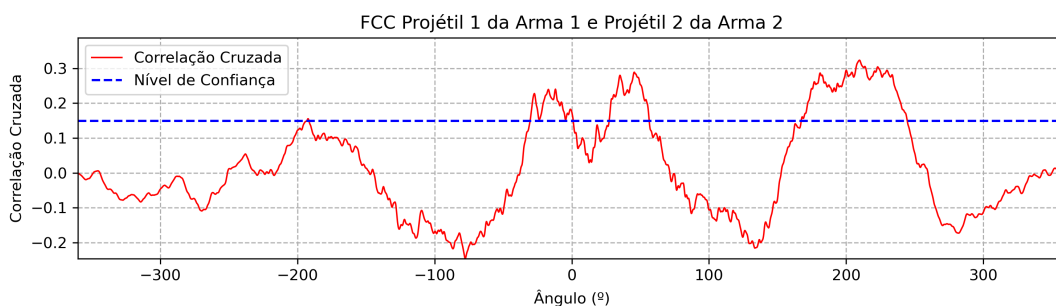


Figura 5.94 – Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 2 da Arma 2 - Fonte: Autor

Diferente do que foi visto anteriormente, a figura 5.94 mostra um gráfico de correlação cruzada (entre o projétil 1 da arma 1 e o projétil 2 da arma 2) com três partes mais altas e acima do nível de confiança. A estrutura, de certa forma, lembra vagamente as estruturas vistas anteriormente, onde era visto três partes mais ressaltadas, porém aqui a amplitude apresentada é pequena, além de que não há um pico mais expressivo que os demais. Dessa forma, considerando que não se sabe a origem dos projéteis, é mais provável considerar que os projéteis não estão relacionados.

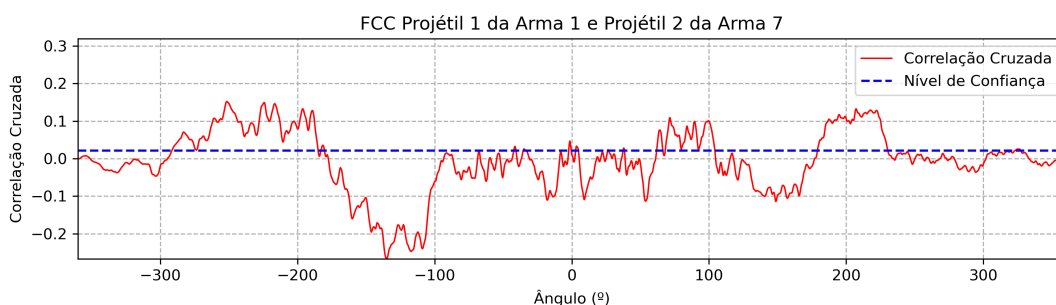


Figura 5.95 – Correlação Cruzada do Projétil 1 da Arma 1 com o Projétil 1 da Arma 7 - Fonte: Autor

Da mesma forma que foi visto anteriormente, o sinal de correlação cruzada visto na figura 5.95 mostra alguns picos acima do zero que apresentam amplitudes próximas, as quais são baixas, além de apresentar vale mais expressivo que as amplitudes. Aqui também é razoável dizer que os projéteis não se relacionam, considerando que a origem dos mesmos não é conhecida.

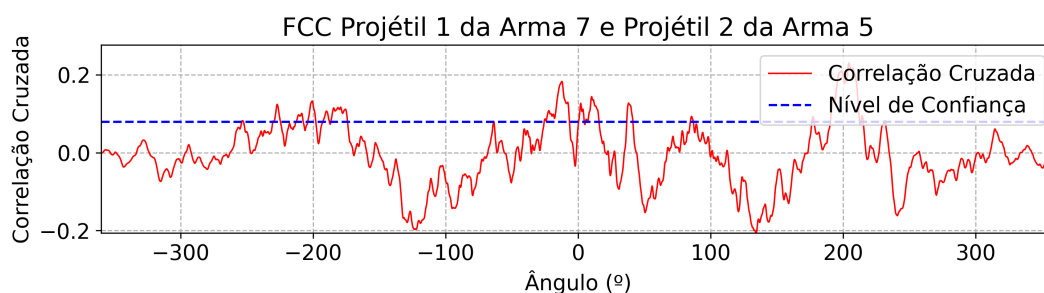


Figura 5.96 – Correlação Cruzada do Projétil 1 da Arma 7 com o Projétil 2 da Arma 5

A figura 5.96 mostra um gráfico de correlação cruzada com um pico ligeiramente mais expressivo no atraso de ângulo próximo a 205°, porém a amplitude desse pico é baixa, havendo vales tão expressivos quanto tal pico. Dessa forma, a probabilidade de as amostras não estarem relacionadas é relativamente grande, sendo considerado o desconhecimento da origem do projéteis para tal análise.

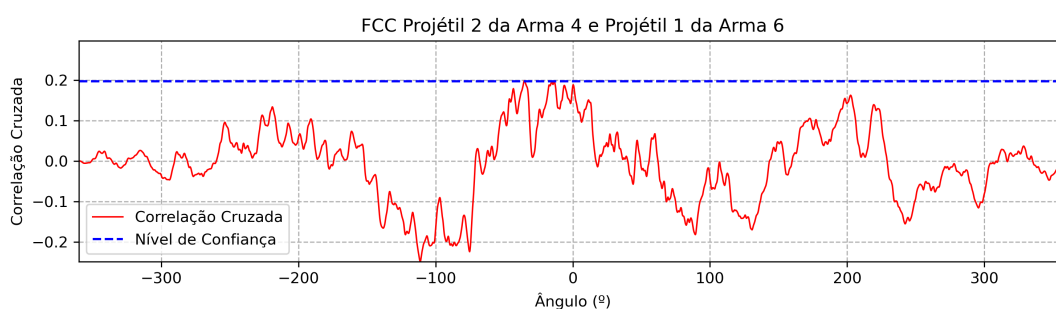


Figura 5.97 – Correlação Cruzada do Projétil 2 da Arma 4 com o Projétil 1 da Arma 6 - Fonte: Autor

A correlação cruzada mostrada na figura 5.97 volta a trazer pouca amplitude em seus picos e, neste caso, um dos vales chega a ser mais expressivo que todos seus picos, mostrando a maior chance de os sinais analisados serem inversamente compatíveis do que serem diretamente compatíveis. Ainda, ambas as amplitudes são baixas, o que sugere, considerando não saber a origem de ambos os projéteis, que os projéteis não estão relacionados.

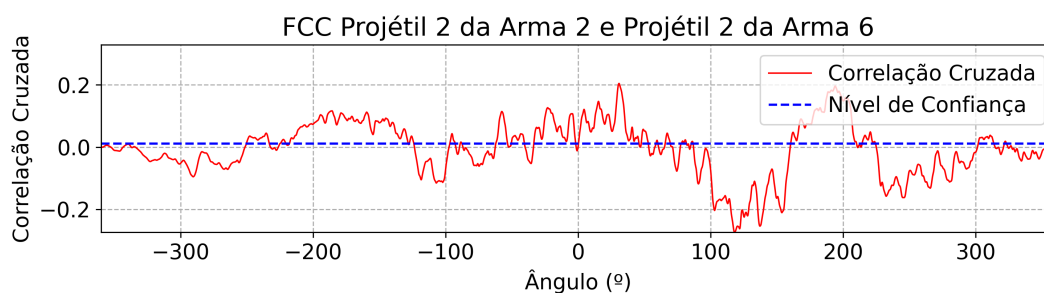


Figura 5.98 – Correlação Cruzada do Projétil 2 da Arma 2 com o Projétil 2 da Arma 6 - Fonte: Autor

O gráfico acima (figura 5.98) é mais um exemplo em que a correlação cruzada resultante é pouco expressiva, tendo um vale mais expressivo que todos os picos, sendo todos de baixa amplitude. Novamente, considerando que se desconhece a origem dos projéteis, é provável que os projéteis analisados não seja compatíveis em nenhum tipo de atraso.

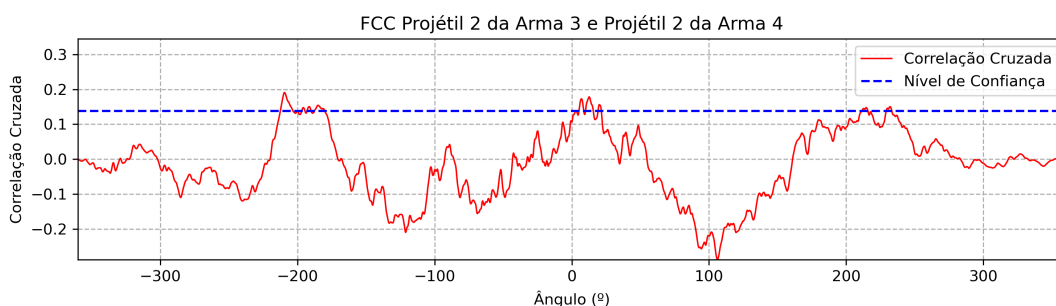


Figura 5.99 – Correlação Cruzada do Projétil 2 da Arma 3 com o Projétil 2 da Arma 4 - Fonte: Autor

Uma nova comparação gerou gráfico de correlação cruzada semelhante aos anteriores. A figura 5.99 mostra baixa amplitude, três picos com correlação cruzada muito semelhantes e um vale mais expressivo que tais picos. Este é mais um caso onde, considerando desconhecimento da origem dos projéteis, julga-se muito provável que os presentes projéteis não se relacionem em nenhuma posição.

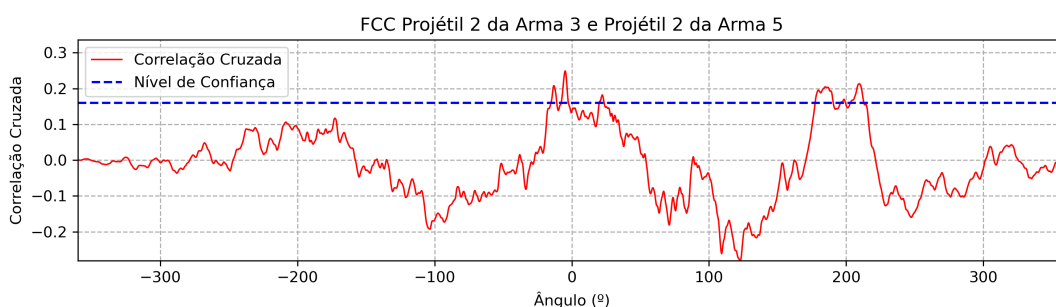


Figura 5.100 – Correlação Cruzada do Projétil 2 da Arma 3 com o Projétil 2 da Arma 5 - Fonte: Autor

Em mais uma correlação cruzada calculada entre dois projéteis (figura 5.100), o resultado mostrou pouca relação entre os projéteis comparados. Picos com baixa amplitude e vale mais expressivo que qualquer pico são características visíveis e que fortificam a probabilidade de que os projéteis não tenham a mesma origem, quando se analisa tal gráfico desconsiderando o conhecimento da origem dos projéteis testados.

Como foi possível reparar em todas as comparações nesta seção, as amplitudes foram todas baixas e, em geral, por volta de 0,2, além de sempre serem vistas alguns picos, todos com amplitudes próximas, nunca havendo picos mais expressivos como era visto na seção anterior. Ainda, nesta seção foram vistos diversas vezes vales mais expressivos que os picos, o que não foi visto na seção anterior. Além disso, na seção anterior foi notada uma semelhança

física entre todos os gráficos, uma regularidade que foi explicada, baseada na iluminação dos projéteis, porém nesta seção não houve qualquer semelhança no formato dos gráficos, apenas entre as suas amplitudes, picos e vales.

Sendo assim, é possível dizer que a correlação cruzada é um método viável para comparar projéteis e auxiliar o(a) perito(a) a chegar a alguma conclusão. Ressalta-se que em todos os casos é necessário que o(a) perito(a) avalie os resultados, sendo tal profissional o(a) responsável pelo laudo final.

5.3.2 Comparação Estatística

A comparação estatística é o segundo método que pretendeu-se testar para comparação entre os projéteis. Para isso, é necessário realizar os cálculos das distâncias entre os vetores compostos pelas medidas estatísticas dos sinais extraídos e gerar a distribuição em densidade de probabilidade em um gráfico. Alguns gráficos foram feitos utilizando as distâncias Euclidiana, de Manhattan e de Minkowski e serão expostos a seguir.

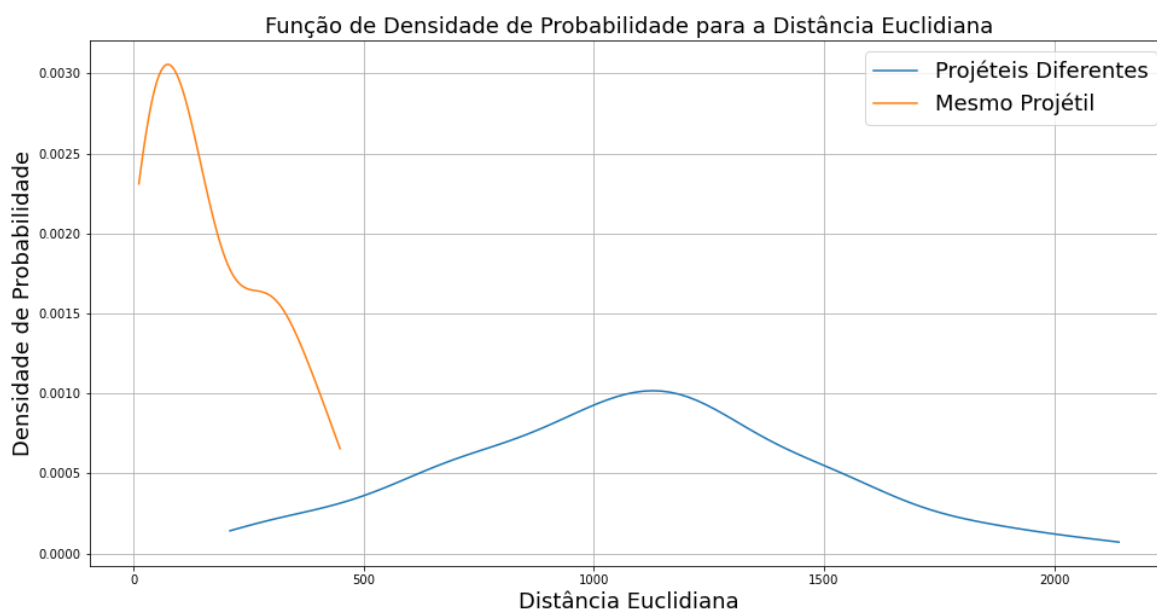


Figura 5.101 – Distância Euclidiana - Fonte: Autor

A figura 5.101 mostra o gráfico de distribuição de densidade de probabilidade gerado para a distância euclidiana entre os projéteis escaneados obtidos. É possível ver duas curvas distintas, onde a curva laranja representa as comparações entre projéteis de mesma arma e a curva azul representa comparações entre projéteis de armas diferentes. Como é possível reparar, a curva laranja apresenta valores pequenos de distância e a curva azul apresenta valores majoritariamente grandes. Esse comportamento é esperado dado que projéteis de mesma arma devem ser parecidos, gerando vetores parecidos e distâncias pequenas entre

eles, e projéteis de armas diferentes devem ser diferentes, gerando vetores diferentes e distâncias grandes. Esse comportamento deve ser o mesmo para os demais cálculos de distância utilizados.

É possível reparar como a curva de projéteis diferentes ficou baixa e larga e suspeita-se que o motivo é a má iluminação utilizada para capturar as imagens. Todas as imagens apresentam um maior brilho na lateral esquerda e um pouco a direita do centro da imagem, onde as ranhuras estão mais definidas. Embora na segunda parte comentada a luz tenha definido mais as ranhuras, ela gerou uma região com intensidade de iluminação semelhante para todas as imagens escaneadas e isso prejudicou os algoritmos de comparação adotados visto que eles seguem a intensidade de cada *pixel*. Dessa forma, a distância entre alguns dos projéteis diferentes diminuiu, achatando a curva.

Sendo assim, caso um novo projétil seja testado para identificar a arma de onde saiu, se a distância entre o vetor gerado para ele e o vetor gerado para o projétil padrão (conhecido) estiver abaixo de 250 a probabilidade de serem projéteis de mesma origem é altíssima. Caso a distância esteja entre 250 e 500, a probabilidade de serem projéteis de mesma origem é alta, mas existe uma pequena probabilidade de serem de origens diferentes. Caso a distância seja maior que 500 é muito provável que os projéteis tenha origens diferentes. De qualquer forma, o método será útil para separar os projéteis mais compatíveis e o perito será responsável por finalizar o laudo.

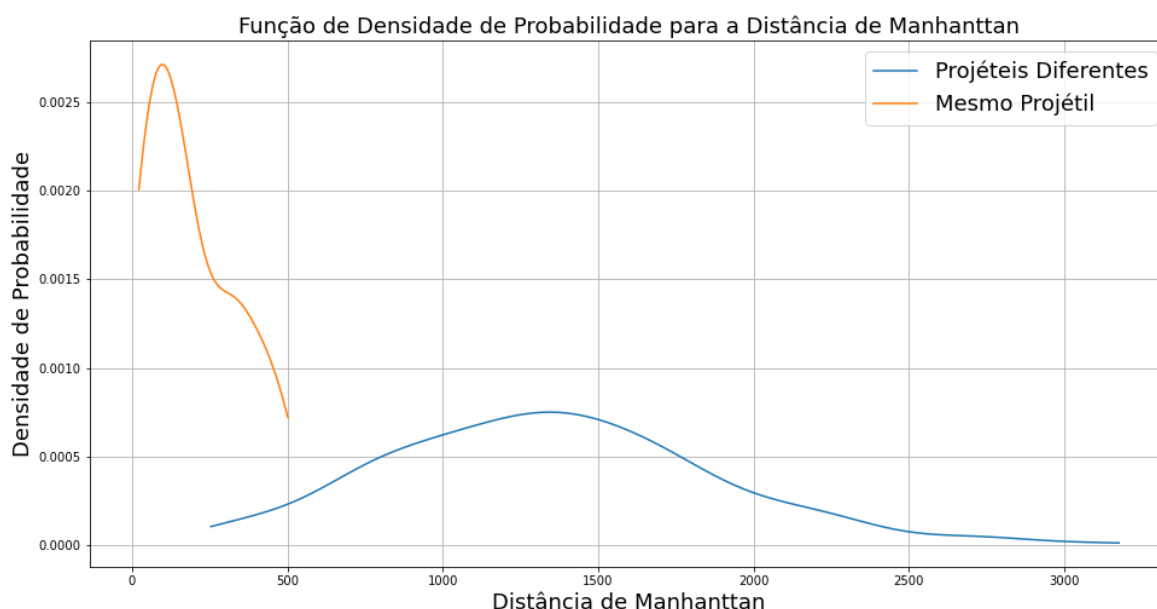


Figura 5.102 – Distância de Manhattan - Fonte: Autor

A distância de Manhattan também foi testada e a figura 5.102 mostra o resultado obtido. É notável a semelhança com o gráfico de distância euclidiana e já era esperado tal semelhança visto que os métodos são parecidos. A distância de Manhattan é tida pela soma das diferenças entre os itens diretos dos vetores, isto é, módulo da diferença entre os itens 1 mais o módulo da diferença entre os itens 2 e assim por diante. Já a distância Euclidiana eleva cada diferença dessa antes de somá-las, depois as soma e por último tira a raiz quadrada da soma. É possível ver tais equações detalhadas na seção 3.3.2.

Sendo assim, a distância de Manhattan será sempre maior que a distância Euclidiana e a diferença aumenta de acordo com que o valor de distância aumenta. É possível reparar esse efeito no gráfico da figura 5.102 onde as distâncias entre projéteis de mesma arma permaneceram entre 0 e 500, se aproximando mais do 500. Já as distâncias entre projéteis diferentes que estavam aproximadamente entre 250 e 2500 passaram a estar entre 250 e 3500, aumentando consideravelmente. Mesmo com tais mudanças, a configuração do gráfico permaneceu a mesma, podendo ser lido conforme descrito para o gráfico da distância Euclidiana.

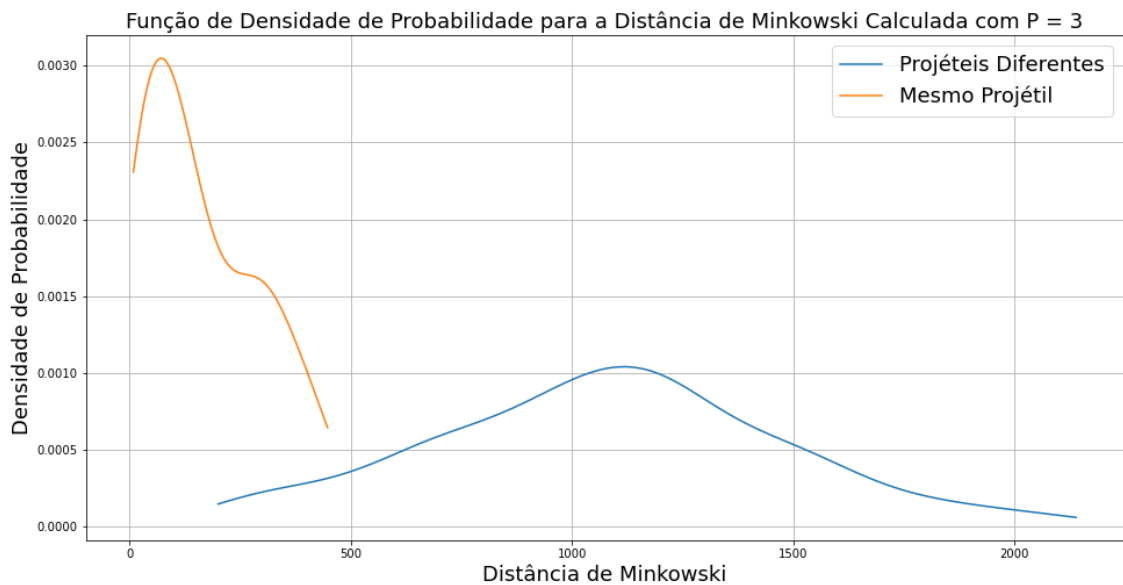
(a) $P = 3$

Figura 5.103 – Distâncias de Minkowski - Fonte Autor

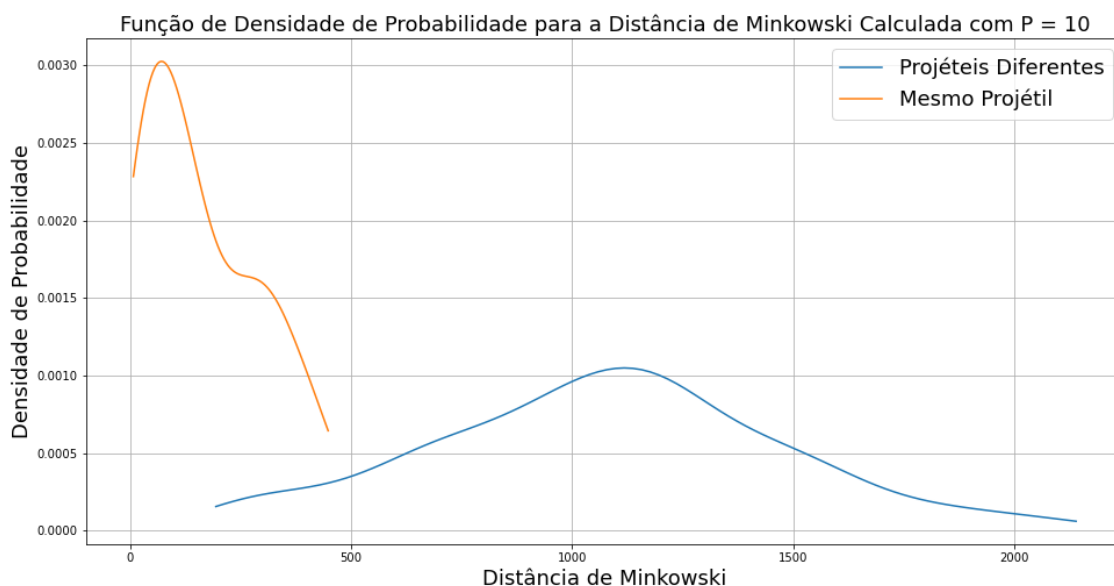
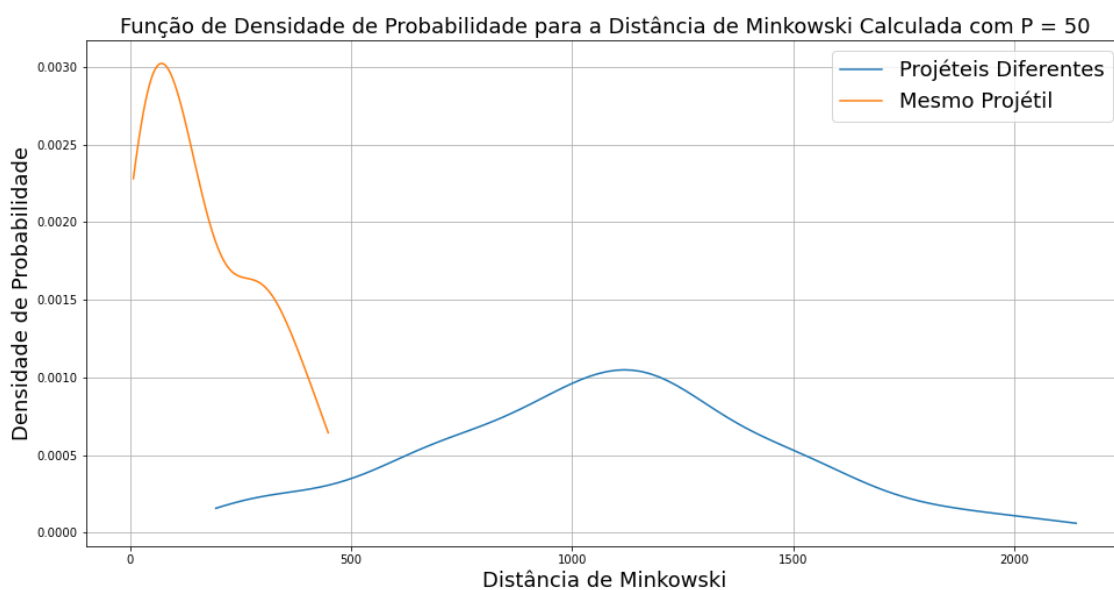
(b) $P = 10$ (c) $P = 50$

Figura 5.103 – Distâncias de Minkowski (continuação) - Fonte Autor

A figura 5.103 mostra três gráficos da distância de Minkowski. Foram plotados três gráficos pelo fato de que a distância referida depende de uma variável P que define o nível da distância. Esse método é semelhante à distância de Manhattan e à distância Euclidiana, sendo que caso o $P = 1$ tem-se a distância de Manhattan e caso o $P = 2$ tem-se a distância Euclidiana. Tal fato pode ser visualizado nas equações descritas na seção 3.3.2. Dessa forma, a distância de Minkowski terá valores inferiores aos da distância Euclidiana a partir de $P = 3$, sendo que quanto maior o P menos o gráfico muda. Os três gráficos obtidos apresentam $P = 3$,

$P = 10$ e $P = 50$ e esses valores foram escolhidos para demonstrar o efeito comentado. É possível ver como o gráfico de $P = 3$ já se assemelha muito ao gráfico de distância Euclidiana, com valores diminuindo ligeiramente. Os gráficos seguintes mudam muito pouco, sendo praticamente imperceptível a mudança entre os gráficos de $P = 10$ e $P = 50$. Dada a mesma estrutura que a vista anteriormente, o processo de comparação de um projétil se dá da mesma forma já comentada.

Com isso, é possível ver que este método também apresenta uma viabilidade quanto à comparação de projéteis e pode ser útil para auxiliar o(a) perito(a) em laudos periciais. Da mesma forma que dito antes para a correlação cruzada, a utilização deste método faz necessário o acompanhamento de um(a) perito(a) especializado(a), que será responsável por tomar as decisões para emissão do laudo final.

6 Considerações Finais

6.1 Conclusões e Discussões

Neste trabalho, foi apresentada uma investigação sobre a implementação da balística forense em campo, visando aproveitar a tecnologia e os avanços para aprimorar a dinâmica dessa área por meio de pré-análises. A motivação por trás desse projeto foi melhorar a perícia criminal diminuindo sua dependência de processos laboratoriais demorados, levando a possibilidade de haver resultados preliminares já em campo. Definiu-se o problema como a necessidade de um dispositivo capaz de realizar os confrontos balísticos ainda em campo, de forma rápida e confiável.

Os objetivos do projeto foram estabelecidos, abrangendo o objetivo geral de estudar, projetar e desenvolver um protótipo físico capaz de escanear projéteis e um software para processar imagens e analisar sinais retirados das ranhuras em projéteis resultantes de disparos com armas de fogo. Objetivos específicos foram definidos com relação à mecânica do dispositivo e aos estágios necessários para o software, passando por pré-requisitos essenciais que foram sempre levados em consideração. A partir de então, a estrutura do trabalho foi organizada para apresentar um fluxo coerente de informações, mostrando o conteúdo introdutório necessário, a base teórica necessária e a metodologia utilizada, chegando aos resultados finais.

A revisão bibliográfica trouxe informações para embasar como funcionam as armas e as munições e mostrar a origem das marcas que são registradas nos projéteis, marcas essas utilizadas para realizar os confrontos balísticos. Ainda, foram abordados os métodos atuais utilizados pelos peritos para se realizar confrontos balísticos, o método manual e os métodos automatizados.

Fundamentos teóricos importantes foram estabelecidos, abrangendo tópicos como microscopia, imagens digitais, visão computacional, processamento de imagens e processamento de sinais, todos utilizados para concluir de início a fim a comparação balística. Tais conceitos foram ligados com a metodologia e foi mostrado como cada um se aplica para o objetivo desejado. O dispositivo físico foi montado e o software codificado baseado na teoria estudada, sendo possível atingir ótimos resultados.

A seção de resultados apresentou a implementação bem-sucedida das metodologias propostas, demonstrando o funcionamento da estrutura e dos algoritmos de processamento de imagens e das técnicas de análise de sinais. Conseguiu-se capturar as imagens, as quais foram processadas com precisão e foi gerada uma imagem completa da face lateral do projétil com sucesso. A imagem gerada passou pela extração de sinal, sendo obtido um sinal refinado

do projétil escaneado. Pela falta de projéteis, foram utilizadas imagens de outro projeto e delas foram também extraídos sinais com sucesso. Tais sinais representaram as ranhuras marcadas nos projéteis, sendo úteis para comparar os projéteis. Dois tipos de comparações entre os sinais foram feitas, por meio de correlação cruzada e por meio de comparações estatísticas. Embora a correlação cruzada não tenha saído conforme esperado, foi possível corrigi-la e interpretar seus resultados, sendo possível relacionar ou diferenciar os projéteis corretamente. A comparação estatística também gerou resultados satisfatórios, trazendo gráficos de densidade de probabilidade que permitem diferenciar com boa confiança quando a comparação é entre projéteis de mesma arma ou armas diferentes. Dado o sucesso dos dois métodos de comparação, é interessante a utilização conjunta deles, trazendo mais informações para que o(a) perito(a) chegue a uma conclusão com mais facilidade.

Em conclusão, este trabalho explorou e aplicou diversas técnicas, gerando um protótipo funcional do dispositivo alvo, contribuindo para o avanço da balística forense em campo. O dispositivo e software desenvolvidos apresentaram resultados promissores e motivadores para a continuação do projeto. O dispositivo, mesmo apenas como protótipo, apresentou grande potencial para auxiliar investigações criminais, fornecendo análises preliminares em cenas de crimes e facilitando a identificação e captura de criminosos. Este trabalho pode servir de base para futuras melhorias e pesquisas e para isso serão oferecidas, na próxima seção, sugestões para aprimorar o dispositivo e software e estender sua aplicabilidade não só a projéteis como também a cápsulas. No geral, o projeto mostrou novas possibilidades para a evolução da balística forense, possibilitando novas abordagens funcionais e que prometem auxiliar o trabalho do(a) perito(a) em campo.

6.2 Perspectivas para Trabalhos Futuros

O trabalho realizado gerou bons resultados e teve um bom desenvolvimento, porém, por ser um protótipo base, há muito espaço para modificações, acréscimos e evoluções. Com este pensamento em mente, surgem propostas para futuros trabalhos:

- Melhoria da distribuição da iluminação, se possível utilizando um suporte em formato de semicírculo para que a luz, após ser difundida, fique o mais homogênea possível;
- Automatização do sistema de foco do microscópio e do sistema de ajuste de posição da iluminação e do microscópio;
- Troca do mandril de metal por um modelo impresso em 3D, sendo mais leve e mais versátil;
- Adaptar o mandril para acoplar cartuchos (estojos) além dos projéteis, sendo necessário também desenvolver o sistema de comparação de estojos;

- Acrescentar algum tipo de junta esférica ao mandril para poder movimentá-lo e ajustar a sua posição para casos de análise de estojos e de projéteis deformados;
- Buscar por microscópio com maior resolução, porém com o custo o mais baixo possível;
- Testar a correlação cruzada bidimensional como método de comparação entre os projéteis;
- Implementar um banco de dados para armazenar os projéteis e estojos que sejam escaneados, possibilitando o acesso rápido e fácil às informações necessárias;
- Implementar uma rede neural capaz de identificar qual o tipo de projétil está em análise, podendo restringir a gama de possibilidades que terão de ser analisadas;
- Desenvolver um sistema classificador para, de forma autônoma, o sistema ser capaz de classificar em ordem os possíveis\prováveis casos compatíveis;
- Desenvolver um modelo de comparador que utilize laser para escanear a superfície dos projéteis e estojos de forma a não depender da resolução ou do foco da câmera, da iluminação e do tipo de material com o qual o projétil e cápsulas são fabricados.

Referências

- AGUIRRE, L. A. **Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais**. Editora UFMG, 2004. Citado nas pp. 71, 72.
- AMAZONAS, T. D. J. D. E. D. Manual de armamento e manuseio seguro de armas de fogo, 2012. Citado na p. 23.
- AMMUNITION, C. G. **CATÁLOGO MUNIÇÕES E COMPONENTES - VAREJO**. 2023. Disponível em: <%5Curl%7Bhttps://www.cbc.com.br/wp-content/uploads/2021/09/catalogo-municoes-componentes-varejo.pdf%7D>. Citado na p. 82.
- AZIMI, E.; BEHRAD, A.; GHAZNAVI-GHOUSHCHI, M. B.; SHANBEHZADEH, J. A fully pipelined and parallel hardware architecture for real-time BRISK salient point extraction. **Journal of Real-Time Image Processing**, Springer, v. 16, p. 1859–1879, 2019. Citado na p. 60.
- BATISTA, R. B. d. S. Ensaio não destrutivo para recuperação de numeração obliterada em armas de fogo, 2019. Citado na p. 33.
- BAY, H.; ESS, A.; TUYTELAARS, T.; VAN GOOL, L. Speeded-up robust features (SURF). **Computer vision and image understanding**, Elsevier, v. 110, n. 3, p. 346–359, 2008. Citado nas pp. 52–55.
- BAY, H.; TUYTELAARS, T.; VAN GOOL, L. Surf: Speeded up robust features. In: SPRINGER. **COMPUTER Vision–ECCV 2006: 9th European Conference on Computer Vision**, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. 2006. P. 404–417. Citado nas pp. 53, 55.
- BOYD, S. P.; VANDENBERGHE, L. **Convex optimization**. Cambridge university press, 2004. Citado na p. 69.
- BUMAGA-S. **A4Tech PK-750MJ webcam PK-750MJ**. Jan. 2012. Disponível em: <%5Curl%7Bhttps://kaypu.com/gi/EN/4f4396143d72220a790f4b20/4711421754378/A4Tech%20PK%20750MJ%20webcam%7D>. Citado na p. 76.
- CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. In: SPRINGER. **COMPUTER Vision–ECCV 2010: 11th European Conference on Computer Vision**, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11. 2010. P. 778–792. Citado na p. 58.
- CARLUCCI, D. E.; JACOBSON, S. S. **Ballistics: theory and design of guns and ammunition**. CRC press, 2018. Citado na p. 26.
- COSTA, D. M. da. **BOAS PRÁTICAS PARA PERÍCIA FORENSE**, 2008. Citado na p. 35.

- DEBETIL, L. F. F. et al. A importância do exame pericial de comparação balística nas decisões dos julgadores em infrações penais com emprego de arma de fogo, 2015. Citado na p. 32.
- DK PUBLISHING, I. **Firearms: An Illustrated History**. DK Publishing, 2014. Citado na p. 21.
- FACULDADE DE MEDICINA UNIVERSIDADE DO PORTO, F. **Sinal e Imagem**. 2006. Disponível em: <http://im.med.up.pt/sinal_imagem/sinal_imagem.html>. Citado nas pp. 70, 71.
- FAN, B.; WANG, Z.; WU, F. et al. **Local image descriptor: modern approaches**. Springer, 2015. v. 108. Citado nas pp. 61, 62.
- FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. Brasport, 1999. v. 1. Citado nas pp. 42–45.
- FIREARMS: An Illustrated History. DK Publishing, 2014. ISBN 9780744034691. Disponível em: <<https://books.google.com.br/books?id=Em3wDwAAQBAJ>>. Citado na p. 27.
- FORENSE, B. **Quante Armi Hanno Sparato?** Jul. 2017. Disponível em: <<https://www.balisticaforense.it/2017/07/11/quante-armi-hanno-sparato/>>. Citado na p. 31.
- FORSYTH, D. A.; PONCE, J. **Computer Vision A MODERN APPROACH**. Pearson Education, 2012. v. 2. Citado na p. 44.
- FREITAS. **Teoria do Armamento e Tiro - Arma Curta - Pistola**. Disponível em: <https://sindespe.org.br/portal/wp-content/uploads/2020/08/Apostila_Arma_Curta-depen-pr.pdf>. Citado na p. 26.
- FREITAS, J. C. D. d. **Identificação de assinaturas químicas em resíduos de disparos de arma de fogo em diferentes alvos**. 2010. Tese (Doutorado) – Universidade de São Paulo. Citado na p. 33.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. Prentice Hall, 2002. v. 2. Citado na p. 43.
- GRANLUND, G. H.; KNUTSSON, H. **Signal processing for computer vision**. Springer Science & Business Media, 2013. Citado na p. 71.
- HARRIS, C. G.; STEPHENS, M. J. A Combined Corner and Edge Detector. In: ALVEY Vision Conference. 1988. Citado na p. 57.
- HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. Cambridge university press, 2003. Citado nas pp. 63, 64, 66.
- HEARD, B. J. **Handbook of firearms and ballistics: examining and interpreting forensic evidence**. John Wiley & Sons, 2011. v. 1. Citado na p. 31.

- HECHT, E. **Optics**. Pearson Education India, 2012. Citado nas pp. 37–39.
- ISHENGOMA, F. R. Authentication system for smart homes based on ARM7TDMI-S and IRIS-fingerprint recognition technologies. **arXiv preprint arXiv:1410.0534**, 2014. Citado nas pp. 68, 69.
- KIATRONICS. **28BYJ-48 – 5V Stepper Motor**. Disponível em: <%5Curl%7Bhttps://pdf1.alldatasheet.com/datasheet-pdf/view/1568317/ETC/28BYJ-48.html%7D>. Citado na p. 79.
- LEE, D.-J.; FULLER, S. G.; MCCOWN, A. S. Optimization and Implementation of Synthetic Basis Feature Descriptor on FPGA. **Electronics**, MDPI, v. 9, n. 3, p. 391, 2020. Citado na p. 56.
- LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. BRISK: Binary robust invariant scalable keypoints. In: IEEE. 2011 International conference on computer vision. 2011. P. 2548–2555. Citado nas pp. 59–61, 63.
- LONDON, B.; STONE, J.; UPTON, J. **PHOTOGRAPHY**. Pearson Education, 2017. v. 12. Citado nas pp. 41, 42.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, p. 91–110, 2004. Citado nas pp. 46–52.
- MENKE, W. Environmental data analysis with MATLAB: principles, applications, and prospects. (**No Title**), 2022. Citado na p. 71.
- MURPHY, D. B. **FUNDAMENTALS of LIGHT MICROSCOPY and ELETRONIC IMAGING**. Wiley-Liss, 2001. v. 1. Citado na p. 40.
- NETO, C. F. P. **Cartuchos, Pólvoras e Projéteis - Noções Básicas**. 2021. Disponível em: <%5Curl%7Bhttps://portaldotiro.com/artigos-tecnicos/municao/cartuchos-polvoras-e-projeteis-nocoes-basicas/%7D>. Citado na p. 23.
- NORRIS, J. **Artillery: a history**. The History Press, 2011. v. 1. Citado na p. 21.
- O'NEILL, B. **Elementary differential geometry**. Elsevier, 2006. Citado nas pp. 67, 68.
- OLIVEIRA JÚNIOR, M. L. L. d. Estudo da confiabilidade de um sistema de identificação balística para estojos e projéteis de munições calibre. 38, 2015. Citado na p. 22.
- OLIVEIRA NETO, A. S. d. O estudo da Balística Forense como elemento essencial na solução dos crimes que envolvem armas de fogo. Universidade Federal de Campina Grande, 2004. Citado nas pp. 28, 29, 35.
- PÁDUA, N. V. d. et al. Invariantes geométricos de pares de curvas planares. Universidade Federal de Goiás, 2020. Citado na p. 63.
- PAULY, R. A.; PAULY, R. **Firearms: the life story of a technology**. Greenwood Publishing Group, 2004. Citado na p. 26.

- PEDROTTI, F. L.; PEDROTTI, L. M.; PEDROTTI, L. S. **Introduction to optics**. Cambridge University Press, 2017. Citado na p. 37.
- PEREIRA, C. B. C. A utilização da química forense na investigação criminal. **Monografia**, 2010. Citado na p. 29.
- PROAKIS, J. G. **Digital signal processing: principles, algorithms, and applications**, 4/E. Pearson Education India, 2007. Citado na p. 72.
- ROCHA, A. F. da; CARVALHO, J. L. A. de; AZEVEDO BERGER, P. de; OLIVEIRA NASCIMENTO, F. A. de. Processamento de Sinais Biológicos. Citado na p. 70.
- ROSIN, P. L. Measuring corner properties. **Computer Vision and Image Understanding**, Elsevier, v. 73, n. 2, p. 291–307, 1999. Citado na p. 58.
- ROSTEN, E.; DRUMMOND, T. Machine learning for high-speed corner detection. In: SPRINGER. **COMPUTER Vision–ECCV 2006: 9th European Conference on Computer Vision**, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. 2006. P. 430–443. Citado na p. 56.
- ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 32, n. 1, p. 105–119, 2008. Citado na p. 57.
- RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. ORB: An efficient alternative to SIFT or SURF. In: IEEE. 2011 International conference on computer vision. 2011. P. 2564–2571. Citado nas pp. 56–58.
- SÁ SOARES, F. de. **Armamento e Tiro**. 2002a. Disponível em: <%5Curl%7Bhttp://www3.dsi.uminho.pt/academiamilitar/2002/pdf/Powerpoint/ARMAS_FOG_AUTOM.pdf%7D>. Citado nas pp. 27, 28.
- SÁ SOARES, F. de. **Glock**. 2002b. Disponível em: <%5Curl%7Bhttp://www3.dsi.uminho.pt/academiamilitar/2002/pdf/Powerpoint/ARMAS_FOG_AUTOM.pdf%7D>. Citado na p. 27.
- SANTOS, L. S. dos. **Avaliação do sistema de identificação balística - Evofinder**. 2015. Disponível em: <%5Curl%7Bhttps://docplayer.com.br/1537878-Pcf-lehi-sudy-dos-santos-instituto-nacional-de-criminalistica-area-de-pericias-em-balistica-forense.html%7D>. Citado nas pp. 33, 35, 36.
- SANTOS, L. S. d. Avaliação de parâmetros que afetam a efetividade da comparação balística automatizada, 2015. Citado nas pp. 22, 23, 25, 28–30, 34–36.
- SANTOS, M. C. dos; ROCHA, A. Revisão de Conceitos em Projeção, Homografia, Calibração de Câmera, Geometria Epipolar, Mapas de Profundidade e Varredura de Planos. **Unicamp, Campinas, Tech. Rep**, 2012. Citado na p. 64.

- SATURNO, A. **Humanos 2.0 | Aprenda a construir um microscópio caseiro utilizando uma webcam**. Fev. 2018. Disponível em: <%5Curl%7Bhttps://canaltech.com.br/inovacao/humanos-20-aprenda-a-construir-um-microscopio-caseiro-utilizando-uma-webcam-108506/%7D>. Citado na p. 75.
- SHARMA, P. **Understanding Distance Metrics Used in Machine Learning**. Fev. 2020. Disponível em: <%5Curl%7Bhttps://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/#h-minkowski-distance%7D>. Citado na p. 69.
- SILVA, D. C. M. da. **Máquina Fotográfica**. Dez. 2012. Disponível em: <%5Curl%7Bhttps://mundoeducacao.uol.com.br/fisica/maquina-fotografica.htm%7D>. Citado na p. 39.
- SILVA, I. O. Munições e explosivos: histórico, química e estado da arte no Brasil, 2022. Citado nas pp. 22–26.
- SILVEIRA, L. **Canos raiados e canos de alma lisa**. 2013. Disponível em: <%5Curl%7Bhttps://defesa.org/dwp/canos-raiados-e-canos-de-alma-lisa/%7D>. Citado na p. 30.
- SILVINO JUNIOR, J. B. Desenvolvimento de um sistema automatizado para captura e comparação de estriamentos de projéteis de armas de fogo. Universidade Federal de Minas Gerais, 2010. Citado nas pp. 93, 101, 114, 116–123, 180–183.
- SINGH, A. **SIFT Algorithm | How to Use SIFT for Image Matching in Python**. Out. 2019. Disponível em: <%5Curl%7Bhttps://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/%7D>. Citado nas pp. 48–52.
- SINGH, T. **Bresenham's Circle Drawing Algorithm Derivation**. Out. 2018. Disponível em: <%5Curl%7Bhttps://getsetcg.blogspot.com/2018/10/bresenhams-circle-drawing-derivation.html%7D>. Citado na p. 56.
- SO, A. T.-p.; CHAN, W. L. **Intelligent building systems**. Springer Science & Business Media, 1999. v. 5. Citado na p. 70.
- SZABO, F. **The linear algebra survival guide: illustrated with Mathematica**. Academic Press, 2015. Citado na p. 69.
- SZELISKI, R. **Computer Vision Algorithms and Applications**. Springer, 2011. v. 1. Citado nas pp. 44, 53.
- THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern recognition**. Elsevier, 2006. Citado na p. 68.
- TIRO, C. do. **Espingarda de Pressão ou Carabina de Pressão**. Nov. 2015. Disponível em: <%5Curl%7Bhttps://www.casadotiro.com.br/novidades-ver/espingarda-de-pressao-ou-carabina-de-pressao/33/%7D>. Citado na p. 25.

WALKER, R. E. **Cartridges and firearm identification**. CRC Press, 2012. Citado na p. [22](#).

Apêndices

Apêndice A – Desenhos Técnicos

Este apêndice mostrará os desenhos técnicos feitos a partir dos modelos 3D desenvolvidos neste trabalho. São ao todo 13 peças modeladas e elas serão dispostas a seguir na seguinte ordem:

- Base do Motor de Passo (Figura [A.104](#))
- Suporte do Rolamento (Figura [A.105](#))
- Encaixe entre Eixo do Motor, Rolamento e Parafuso do Mandril (Figura [A.106](#))
- Suporte do Microscópio (Figura [A.107](#))
- Base do Microscópio e da Iluminação Lateral (Figura [A.108](#))
- Trilho para o Microscópio (Figura [A.109](#))
- Encaixe do Trilho do Microscópio - "Vagão"(Figura [A.110](#))
- Base do Suporte de LEDs Lateral (Figura [A.111](#))
- Placa de LEDs das Laterais (Figura [A.112](#))
- Capa e Suporte de Difusor para Placa de LEDs das Laterais (Figura [A.113](#))
- Placa de LEDs para o Microscópio (Figura [A.114](#))
- Capa Traseira da Placa de LEDs do Microscópio (Figura [A.115](#))
- Suporte de Difusor para Placa de LEDs do Microscópio (Figura [A.116](#))

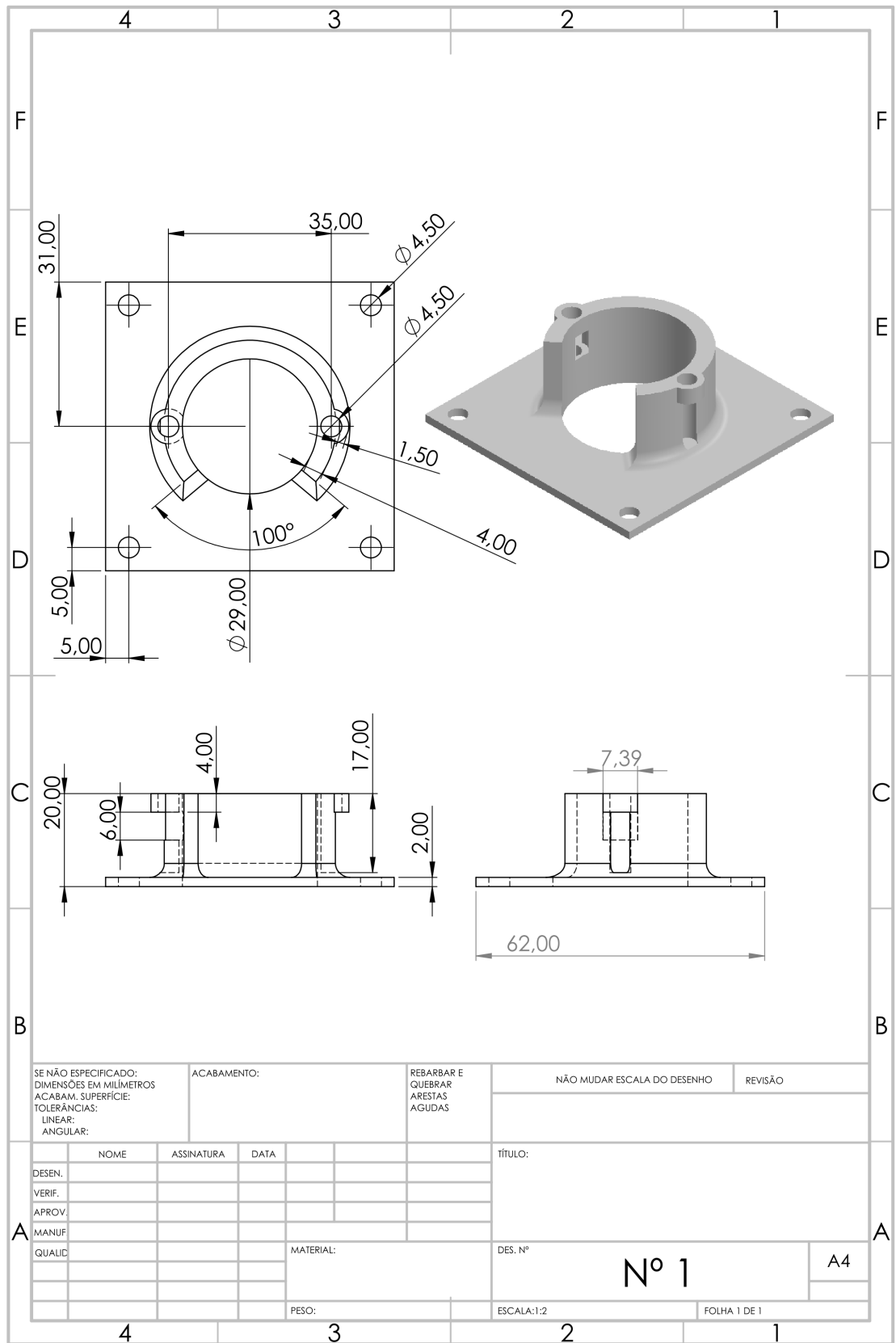


Figura A.104 – Desenho Técnico da Base do Motor de Passo

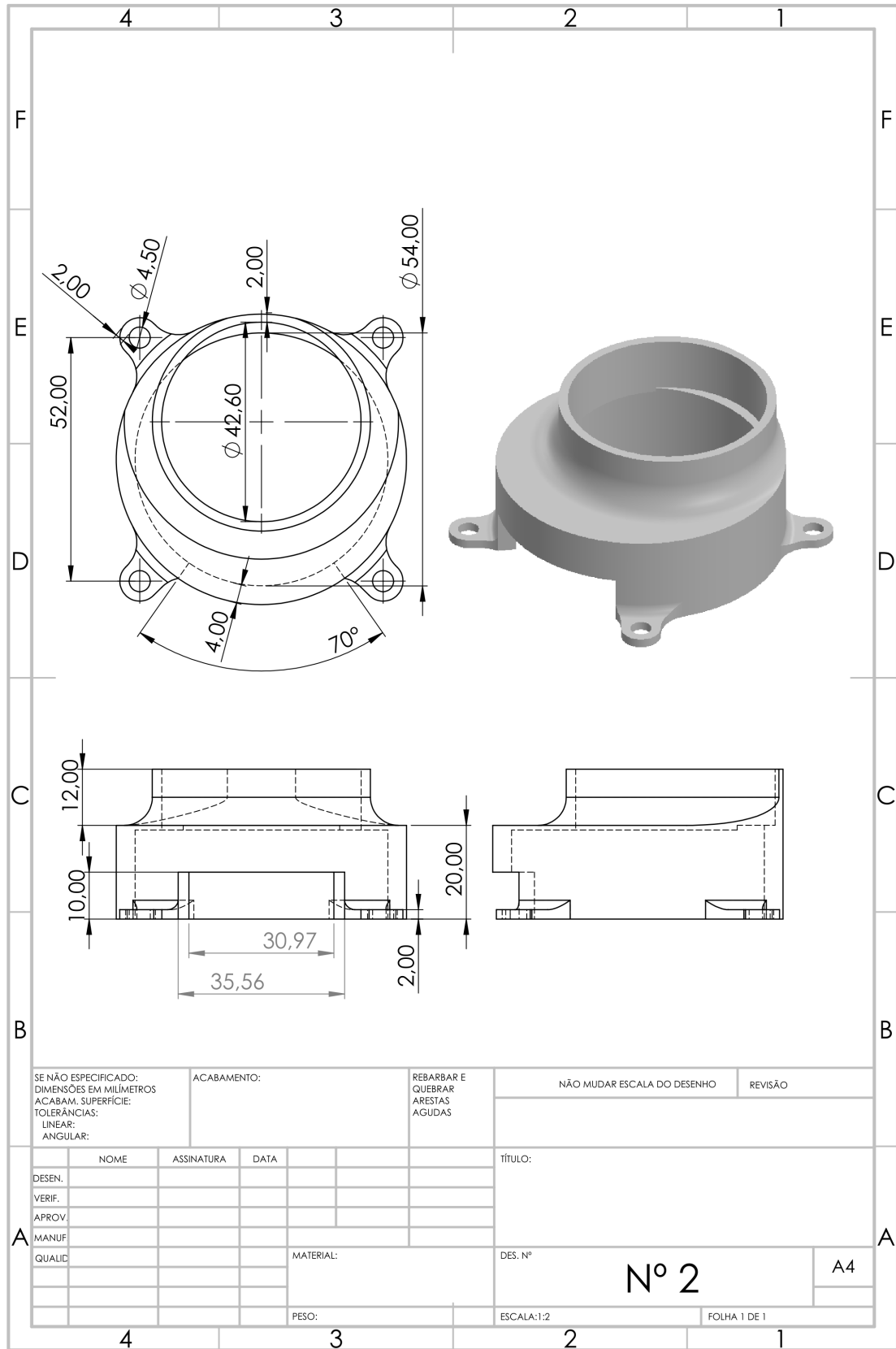


Figura A.105 – Desenho Técnico do Suporte do Rolamento

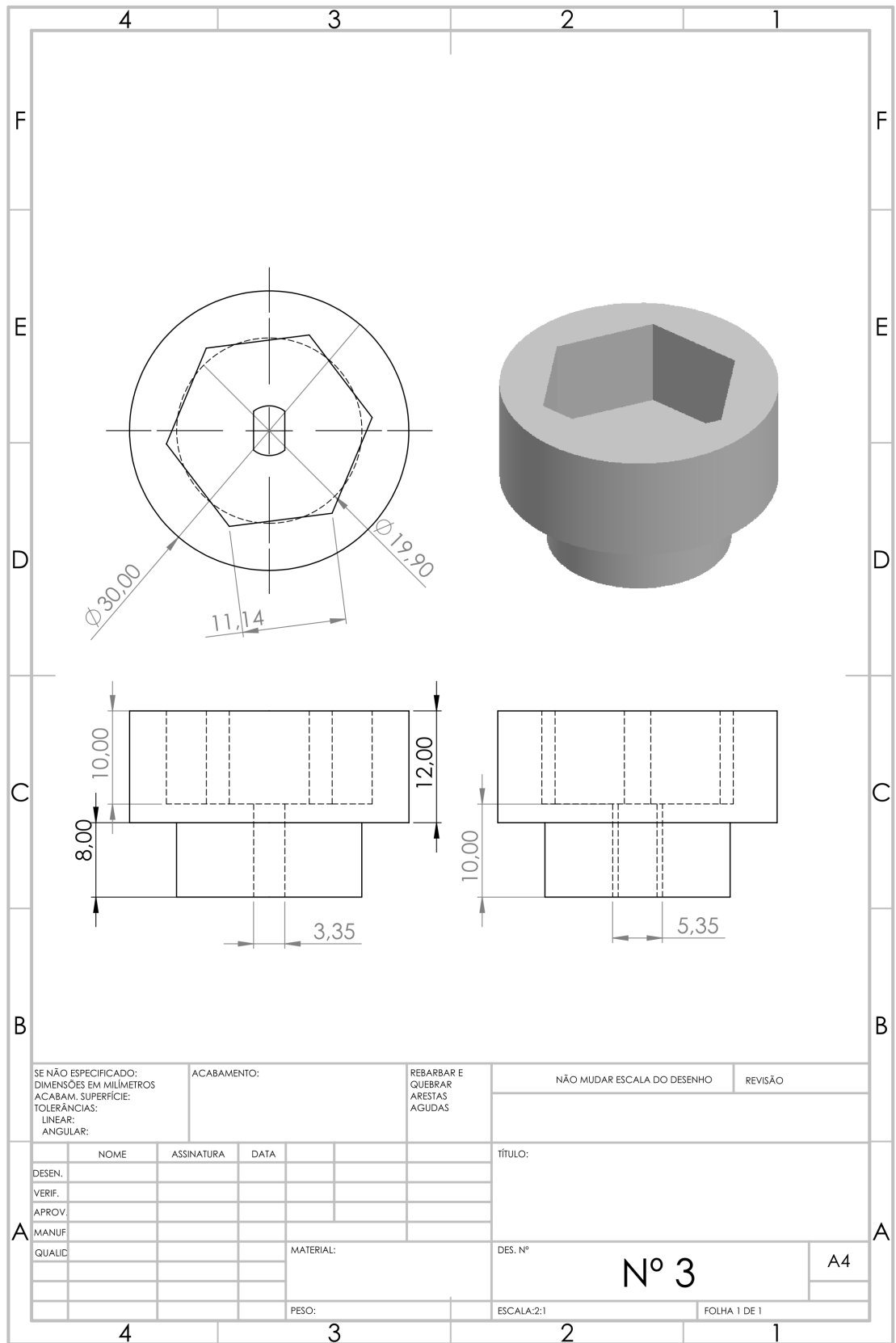


Figura A.106 – Desenho Técnico do Encaixe entre Eixo do Motor, Rolamento e Parafuso do Mandril

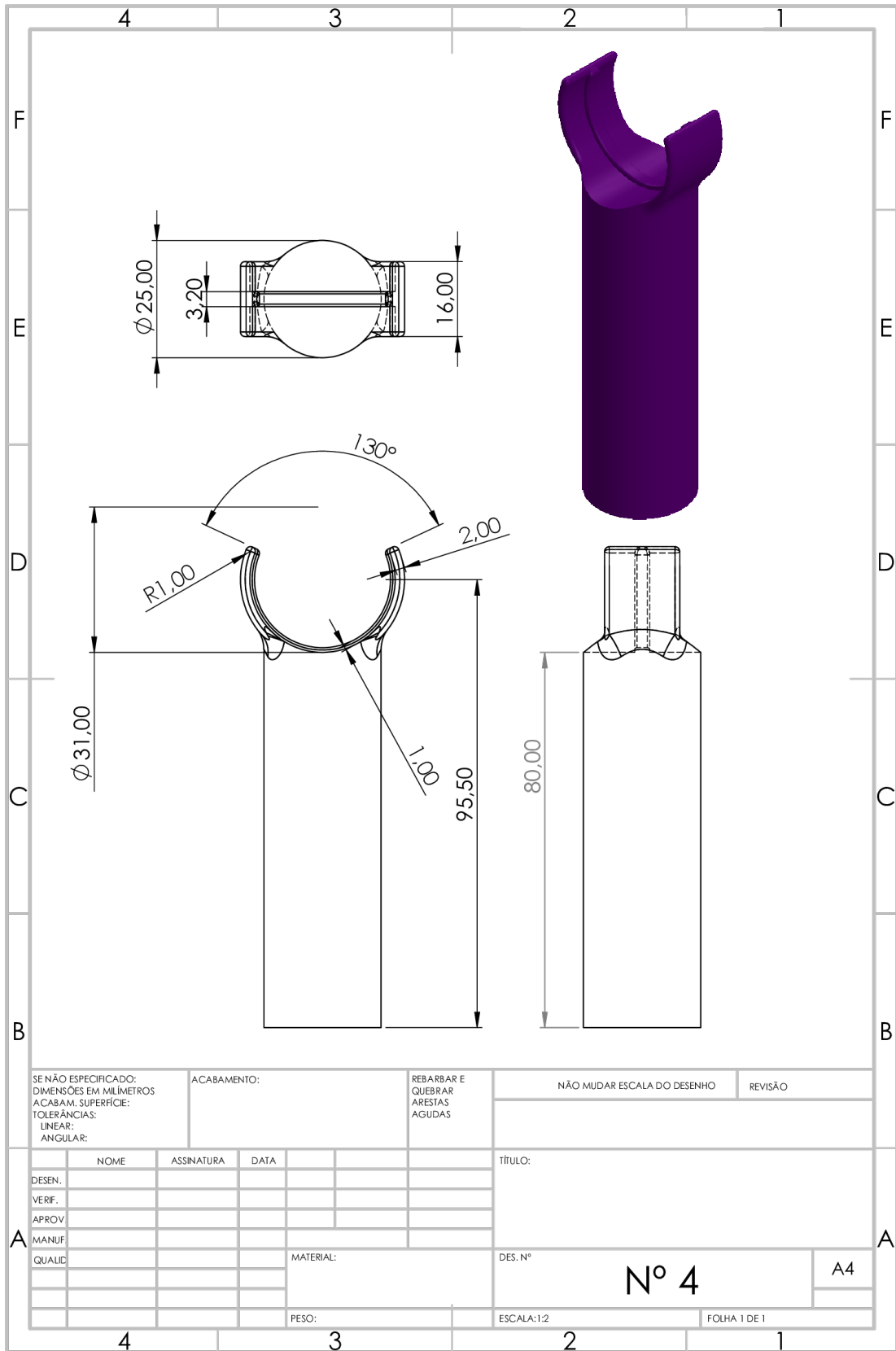


Figura A.107 – Desenho Técnico do Suporte do Microscópio

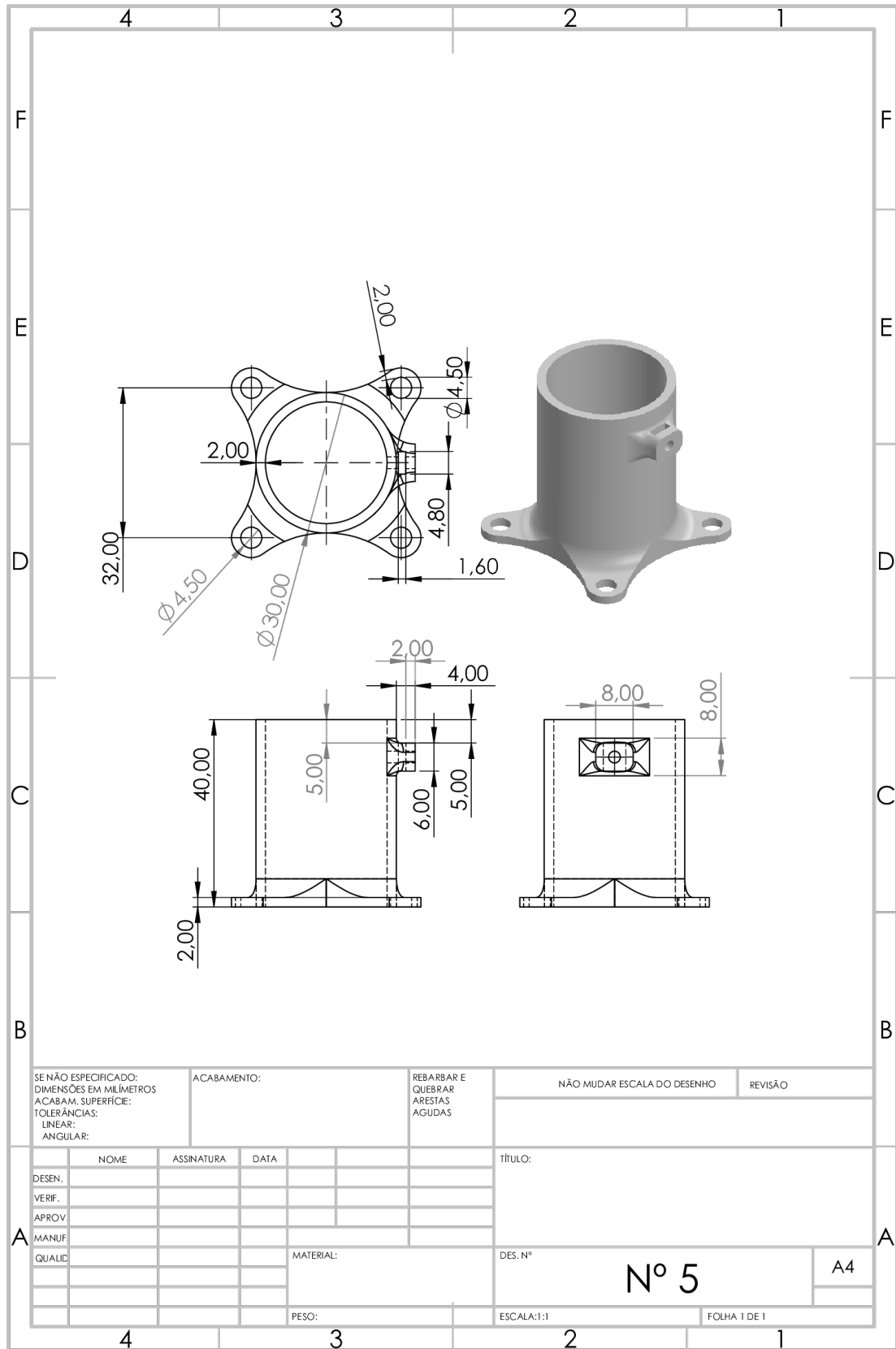


Figura A.108 – Desenho Técnico da Base do Microscópio e da Iluminação Lateral

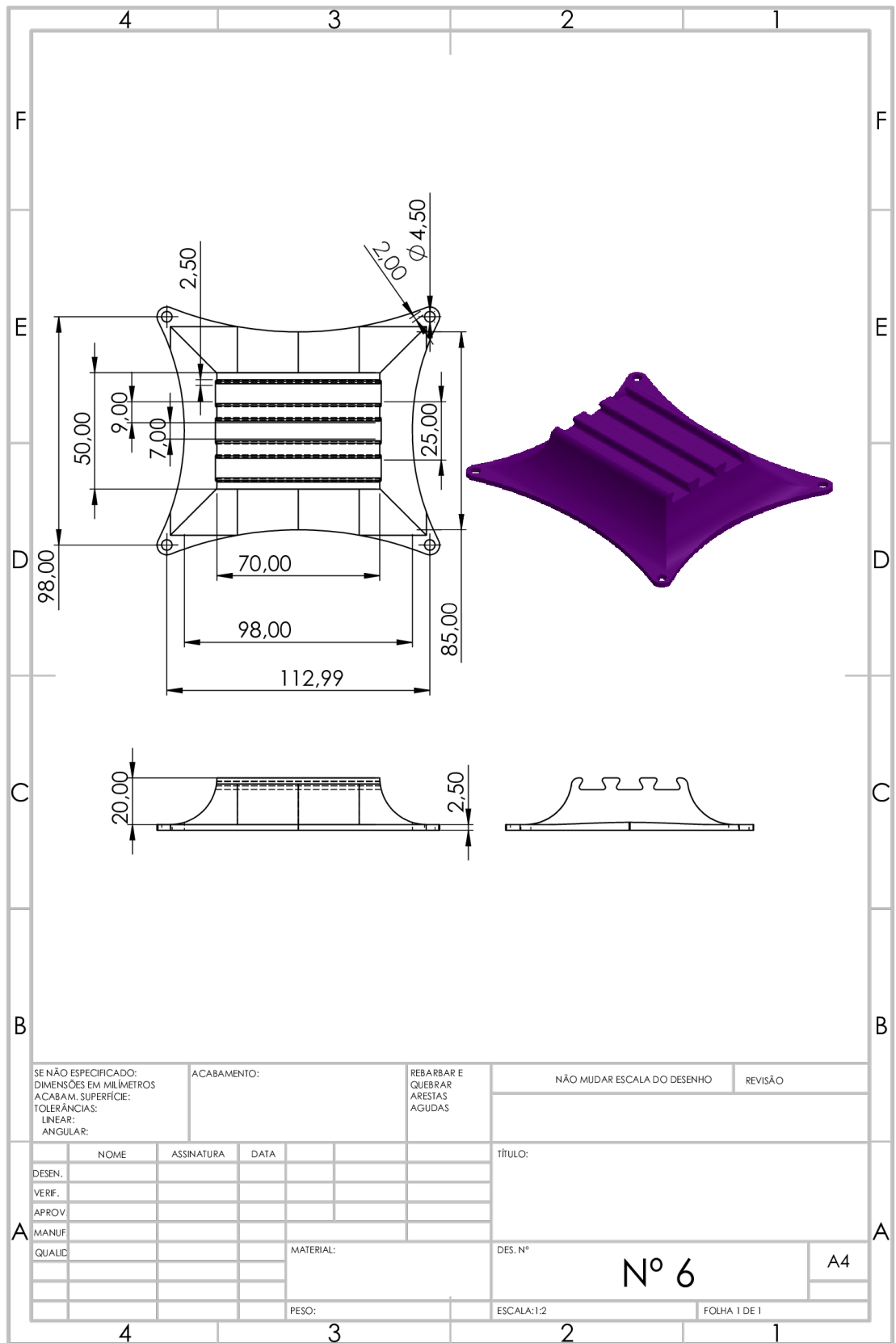


Figura A.109 – Desenho Técnico do Trilho para o Microscópio

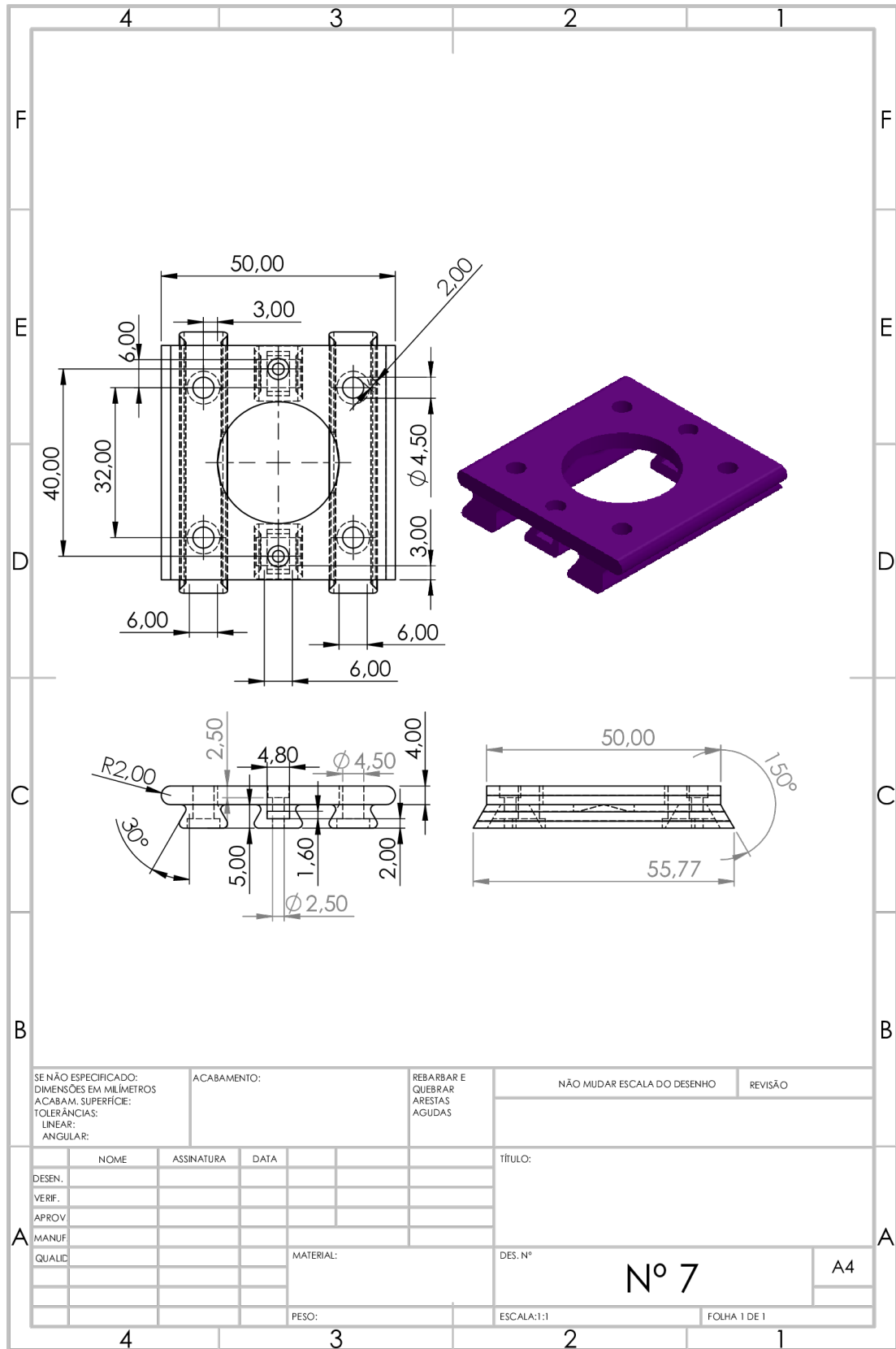


Figura A.110 – Desenho Técnico do Encaixe do Trilho do Microscópio

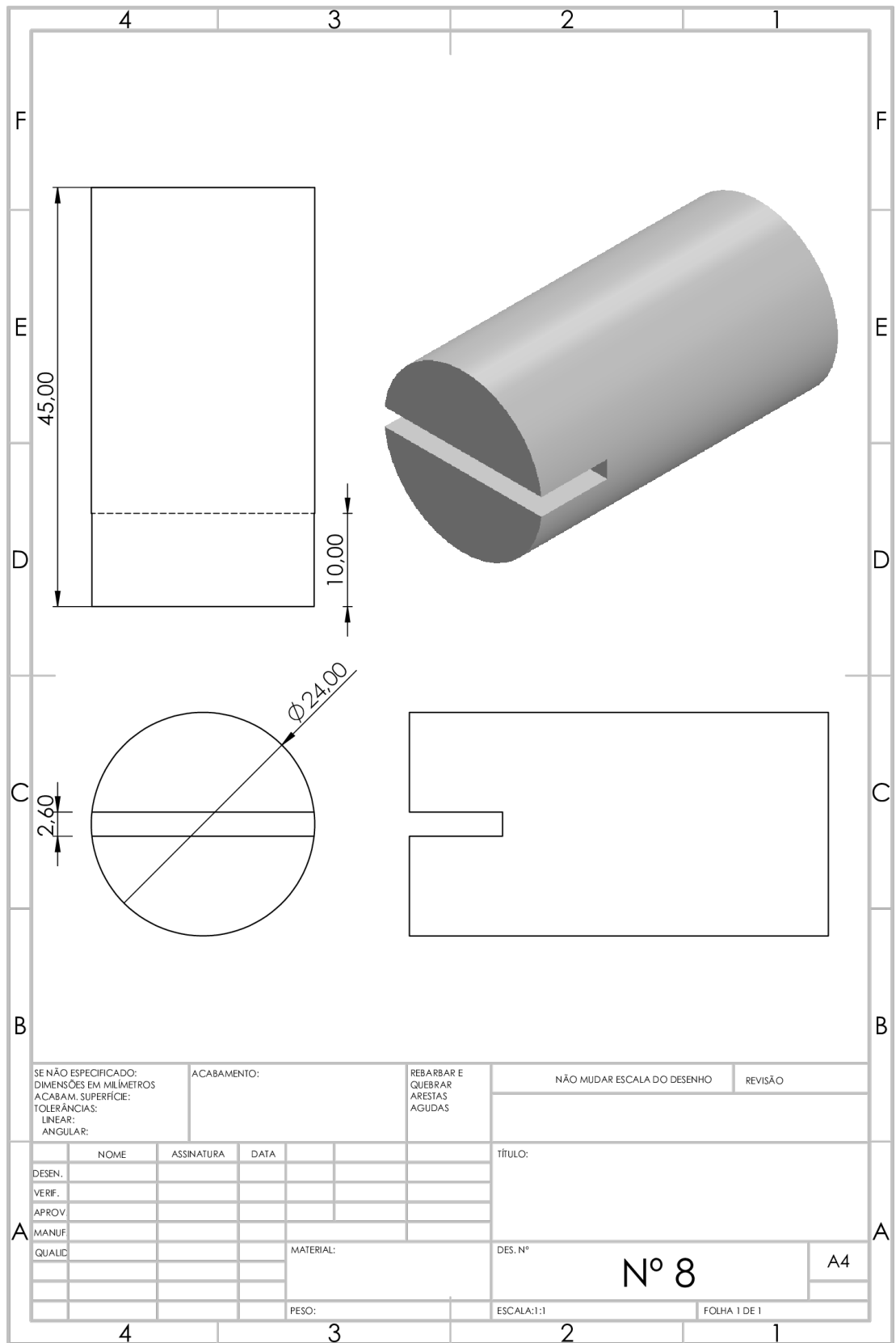


Figura A.111 – Desenho Técnico da Base do Suporte de LEDs Lateral

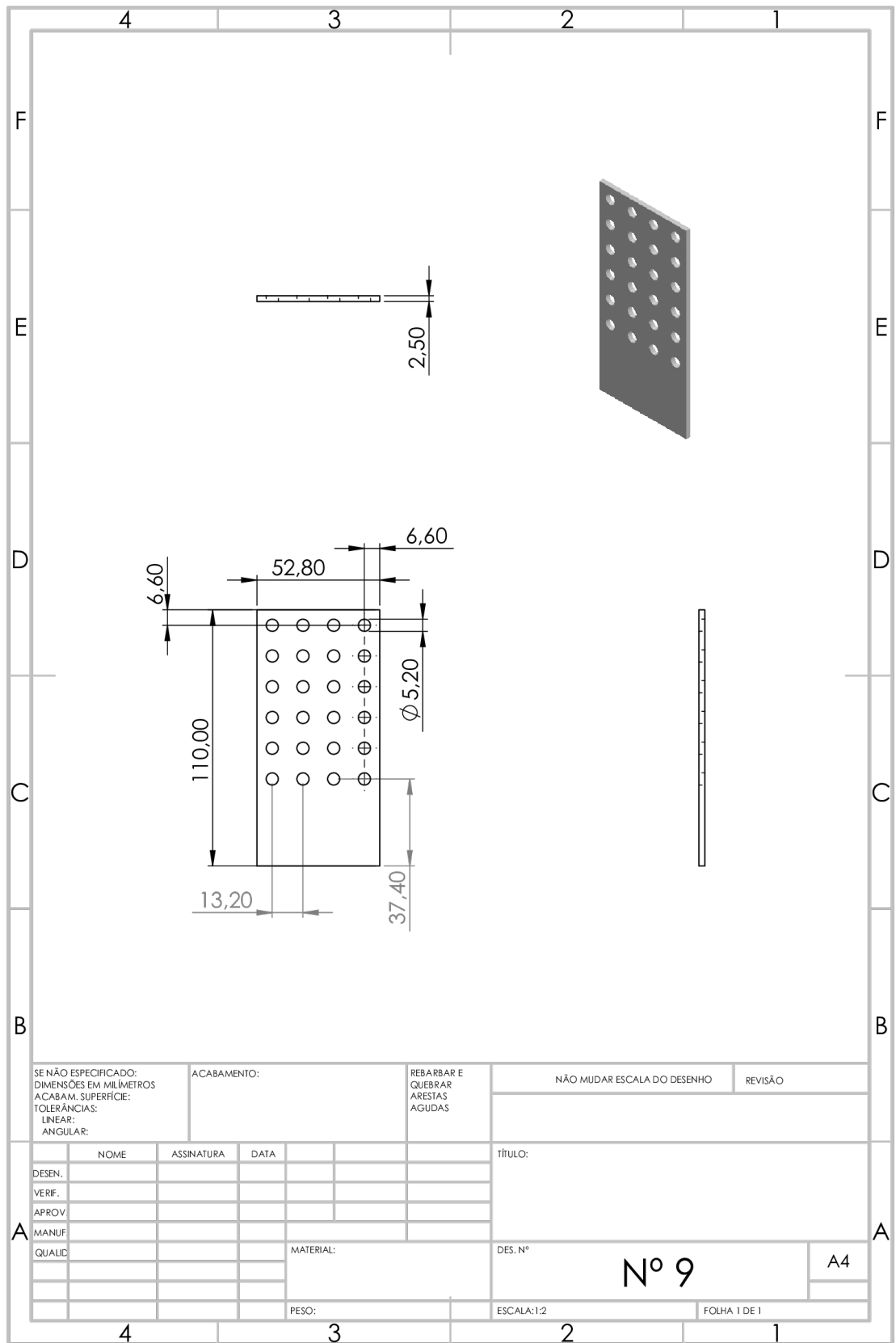


Figura A.112 – Desenho Técnico da Placa de LEDs das Laterais

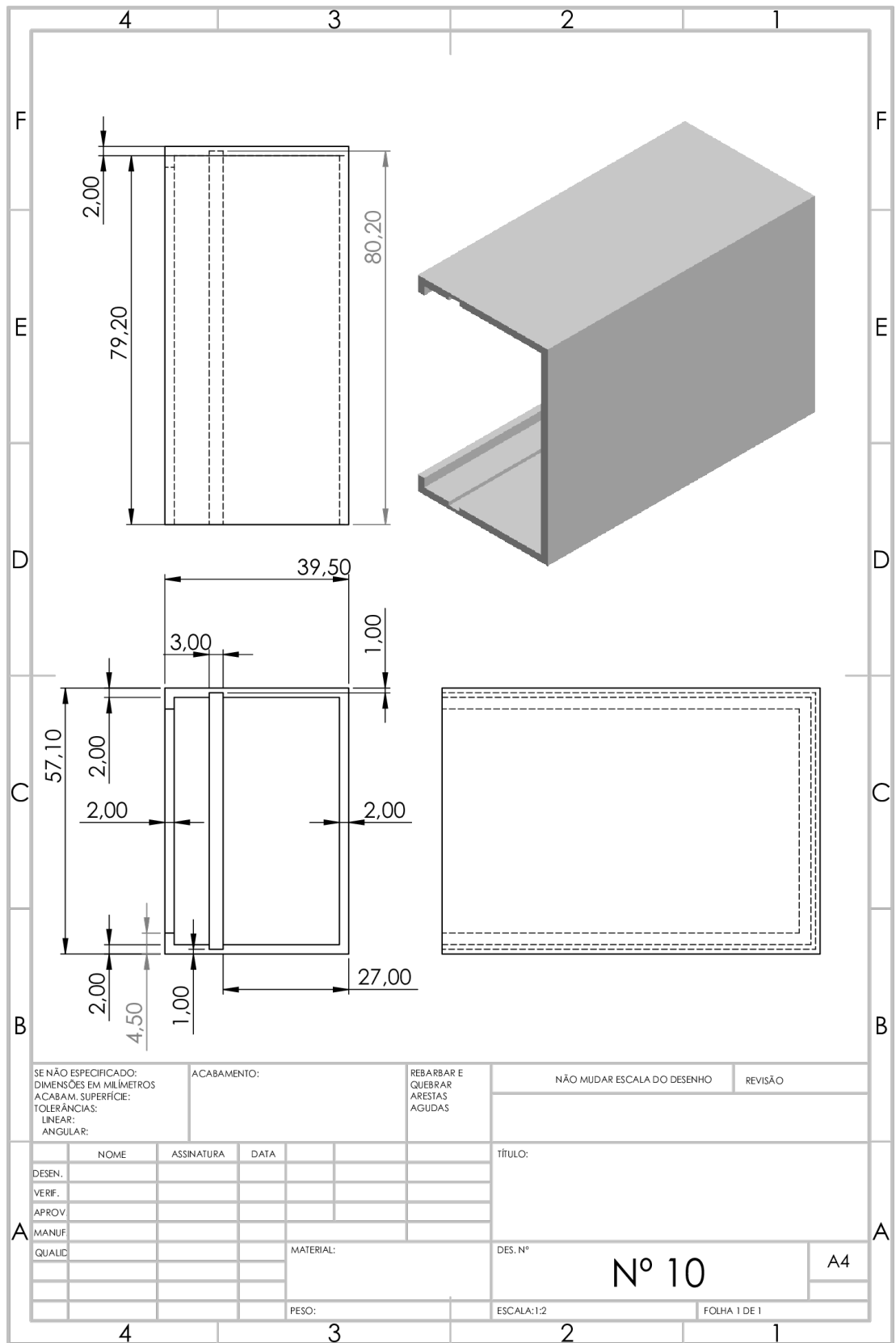


Figura A.113 – Desenho Técnico da Capa e Suporte de Difusor para Placa de LEDs das Laterais

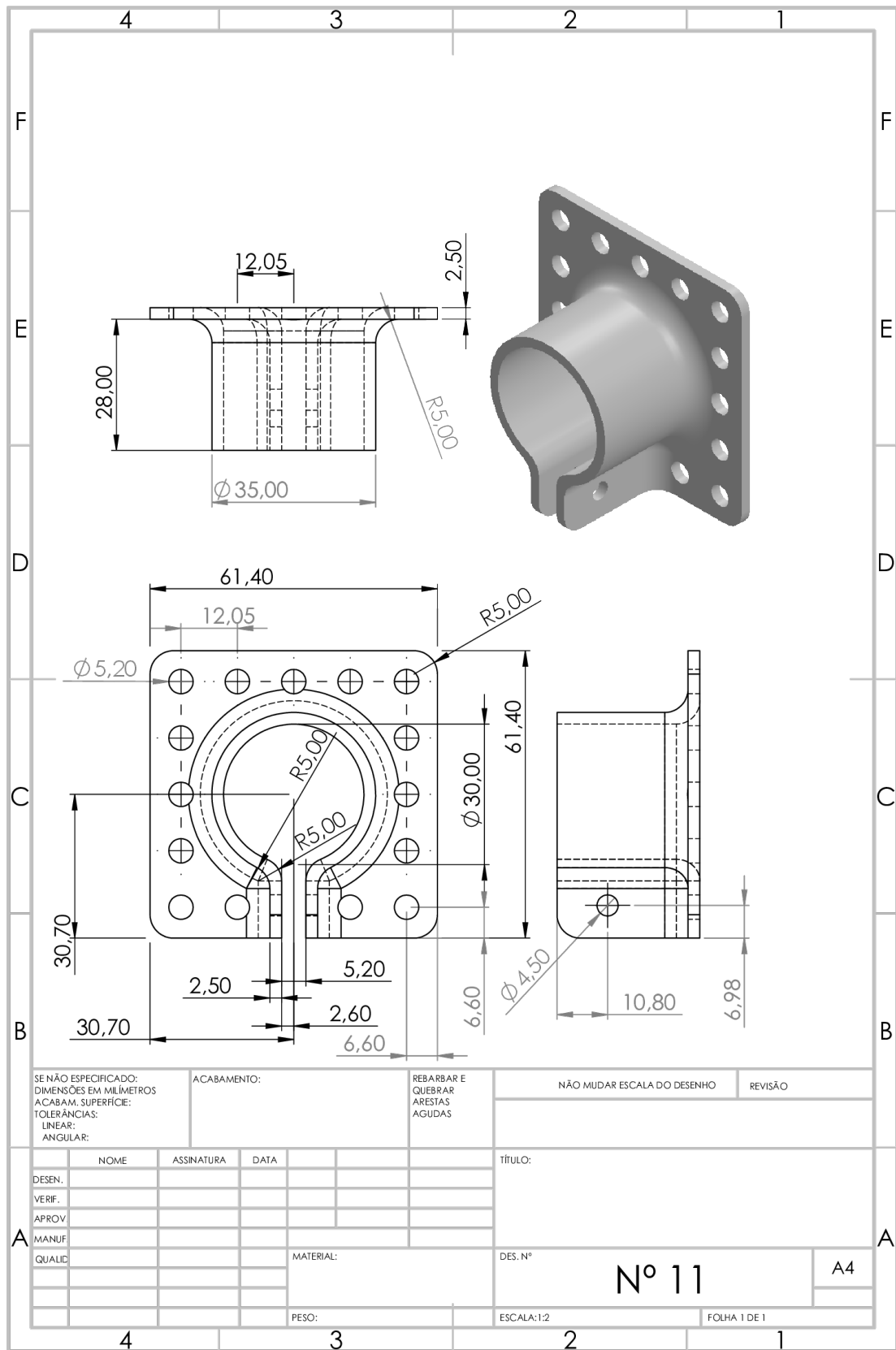


Figura A.114 – Desenho Técnico da Placa de LEDs para o Microscópio

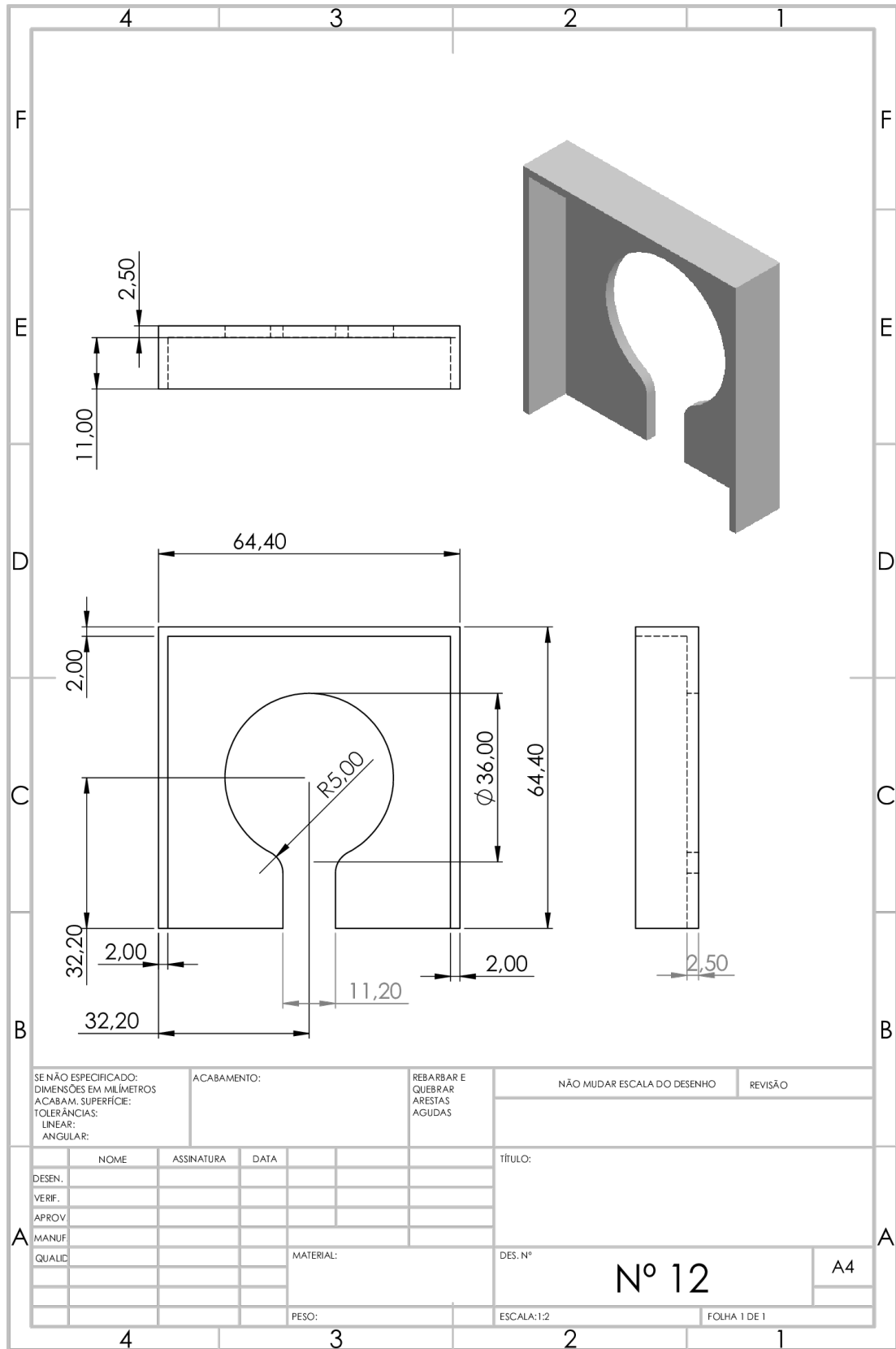


Figura A.115 – Desenho Técnico da Capa Traseira da Placa de LEDs do Microscópio

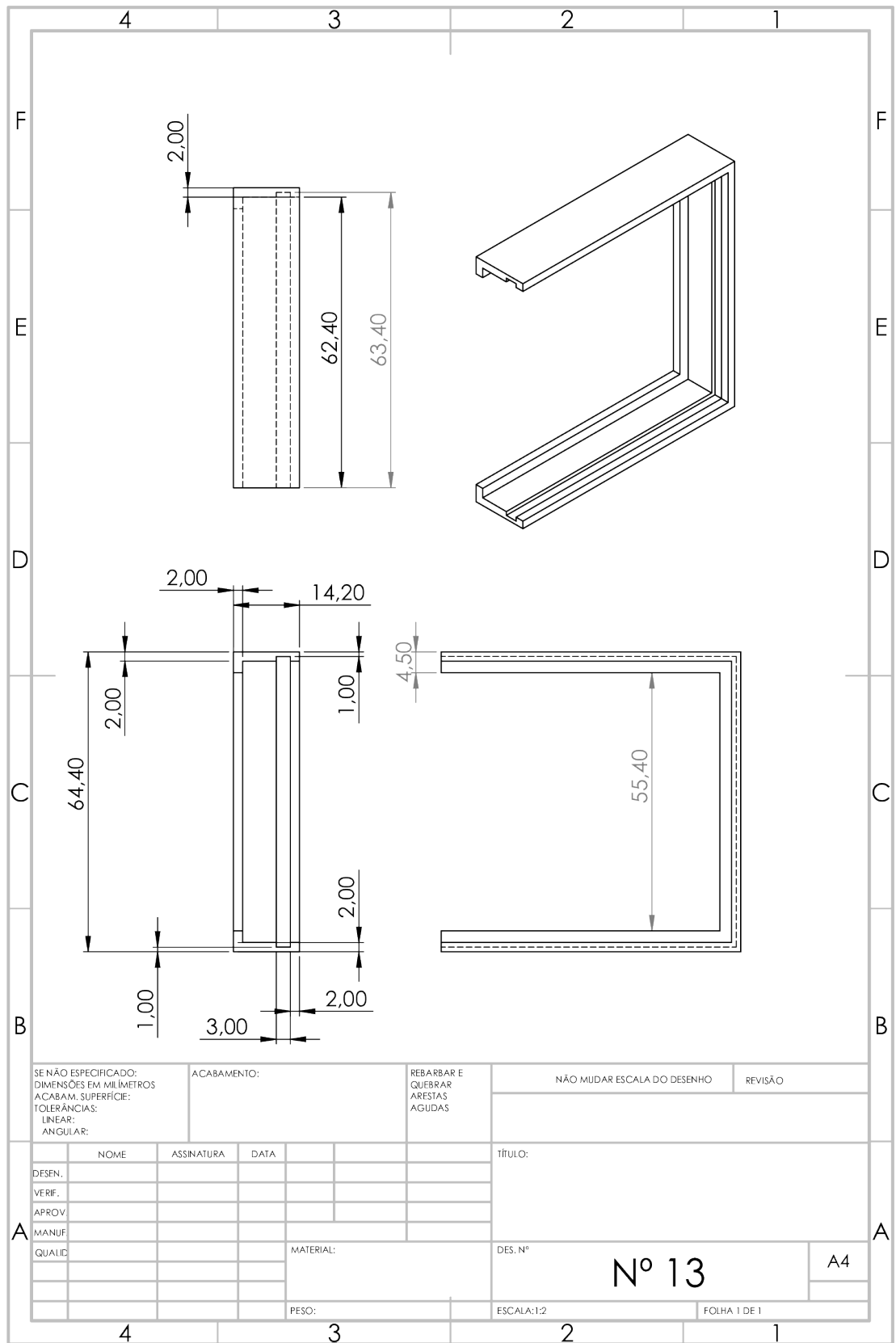


Figura A.116 – Desenho Técnico do Suporte de Difusor para Placa de LEDs do Microscópio

Apêndice B – Códigos de programação

B.1 Código para Movimentação do Motor e Captura das Imagens

Código B.1 – Código de Python

```
1 # Bibliotecas
2 import pyfirmata
3 from pyfirmata import Arduino, util
4 import time
5 import cv2
6
7 # Conecta com o Arduino
8 board = Arduino('COM6')
9 it = util.Iterator(board)
10 it.start()
11
12
13 try:
14     input('Clique no ENTER para iniciar o teste')
15
16     # Configura Portas dos LEDs como PWM
17     board.digital[11].mode = pyfirmata.PWM
18     board.digital[10].mode = pyfirmata.PWM
19     board.digital[9].mode = pyfirmata.PWM
20
21     # Liga os LEDs
22     board.digital[11].write(1)
23     board.digital[10].write(1)
24     board.digital[9].write(1)
25
26
27     # Liga o Microscopio e Inicia a Captação de Frames
28     webcam = cv2.VideoCapture(0)
29     tempo = 2/1000
30     i = 0
31     j = 1
32
33     # Rotaciona o Motor e Sempre que Chegar no Angulo Definido,
34     # Captura o Frame e Salva
35     if webcam.isOpened():
36         validacao, frame = webcam.read()
37         while validacao:
```

```
37     while i<2048:
38         validacao , frame = webcam.read()
39         cv2.imshow("Video da Webcam", frame)
40         key = cv2.waitKey(5)
41
42         board.digital[4].write(0)
43         board.digital[5].write(0)
44         board.digital[6].write(1)
45         board.digital[7].write(1)
46         time.sleep(tempo)
47         if i/j == 11:
48
49             nome = 'Fotos/teste' + str(j) + '.png'
50             cv2.imwrite(nome, frame)
51             j = j + 1
52         i = i + 1
53         board.digital[4].write(0)
54         board.digital[5].write(1)
55         board.digital[6].write(1)
56         board.digital[7].write(0)
57         time.sleep(tempo)
58         if i/j == 11:
59
60             nome = 'Fotos/teste' + str(j) + '.png'
61             cv2.imwrite(nome, frame)
62             j = j + 1
63         i = i + 1
64         board.digital[4].write(1)
65         board.digital[5].write(1)
66         board.digital[6].write(0)
67         board.digital[7].write(0)
68         time.sleep(tempo)
69         if i/j == 11:
70
71             nome = 'Fotos/teste' + str(j) + '.png'
72             cv2.imwrite(nome, frame)
73             j = j + 1
74         i = i + 1
75         board.digital[4].write(1)
76         board.digital[5].write(0)
77         board.digital[6].write(0)
78         board.digital[7].write(1)
79         time.sleep(tempo)
80         if i/j == 11:
81
82             nome = 'Fotos/teste' + str(j) + '.png'
83             cv2.imwrite(nome, frame)
84             j = j + 1
85         i = i + 1
86         print(i)
87     break
88
```

```

89     # Desconecta o Microscopio
90     webcam.release()
91     cv2.destroyAllWindows()
92
93 finally:
94
95     # Desconecta o Arduino
96     board.exit()

```

B.2 Código para Testar os Métodos Descritores

Código B.2 – Código de Python

```

1  # Bibliotecas
2  import cv2
3  import numpy as np
4  from matplotlib import pyplot as plt
5
6  def match_features(detector, img1, img2, img3, img4):
7      # Detecta keypoints e descritores nas duas imagens
8      keypoints1, descritores1 = detector.detectAndCompute(img1,
9      None)
10     keypoints2, descritores2 = detector.detectAndCompute(img2,
11     None)
12     print(len(keypoints1))
13     print(len(keypoints2))
14     # Encontra as combinações
15     bf = cv2.BFMatcher()
16
17     # Encontra as melhores combinações usando os descritores
18     matches = bf.knnMatch(descritores1, descritores2, k=2)
19
20     # Aplica o teste de razão de Lowe para selecionar os melhores
21     matches
22     good_matches = []
23     great_matches = []
24     for m, n in matches:
25         if m.distance < 0.75 * n.distance:
26             good_matches.append(m)
27         if m.distance <= 125:
28             great_matches.append(m)
29
30     # Desenha as combinações na imagem
31     img_matches = cv2.drawMatches(img3, keypoints1, img4,
32     keypoints2, good_matches, None, flags=2)
33
34     return img_matches
35
36 # Carrega as duas imagens
37 img1 = cv2.imread("Fotos_teste/R_a6.png", cv2.IMREAD_GRAYSCALE)

```

```
34 img1c = cv2.imread("Fotos_teste/R_a6.png")
35 img2 = cv2.imread("Fotos_teste/R_a1.png", cv2.IMREAD_GRAYSCALE)
36 img2c = cv2.imread("Fotos_teste/R_a1.png")
37
38 # Carrega as outras duas imagens
39 img3 = cv2.imread("Fotos_teste/R_a5.png", cv2.IMREAD_GRAYSCALE)
40 img3c = cv2.imread("Fotos_teste/R_a5.png")
41 img4 = cv2.imread("Fotos_teste/R_u6.png", cv2.IMREAD_GRAYSCALE)
42 img4c = cv2.imread("Fotos_teste/R_u6.png")
43
44 # Cria os objetos dos detectores SURF, SIFT, ORB e BRISK
45 surf = cv2.xfeatures2d.SURF_create()
46 sift = cv2.SIFT_create()
47 orb = cv2.ORB_create()
48 brisk = cv2.BRISK_create()
49
50 # Chama a função que vai descobrir as combinações de keypoints e
    desenha-las nas imagens
51 print("Entrando no Surf:")
52 img_surf = match_features(surf, img1, img2, img1c, img2c)
53 print("Entrando no Sift:")
54 img_sift = match_features(sift, img1, img2, img1c, img2c)
55 print("Entrando no Orb:")
56 img_orb = match_features(orb, img1, img2, img1c, img2c)
57 print("Entrando no Brisk:")
58 img_brisk = match_features(brisk, img1, img2, img1c, img2c)
59
60 # Mostra as imagens com as combinações
61 plt.figure(figsize=(12, 8))
62 plt.subplot(221), plt.imshow(img_surf), plt.title('SURF')
63 plt.subplot(222), plt.imshow(img_sift), plt.title('SIFT')
64 plt.subplot(223), plt.imshow(img_orb), plt.title('ORB')
65 plt.subplot(224), plt.imshow(img_brisk), plt.title('BRISK')
66 plt.show()
67
68 cv2.imshow("Image1", img_surf)
69 cv2.imshow("Image2", img_sift)
70 cv2.imshow("Image3", img_orb)
71 cv2.imshow("Image4", img_brisk)
72 cv2.waitKey(0)
73 cv2.destroyAllWindows()
74 # cv2.imwrite('imgs_resultado/matches_surf.png', img_surf)
75 # cv2.imwrite('imgs_resultado/matches_sift.png', img_sift)
76 # cv2.imwrite('imgs_resultado/matches_orb.png', img_orb)
77 # cv2.imwrite('imgs_resultado/matches_brisk.png', img_brisk)
```


B.3 Código para União das Imagens

Código B.3 – Código de Python

```

1 # Bibliotecas
2 import cv2
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from PIL import Image
6 import math
7 import glob
8 import copy
9
10 # Resolução das imagens
11 Width = 640
12 Height = 480
13
14 # Função que Calcula os 4 melhores pontos para a homografia
    (cantos)
15 def quatro_pontas(pontos):
16     # Ordena os pontos pelo eixo x crescente
17     p_xc = sorted(pontos, key=lambda ponto: ponto[0])
18     #print("Ponto Ord X: " + str(pontos_ordenados_xc))
19
20     # Ordena os pontos pelo eixo x decrescente
21     p_xd = sorted(pontos, key=lambda ponto: ponto[0], reverse=True)
22     #print("Ponto Ord X: " + str(pontos_ordenados_xd))
23
24     # Ordena os pontos pelo eixo y crescente
25     p_yc = sorted(pontos, key=lambda ponto: ponto[1])
26     #print("Ponto Ord Y: " + str(pontos_ordenados_yc))
27
28     # Ordena os pontos pelo eixo y decrescente
29     p_yd = sorted(pontos, key=lambda ponto: ponto[1], reverse=True)
30     #print("Ponto Ord Y: " + str(pontos_ordenados_yd))
31
32     tam = len(pontos)
33
34     # Ponto 00
35     pts00 = copy.deepcopy(pontos)
36     for x in range(tam):
37         for y in range(tam):
38             if p_xc[x] == p_yc[y]:
39                 pts00[x] = p_xc[x] + (x+y, 0)
40     #print("PX00: " + str(pts00))
41     pts00 = sorted(pts00, key=lambda pts00: pts00[2])
42     #print("PX00 Ord: " + str(pts00))
43     ponto00 = pts00[0][0:2]
44
45     # Ponto 01
46     pts01 = copy.deepcopy(pontos)
47     for x in range(tam):

```

```
48     for y in range(tam):
49         if p_xc[x] == p_yd[y]:
50             pts01[x] = p_xc[x] + (x+y, 0)
51     #print("PX01: " + str(pts01))
52     pts01 = sorted(pts01, key=lambda pts01: pts01[2])
53     #print("PX01 Ord: " + str(pts01))
54     ponto01 = pts01[0][0:2]
55
56     # Ponto 10
57     pts10 = copy.deepcopy(pontos)
58     for x in range(tam):
59         for y in range(tam):
60             if p_xd[x] == p_yc[y]:
61                 pts10[x] = p_xd[x] + (x+y, 0)
62     #print("PX10: " + str(pts10))
63     pts10 = sorted(pts10, key=lambda pts10: pts10[2])
64     #print("PX10 Ord: " + str(pts10))
65     ponto10 = pts10[0][0:2]
66
67     # Ponto 11
68     pts11 = copy.deepcopy(pontos)
69     for x in range(tam):
70         for y in range(tam):
71             if p_xd[x] == p_yd[y]:
72                 pts11[x] = p_xd[x] + (x+y, 0)
73     #print("PX11: " + str(pts11))
74     pts11 = sorted(pts11, key=lambda pts11: pts11[2])
75     #print("PX11 Ord: " + str(pts11))
76     ponto11 = pts11[0][0:2]
77
78     # Retorna os quatro pontos selecionados
79     return [ponto00, ponto01, ponto10, ponto11]
80
81 # Le as imagens capturadas e salva os nomes dos arquivos
82 nomesImgs = glob.glob("158Imagens/R*")
83 #nomesImgs.sort()
84
85 imagens0 = []
86 imagens = []
87
88 # Abre as imagens capturadas, corta as distorções e salva as
89     # imagens em uma pasta nova
90     for nome in nomesImgs:
91         #img = Image.open(nome)
92         img0 = cv2.imread(nome)
93         img0 = cv2.cvtColor(img0, cv2.COLOR_BGR2RGB)
94         #Pega dimensões da imagem
95         alt = img0.shape[0]
96         larg = img0.shape[1]
97         img1 = img0[0:alt,200:600]
98         name = ("158Imagens/cortadas/" + nome[10:])
99         cv2.imwrite(name, img1)
```

```
99     print(name)
100     imagens.append(img1)
101
102 num_imagens = len(imagens)
103
104 # Mostra as imagens em Grade (teste para ver se esta pegando as
105     imagens corretas)
106 #plt.figure(figsize=[30, 10])
107 #num_cols = 3
108 #num_rows = math.ceil(num_imagens / num_cols)
109 #for i in range(0, num_imagens):
110     #plt.subplot(num_rows, num_cols, i+1)
111     #plt.axis('off')
112     #plt.imshow(imagens[i])
113
114 # Loop que encontra os keypoints, realiza a homografia, encontra
115     os novos keypoints e uni as imagens
116 for n in range(num_imagens):
117     if n < num_imagens - 1:
118
119         # Detectando os keypoints com SIFT para usar na homografia
120         sift = cv2.SIFT_create()
121         kp1, des1 = sift.detectAndCompute(imagens[n], None)
122         kp2, des2 = sift.detectAndCompute(imagens[n+1], None)
123         #cv2.imshow("Img 1", imagens[n])
124         #cv2.imshow("Img 2", imagens[n+1])
125         cv2.waitKey(0)
126         cv2.destroyAllWindows()
127
128         # Encontrando as combinações entre keypoints
129         bf = cv2.BFMatcher()
130         matches = bf.match(des1, des2)
131
132         # Filtrando os melhores pontos
133         bons = []
134         otimos = []
135         for m in matches:
136             #print(m)
137             bons.append(m)
138             if m.distance <= 100.0 and
139                 kp1[m.queryIdx].pt[0] > (imagens[n].shape[1] - 400):
140                 otimos.append(m)
141         print(len(otimos))
142
143         points = []
144         for contador in range(len(otimos)):
145             (points).append(kp1[otimos[contador].queryIdx].pt)
146
147         # Chamando função quatro_pontos que encontra os 4 cantos
148             dos pontos das combinações
149         Qpontos = quatro_pontos(points)
150         #print("Os quatro pontos 1 sao: "+str(Qpontos))
```

```

147
148     # Descubre os pontos da segunda imagem equivalentes aos
149         cantos encontrados na primeira imagem
150     posicoes = []
151     posicao = 0
152     parada = 0
153     while(True):
154         #if kp1[otimos[contador].queryIdx].pt ==
155             Qpontos[posicao]:
156         if points[contador] == Qpontos[posicao]:
157             posicoes.append(contador)
158             posicao = posicao + 1
159             contador = contador + 1
160         if contador == len(otimos):
161             contador = 0
162         if posicao == 4:
163             break
164
165     Qpontos2 = []
166     Qpontos2.append(kp2[otimos[posicoes[0]].trainIdx].pt)
167     Qpontos2.append(kp2[otimos[posicoes[1]].trainIdx].pt)
168     Qpontos2.append(kp2[otimos[posicoes[2]].trainIdx].pt)
169     Qpontos2.append(kp2[otimos[posicoes[3]].trainIdx].pt)
170
171     QP1npy = np.array(Qpontos)
172     QP2npy = np.array(Qpontos2)
173     #print("Os quatro pontos 2 sao: "+str(QP2npy))
174
175     # Realizacao de Homografia da imagem
176     # Quatro pontos na imagem que vai ser modificada
177     pts_src = QP2npy
178
179     # Quatro pontos na imagem de referencia
180     pts_dst = QP1npy
181
182     # Calcula a Homografia
183     h, status = cv2.findHomography(pts_src, pts_dst,
184         cv2.RANSAC, 5.0)
185
186     # Modifica a imagem para a posição de referencia
187     im_out = cv2.warpPerspective(imagens[n+1], h,
188         (2*((imagens[n]).shape[1]),(imagens[n]).shape[0]))
189
190     # Corta partes desnecessarias da imagem modificada
191     zeros = np.array([0,0,0])
192     for cont1 in range(im_out.shape[1]):
193         for cont in range(im_out.shape[0]):
194             if (im_out[cont][cont1]).all() != zeros.all():
195                 break
196             if (im_out[cont][cont1]).all() != zeros.all():
197                 break

```

```

195     for cont2 in range(im_out.shape[1]):
196         for cont in range(im_out.shape[0]):
197             if (im_out[im_out.shape[0] -
198                     (cont+1)][im_out.shape[1]-(cont2+1)].all() !=
199                     zeros.all()):
200                 break
201             if (im_out[im_out.shape[0] -
202                     (cont+1)][im_out.shape[1]-(cont2+1)].all() !=
203                     zeros.all()):
204                 break
205         im_out = im_out[0:im_out.shape[0],
206                     cont1:im_out.shape[1]-(cont2+1)]
207
208     # Mostra o resultado depois da homografia
209     #cv2.imshow("Imagem torta", imagens[n+1])
210     #cv2.imshow("Imagem de referencia", imagens[n])
211     #cv2.imshow("Imagem cortada", im_out)
212     #cv2.waitKey(0)
213
214     img4 =
215         cv2.drawMatches(imagens[n], kp1, imagens[n+1], kp2, otimos,
216                       None, flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
217
218     # Detectando os keypoints com SIFT novamente, agora para
219     # unir as imagens
220     sift = cv2.SIFT_create()
221     kp1, des1 = sift.detectAndCompute(imagens[n], None)
222     kp2, des2 = sift.detectAndCompute(im_out, None)
223
224     # Encontrando as combinações entre keypoints
225     bf = cv2.BFMatcher()
226     matches = bf.match(des1, des2)
227
228     # Filtrando os melhores pontos
229     bons = []
230     for m in matches:
231         #print(m)
232         bons.append(m)
233     bons = sorted(bons, key = lambda x:x.distance)
234
235     # Selecionando os pontos onde as duas imagens serao
236     # cortadas
237     #print('pt1: {}, pt2:
238           {}'.format(kp1[matches[0].queryIdx].pt[0],
239                     kp2[matches[0].trainIdx].pt))
240     if n > 3*num_imagens/4:
241         print('a')
242         cont = 0
243         pt11 = kp1[bons[cont].queryIdx].pt
244         while int(pt11[0]) <= 400:
245             cont = cont + 1
246             pt11 = kp1[bons[cont].queryIdx].pt

```

```

236
237     pt12 = kp1[bons[cont+1].queryIdx].pt
238     pt21 = kp2[bons[cont].trainIdx].pt
239     pt22 = kp2[bons[cont+1].trainIdx].pt
240 else:
241     print('b')
242     pt11 = kp1[bons[0].queryIdx].pt
243     pt12 = kp1[bons[1].queryIdx].pt
244     pt21 = kp2[bons[0].trainIdx].pt
245     pt22 = kp2[bons[1].trainIdx].pt
246
247     #pt11 = (1100,425)
248     #pt12 = (1100,131)
249     #pt21 = (728,413)
250     #pt22 = (728,134)
251
252     # Visualiza as combinações encontradas entre as duas
253     # imagens
254     img3 = cv2.drawMatches(imagens[n],kp1,im_out,kp2,bons,
255     None,flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
256     #cv2.imshow("Matching result 1", img4)
257     #cv2.imshow("Matching result 2", img3)
258     # cv2.waitKey(0)
259     # cv2.destroyAllWindows()
260
261     # Cortando e juntando as imagens com base nos pontos
262     # definidos
263     corte1 = imagens[n][0:Height,0:int(pt11[0])]
264     corte2 = imagens[n+1][0:Height,int(pt21[0]):Width]
265     imgF = cv2.hconcat([corte1,corte2])
266     imagens[n+1] = imgF
267     nome = (str(158)+"Imagens/montando/R_" + str((n+1)) +
268     ".png")
269     cv2.imwrite(nome,imgF)
270     plt.imshow(imgF)
271     plt.show()
272
273     # Esse else e utilizado para mexer no final da uniao das
274     # imagens, quando for a ultima imagem o codigo vai unir e
275     # cortar o excesso para nao repetir nenhuma parte do projetil
276 else:
277
278     # Detectando os keypoints com SIFT para usar na homografia
279     sift = cv2.SIFT_create()
280     kp1, des1 = sift.detectAndCompute(imagens[n], None)
281     kp2, des2 = sift.detectAndCompute(imagens[n+1], None)
282     #cv2.imshow("Img 1", imagens[n])
283     #cv2.imshow("Img 2", imagens[n+1])
284     cv2.waitKey(0)
285     cv2.destroyAllWindows()
286
287     # Encontrando as combinações entre keypoints

```

```

282     bf = cv2.BFMatcher()
283     matches = bf.match(des1, des2)
284
285     # Filtrando os melhores pontos
286     bons = []
287     otimos = []
288     for m in matches:
289         #print(m)
290         bons.append(m)
291         if m.distance <= 100.0 and
292             kp1[m.queryIdx].pt[0] > (imagens[n].shape[1]-400):
293             otimos.append(m)
294     print(len(otimos))
295
296     points = []
297     for contador in range(len(otimos)):
298         (points).append(kp1[otimos[contador].queryIdx].pt)
299
300     # Chamando funcao quatro_pontos que encontra os 4 cantos
301     # dos pontos das combinações
302     Qpontos = quatro_pontos(points)
303     #print("Os quatro pontos 1 sao: "+str(Qpontos))
304
305     # Descobre os pontos da segunda imagem equivalentes aos
306     # cantos encontrados na primeira imagem
307     posicoes = []
308     posicao = 0
309     parada = 0
310     while(True):
311         #if kp1[otimos[contador].queryIdx].pt ==
312         #    Qpontos[posicao]:
313         if points[contador] == Qpontos[posicao]:
314             posicoes.append(contador)
315             posicao = posicao + 1
316             contador = contador + 1
317         if contador == len(otimos):
318             contador = 0
319         if posicao == 4:
320             break
321
322     Qpontos2 = []
323     Qpontos2.append(kp2[otimos[posicoes[0]].trainIdx].pt)
324     Qpontos2.append(kp2[otimos[posicoes[1]].trainIdx].pt)
325     Qpontos2.append(kp2[otimos[posicoes[2]].trainIdx].pt)
326     Qpontos2.append(kp2[otimos[posicoes[3]].trainIdx].pt)
327
328     QP1npy = np.array(Qpontos)
329     QP2npy = np.array(Qpontos2)
330     #print("Os quatro pontos 2 sao: "+str(QP2npy))
331
332     # Realizacao de Homografia da imagem

```

```

330         # Quatro pontos na imagem que vai ser modificada
331     pts_src = QP2npy
332
333         # Quatro pontos na imagem de referencia
334     pts_dst = QP1npy
335
336         # Calcula a Homografia
337     h, status = cv2.findHomography(pts_src, pts_dst,
338                                     cv2.RANSAC, 5.0)
339
340         # Modifica a imagem para a posição de referencia
341     im_out = cv2.warpPerspective(imagens[n+1], h,
342                                   (2*((imagens[n]).shape[1]),(imagens[n]).shape[0]))
343
344         # Corta partes desnecessarias da imagem modificada
345     zeros = np.array([0,0,0])
346     for cont1 in range(im_out.shape[1]):
347         for cont in range(im_out.shape[0]):
348             if (im_out[cont][cont1]).all() != zeros.all():
349                 break
350             if (im_out[cont][cont1]).all() != zeros.all():
351                 break
352         for cont2 in range(im_out.shape[1]):
353             for cont in range(im_out.shape[0]):
354                 if (im_out[im_out.shape[0] -
355                               (cont+1)][im_out.shape[1]-(cont2+1)]).all() !=
356                     zeros.all():
357                     break
358                 if (im_out[im_out.shape[0] -
359                               (cont+1)][im_out.shape[1]-(cont2+1)]).all() !=
360                     zeros.all():
361                     break
362         im_out = im_out[0:im_out.shape[0],
363                           cont1:im_out.shape[1]-(cont2+1)]
364
365         # Mostra o resultado depois da homografia
366     #cv2.imshow("Imagem torta", imagens[n+1])
367     #cv2.imshow("Imagem de referencia", imagens[n])
368     #cv2.imshow("Imagem cortada", im_out)
369     #cv2.waitKey(0)
370
371     img4 =
372         cv2.drawMatches(imagens[n],kp1,imagens[n+1],kp2,otimos,
373                         None,flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
374
375         # Detectando os keypoints com SIFT novamente, agora para
376         unir as imagens
377     sift = cv2.SIFT_create()
378     kp1, des1 = sift.detectAndCompute(imagens[n], None)
379     kp2, des2 = sift.detectAndCompute(im_out, None)
380
381         # Encontrando as combinações entre keypoints

```



```
372     bf = cv2.BFMatcher()
373     matches = bf.match(des1, des2)
374
375     # Filtrando os melhores pontos
376     bons = []
377     for m in matches:
378         #print(m)
379         bons.append(m)
380     bons = sorted(bons, key = lambda x:x.distance)
381
382     # Selecionando os pontos onde as duas imagens serao
383     # cortadas
384     #print('pt1: {}, pt2:
385     #       {}'.format(kp1[matches[0].queryIdx].pt[0],
386     #                   kp2[matches[0].trainIdx].pt))
387     cont = 0
388     pt11 = kp1[bons[cont].queryIdx].pt
389     while int(pt11[0]) <= 400:
390         cont = cont + 1
391         pt11 = kp1[bons[cont].queryIdx].pt
392
393     pt12 = kp1[bons[cont+1].queryIdx].pt
394     pt21 = kp2[bons[cont].trainIdx].pt
395     pt22 = kp2[bons[cont+1].trainIdx].pt
396
397     #pt11 = (1100,425)
398     #pt12 = (1100,131)
399     #pt21 = (728,413)
400     #pt22 = (728,134)
401
402     # Cortando e juntando as imagens com base nos pontos
403     # definidos, cortando o final da imagem para nao repetir
404     # nada
405     imgF = imagens[n][0:Height,0:int(pt11[0])]
406     nome = (str(158)+"Imagens/montando/R_" + str((n+1)) +
407            ".png")
408     cv2.imwrite(nome, imgF)
409     print("c")
410
411     plt.imshow(imgF)
412     plt.show()
```

B.4 Código para Extração de Sinal das Imagens Montadas

Código B.4 – Código de Python

```
1 # Bibliotecas
2 import cv2
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import pandas as pd
6 import pickle
7
8 # Abre a imagem que se deseja extrair o sinal
9 n = 2
10 k = 7
11 nome = "Resultados/imgs_projeteis_JB/projetil" + str(n) + "_arma"
12       + str(k) + ".png"
13 img01 = cv2.imread(nome, cv2.IMREAD_GRAYSCALE)
14
15 linha = []
16 sinal = []
17
18 # Loop que percorre a imagem e salva as 10 linhas de onde o sinal
19   sera retirado para fazer a media
20 loop = True
21 i = 90
22 while loop:
23     for j in range(img01.shape[1]):
24         linha.append(img01[i][j])
25         # print("Linha 1 primeiro termo:" + str(linha[0]))
26     sinal.append(linha)
27     linha = []
28     i = i + 1
29     if i == 100:
30         loop = False
31
32 # Loop que percorre as 10 linhas salvas e realiza a media entre os
33   pixels dessas linhas pré-definidas
34 media_sinal = []
35 valor = 0
36 loop = True
37 i = 0
38 rotacao = [i * 0.24340777 for i in range(0, 1480)]
39 while loop:
40     for j in range(10):
41         valor = valor + sinal[j][i]
42     media_sinal.append({"Angulo":rotacao[i],
43                       "Intensidade":int(valor/10)})
44     valor = 0
45     i = i + 1
46     if i == 1480:
```

```
43     loop = False
44
45 # Coloca os valores da média em um dataframe
46 media = pd.DataFrame(media_sinal)
47 print(media)
48 dpi = 100
49
50 # Plota os valores em um gráfico
51 fig, ax = plt.subplots(figsize=(img01.shape[1]/dpi,
52                               img01.shape[0]/dpi), dpi=dpi)
53 fig = sns.lineplot(data=media, x='Angulo', y='Intensidade', color
54                   = 'green', ax=ax )
55 ax.set_xlim(0, 360)
56 ax.set_ylim(0, 255)
57 fig.set_title("Sinal 1 - Projétil Testado")
58
59 # Salva o gráfico na pasta
60 nome2 = "Resultados/Sinais_Graficos/Sinal1_projétil" + str(n) +
61        "_arma" + str(k)
62 nome21 = nome2 + ".png"
63 plt.savefig(nome21, dpi=300)
64 plt.show(fig)
65
66 # Salva o dataframe das médias na pasta
67 nome22 = nome2 + ".pkl"
68 with open(nome22,"wb") as f:
69     pickle.dump(media,f)
70
71 # Daqui para frente refaz o mesmo processo, mudando a posicao das
72 # 10 linhas de pixels utilizadas (utilizado para comparar sinais
73 # de posicoes diferentes em um mesmo projétil)
74 linha = []
75 sinal = []
76
77 # Loop que percorre a imagem e salva as 10 linhas de onde o sinal
78 # sera retirado para fazer a media
79 loop = True
80 i = 290
81 while loop:
82     for j in range(img01.shape[1]):
83         linha.append(img01[i][j])
84         # print("Linha 1 primeiro termo:" + str(linha[0]))
85     sinal.append(linha)
86     linha = []
87     i = i + 1
88     if i == 300:
89         loop = False
90
91 # Loop que percorre as 10 linhas salvas e realiza a media entre os
92 # pixels dessas linhas pre-definidas
93 media_sinal = []
94 valor = 0
```

```
88 loop = True
89 i = 0
90 rotacao = [i * 0.24340777 for i in range(0, 1480)]
91 while loop:
92     for j in range(10):
93         valor = valor + sinal[j][i]
94         media_sinal.append({"Angulo":rotacao[i],
95                             "Intensidade":int(valor/10)})
96     valor = 0
97     i = i + 1
98     if i == 1480:
99         loop = False
100 # Coloca os valores da média em um dataframe
101 media = pd.DataFrame(media_sinal)
102 print(media)
103 dpi = 100
104
105 # Plota os valores em um gráfico
106 fig, ax = plt.subplots(figsize=(img01.shape[1]/dpi,
107                               img01.shape[0]/dpi), dpi=dpi)
108 fig = sns.lineplot(data=media, x='Angulo', y='Intensidade', color
109                    = 'green', ax=ax )
110 ax.set_xlim(0, 360)
111 ax.set_ylim(0, 255)
112 fig.set_title("Sinal 1 - Projétil Testado")
113
114 # Salva o gráfico na pasta
115 nome3 = "Resultados/Sinais_Graficos/Sinal2_projétil" + str(n) +
116        "_arma" + str(k)
117 nome31 = nome3 + ".png"
118 plt.savefig(nome31, dpi=300)
119 plt.show(fig)
120
121 # Salva o dataframe das médias na pasta
122 nome32 = nome3 + ".pkl"
123 with open(nome32,"wb") as f:
124     pickle.dump(media,f)
```

B.5 Código que Realiza a Autocorrelação e Correlação Cruzada

Código B.5 – Código de Python

```

1 # Bibliotecas
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pickle
5 import pandas as pd
6 import seaborn as sns
7 import cv2
8
9 # Abre os dataframes de duas imagens que serão comparadas
10 nomea = "Resultados/Sinais_Graficos/Sinal1_projetil1_arma3.pkl"
11 nomeb = "Resultados/Sinais_Graficos/Sinal1_projetil2_arma4.pkl"
12 with open(nomea,"rb") as grf:
13     grafico1 = pickle.load(grf)
14 with open(nomeb,"rb") as grf2:
15     grafico2 = pickle.load(grf2)
16
17 # União dos dois dataframes em um so
18 del grafico2['Angulo']
19 grafico2.rename(columns={'Intensidade': 'Intensidade2'},
20                 inplace=True)
21 concat = pd.concat([grafico1, grafico2], axis=1)
22
23 # Passa a coluna referente ao sinal do primeiro projétil para um
24 # vetor
25 signal1 = concat['Intensidade'].to_numpy()
26 signal1 = np.array(signal1)
27
28 # Aplica a translação vertical pela mediana
29 signal1 = np.array(signal1) - np.median(signal1)
30
31 # Passa a coluna referente ao sinal do segundo projétil para um
32 # vetor
33 signal2 = concat['Intensidade2'].to_numpy()
34 signal2 = np.array(signal2)
35
36 # Aplica a translação vertical pela mediana
37 signal2 = np.array(signal2) - np.median(signal2)
38
39 # print(signal1)
40
41 # Cria o vetor para o eixo X dos gráficos de correlação cruzada (0
42 # a 360 graus, sendo que a funcao faz o complementar de -360 a 0
43 # graus)
44 fs = len(signal1)
45 time = np.linspace(0, 360, fs)
46 # print(len(time))

```

```
42
43
44 # Defasagem Manual realizada para mostrar que a correlação
    consegue definir quanto de atraso um sinal tem em relação ao
    outro
45 # def circular(vetor, posicoes):
46 #     return np.concatenate((vetor[-posicoes:], vetor[:-posicoes]))
47
48 # signal2 = circular(signal2, -370)
49
50
51 # Plotagem do sinal 1 deslocado pela mediana
52 plt.figure(figsize = (10, 3))
53 plt.plot(time, signal1, linewidth = 1, alpha = 0.7, color =
    'green')
54 plt.grid(linestyle = '--', which='both')
55 plt.ylabel(r'Intensidade')
56 plt.xlim((0, time[-1]))
57 plt.title('Gráfico1 Deslocado pela Mediana')
58 # plt.ylim((-4, 4))
59 plt.xlabel('Angulo')
60 plt.tight_layout()
61
62 # Salva o gráfico na pasta
63 # nome2 = "Resultados/Sinais_Graficos/grafico_deslocado1"
64 # nome21 = nome2 + ".png"
65 # plt.savefig(nome21, dpi=300)
66
67 # Plotagem do sinal 2 deslocado pela mediana
68 plt.figure(figsize = (10, 3))
69 plt.plot(time, signal2, linewidth = 1, alpha = 0.7, color =
    'green')
70 plt.grid(linestyle = '--', which='both')
71 plt.ylabel(r'Intensidade')
72 plt.xlim((0, time[-1]))
73 plt.title('Gráfico2 Deslocado pela Mediana')
74 # plt.ylim((-4, 4))
75 plt.xlabel('Angulo')
76 plt.tight_layout()
77
78 # Salva o gráfico na pasta
79 # nome2 = "Resultados/Sinais_Graficos/grafico_deslocado2"
80 # nome21 = nome2 + ".png"
81 # plt.savefig(nome21, dpi=300)
82
83 # Calcula o coeficiente de correlação para determinar o nível de
    confiança
84 correlation_coefficient = np.corrcoef(signal1, signal2)[0, 1]
85
86 # Número de amostras
87 n = len(signal1)
88
```

```
89 # Graus de liberdade
90 df = n - 2
91
92 # Nível de confiança desejado (99,7\% ou 0,997)
93 confidence_level = 0.997
94
95 # Valor crítico da distribuição t-Student
96 t_critical = np.abs(np.random.standard_t(df, int(1e6))).mean()
97
98 # Erro padrão do coeficiente de correlação
99 standard_error = 1 / np.sqrt(n - 3)
100
101 # Intervalo de confiança
102 lower_bound = correlation_coefficient - t_critical * standard_error
103 upper_bound = correlation_coefficient + t_critical * standard_error
104
105 print("Intervalo de confiança:", lower_bound, "a", upper_bound)
106
107 # Aplica a correlação cruzada entre os 2 sinais
108 signal1 = signal1/np.std(signal1)
109 signal2 = signal2/np.std(signal2)
110 Rxx = np.correlate(signal1, signal2, mode = 'full')
111 tau= np.linspace(-len(Rxx)/8.22, len(Rxx)/8.22, len(Rxx))
112
113 # Plotagem do gráfico de correlação cruzada
114 plt.figure(figsize = (10, 3))
115 plt.plot(tau, Rxx/fs, linewidth = 1, color = 'red', label =
116         'Correlação Cruzada')
117 plt.axhline(y=lower_bound, color='blue', linestyle='--',
118            label='Nível de Confiança')
119 plt.grid(linestyle = '--', which='both')
120 plt.ylabel(r'Correlação Cruzada')
121 # plt.title('FCC Projeteis 1 e 2 da Arma 2')
122 plt.title('FCC Projétil 1 da Arma 3 e Projétil 2 da Arma 4')
123 # plt.title('FCC Grafico Deslocado')
124 plt.legend()
125 plt.xlim((tau[0], tau[-1]))
126 plt.ylim((min(Rxx/fs), 1.2*max(np.abs(Rxx/fs))))
127 plt.xlabel(r'Angulo')
128 plt.tight_layout()
129
130 # Salva o grafico de correlacao cruzada na pasta
131 nome2 = "Resultados/Sinais_Graficos/FCC_projeteis12_arma34"
132 # nome2 = "Resultados/Sinais_Graficos/grafico_deslocado_FAC"
133 nome21 = nome2 + ".png"
134 plt.savefig(nome21, dpi=300)
```

B.6 Código que Realiza a Comparação Estatística

Código B.6 – Código de Python

```

1 # Bibliotecas
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pickle
5 import glob
6 from itertools import combinations
7 from tqdm import tqdm
8 from scipy.stats import gaussian_kde
9
10 # Abre os dataframes dos sinais de todos os projeteis escaneados e
    salva todos em um único vetor
11 dfs = glob.glob("Resultados/Sinais_Graficos/dataFrames/*.pkl")
12 # num = 0
13 # nome = "gfr" + str(num)
14 # nome2 = "df" + str(num)
15 sinais = []
16 for filename in dfs:
17     with open(filename,"rb") as nome:
18         nome2 = pickle.load(nome)
19         # num = num + 1
20         sinal1 = nome2['Intensidade'].to_numpy()
21         sinais.append(sinal1)
22
23 # print(len(sinais))
24
25 # Função que gera combinações 2 a 2 sem repetição
26 def gerar_combinacoes(numeros):
27     return list(combinations(numeros, 2))
28
29 # Lista de números para as combinações
30 numeros = list(range(0,28,1))
31 # print(numeros)
32
33 # Gera a lista de combinações possíveis
34 comb_2_a_2 = gerar_combinacoes(numeros)
35 # print(comb_2_a_2[0][1])
36
37 # Vetores onde serão guardadas as 3 distâncias calculadas para
    todas as combinações 2 a 2 entre os projetéis, sendo vetorI os
    vetores para comparações entre projetéis de armas Iguais e
    vetorD os vetores para comparações entre projetéis de armas
    Diferentes
38 vetorI1 = []
39 vetorD1 = []
40 vetorI2 = []
41 vetorD2 = []
42 vetorI3 = []
43 vetorD3 = []

```



```
44
45 # Calcula as medidas estatísticas para todos os sinais de 2 em 2,
    calculando a distância entre os 2 e salva no vetor mencionado
    antes
46 for i in tqdm(range(len(comb_2_a_2)), desc="Compilando",
    unit="arquivo"):
47     v1 = []
48     v2 = []
49
50     # Calcula a média
51     media = np.mean(sinais[comb_2_a_2[i][0]])
52     v1.append(media)
53     media = np.mean(sinais[comb_2_a_2[i][1]])
54     v2.append(media)
55
56     # Calcula o desvio padrão
57     desvio_padrao = np.std(sinais[comb_2_a_2[i][0]])
58     v1.append(desvio_padrao)
59     desvio_padrao = np.std(sinais[comb_2_a_2[i][1]])
60     v2.append(desvio_padrao)
61
62     # Calcula a variância
63     variancia = np.var(sinais[comb_2_a_2[i][0]])
64     v1.append(variancia)
65     variancia = np.var(sinais[comb_2_a_2[i][1]])
66     v2.append(variancia)
67
68     # Calcula a mediana
69     mediana = np.median(sinais[comb_2_a_2[i][0]])
70     v1.append(mediana)
71     mediana = np.median(sinais[comb_2_a_2[i][1]])
72     v2.append(mediana)
73
74     # Calcula a amplitude
75     amplitude = np.max(sinais[comb_2_a_2[i][0]]) -
    np.min(sinais[comb_2_a_2[i][0]])
76     v1.append(amplitude)
77     amplitude = np.max(sinais[comb_2_a_2[i][1]]) -
    np.min(sinais[comb_2_a_2[i][1]])
78     v2.append(amplitude)
79
80     # Calcula o coeficiente de variação
81     coeficiente_variacao = (desvio_padrao / media) * 100
82     v1.append(coeficiente_variacao)
83     coeficiente_variacao = (desvio_padrao / media) * 100
84     v2.append(coeficiente_variacao)
85
86     # Calcula a skewness
87     skewness = np.mean((sinais[comb_2_a_2[i][0]] - media)**3) /
    (desvio_padrao**3)
88     v1.append(skewness)
```

```

89     skewness = np.mean((sinais[comb_2_a_2[i][1]] - media)**3) /
90         (desvio_padrao**3)
91     v2.append(skewness)
92
93     # Calcula a curtose
94     curtose = np.mean((sinais[comb_2_a_2[i][0]] - media)**4) /
95         (desvio_padrao**4) - 3
96     v1.append(curtose)
97     curtose = np.mean((sinais[comb_2_a_2[i][1]] - media)**4) /
98         (desvio_padrao**4) - 3
99     v2.append(curtose)
100
101     # Transforma a lista em vetor
102     v1 = np.array(v1)
103     v2 = np.array(v2)
104
105     # Calcula a distância euclidiana entre os vetores
106     distancia_euclidiana = np.linalg.norm(v1 - v2)
107
108     # Calcula a distância de Manhattan entre os vetores
109     distancia_manhattan = np.sum(np.abs(v1 - v2))
110
111     # Defina o parametro p para a distância de Minkowski
112     p = 3
113     # Calcula a distância de Minkowski entre os vetores
114     distancia_minkowski = np.power(np.sum(np.power(np.abs(v1 -
115         v2), p)), 1/p)
116
117     # # Imprime os resultados
118     # print("Distância Euclidiana:", distancia_euclidiana)
119     # print("Distância de Manhattan:", distancia_manhattan)
120     # print("Distância de Minkowski (p={}):".format(p),
121         distancia_minkowski)
122     # print(distancia_euclidiana)
123
124     # Se as comparações forem entre projéteis de mesma arma, salva
125     # nos vetores "vetorI..."
126     if (comb_2_a_2[i][0] == comb_2_a_2[i][1]) or
127         ((comb_2_a_2[i][0] + 7) == comb_2_a_2[i][1]) or
128         ((comb_2_a_2[i][0] + 14) == comb_2_a_2[i][1]) or
129         ((comb_2_a_2[i][0] + 21) == comb_2_a_2[i][1]):
130         vetorI1.append(distancia_euclidiana)
131         vetorI2.append(distancia_manhattan)
132         vetorI3.append(distancia_minkowski)
133
134     # Se as comparações forem entre projéteis de armas diferentes,
135     # salva nos vetores "vetorD..."
136     else:
137         vetorD1.append(distancia_euclidiana)

```

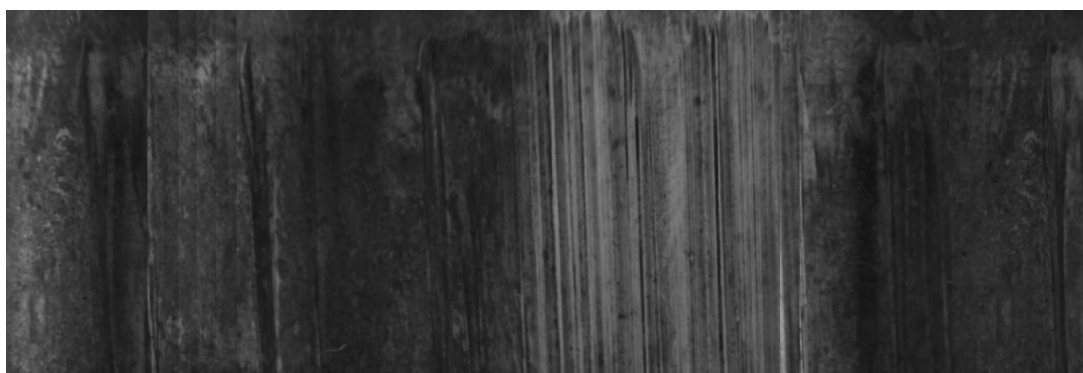
```
131         vetorD2.append(distancia_manhattan)
132         vetorD3.append(distancia_minkowski)
133
134 # Converte a lista em vetor
135 DistI = np.array(vetorI1)
136 DistD = np.array(vetorD1)
137 # DistI = np.array(vetorI2)
138 # DistD = np.array(vetorD2)
139 # DistI = np.array(vetorI3)
140 # DistD = np.array(vetorD3)
141
142 # Calcular o histograma para os vetores
143 hist_vetor1, bins_vetor1 = np.histogram(DistD, bins=20,
144     density=True)
145 hist_vetor2, bins_vetor2 = np.histogram(DistI, bins=20,
146     density=True)
147
148 # Calcular a PDF usando kernel density estimation (KDE)
149 kde_vetor1 = gaussian_kde(DistD)
150 kde_vetor2 = gaussian_kde(DistI)
151
152 # Gerar valores para a PDF
153 x_vetor1 = np.linspace(DistD.min(), DistD.max(), 100)
154 x_vetor2 = np.linspace(DistI.min(), DistI.max(), 100)
155
156 # Plotagem dos gráficos das PDFs. O código gera um gráfico por
157 # vez, sendo preciso trocar os vetores para cada tipo de
158 # distância e atualizar a legenda, título e nome do eixo para
159 # cada gráfico
160 plt.figure(figsize = (16, 8))
161 plt.plot(x_vetor1, kde_vetor1(x_vetor1), label='Projeteis
162     Diferentes')
163 plt.plot(x_vetor2, kde_vetor2(x_vetor2), label='Mesmo Projétil')
164 plt.legend(fontsize=18)
165 plt.grid()
166 plt.xlabel('Distância Euclidiana', fontsize=18)
167 # plt.xlabel('Distância de Manhaattan', fontsize=18)
168 # plt.xlabel('Distância de Minkowski', fontsize=18)
169 plt.ylabel('Densidade de Probabilidade', fontsize=18)
170 plt.title('Função de Densidade de Probabilidade para a Distância
171     Euclidiana', fontsize=18)
172 # plt.title('Função de Densidade de Probabilidade para a Distância
173     de Manhattan', fontsize=18)
174 # plt.title('Função de Densidade de Probabilidade para a Distância
175     de Minkowski com P = 3', fontsize=18)
176 plt.savefig("Resultados/Sinais_Graficos/distanciaEuclidiana.png")
177 #plt.savefig("Resultados/Sinais_Graficos/distanciaManhattan.png")
178 #plt.savefig("Resultados/Sinais_Graficos/distanciaMinkowski.png")
179 plt.show()
```

Anexos

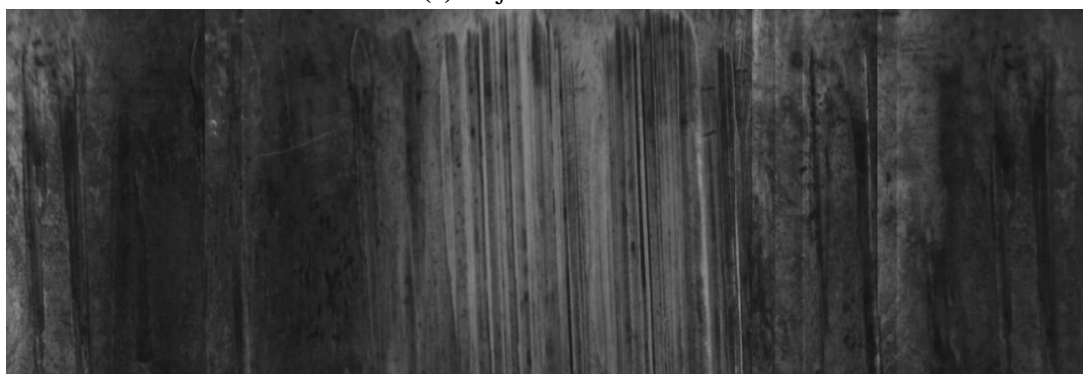
Anexo A – Imagens Externas Utilizadas

A.1 Imagens de Projéteis Utilizadas Para Análise

As imagens a seguir foram retiradas do trabalho de João Bosco ([SILVINO JUNIOR, 2010](#)).



(a) Projétil 1

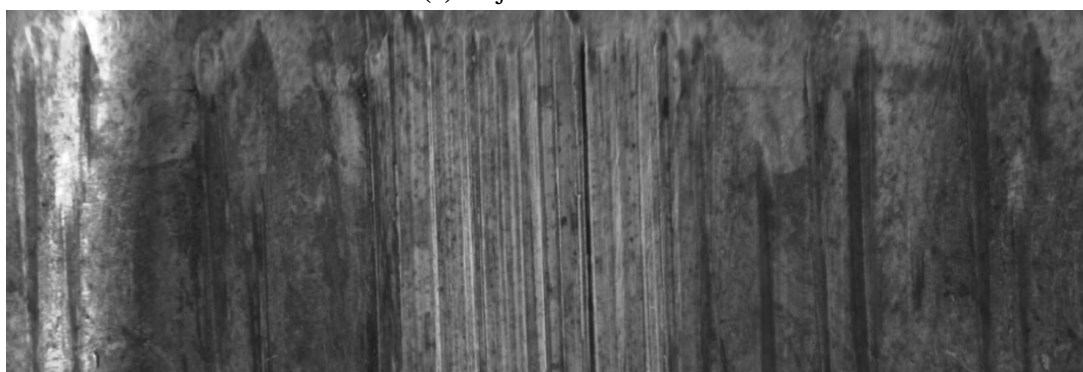


(b) Projétil 2

Figura A.117 – Projéteis da Arma 1 - Fonte: ([SILVINO JUNIOR, 2010](#))

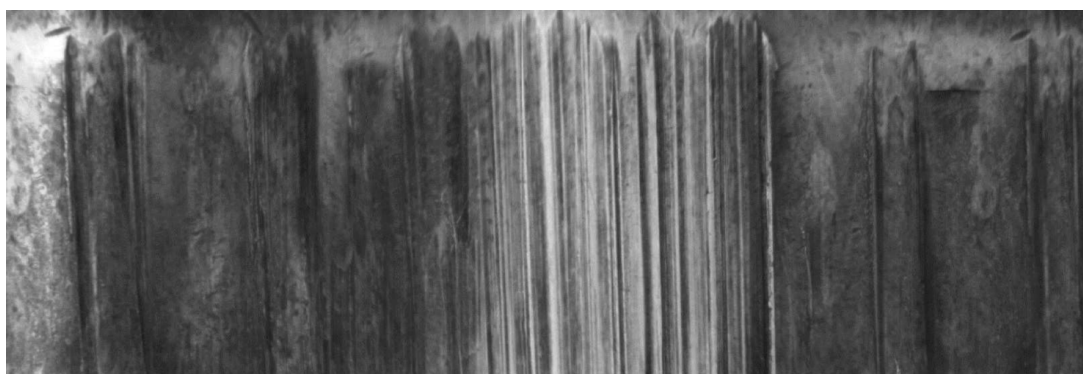


(a) Projétil 1



(b) Projétil 2

Figura A.118 – Projéteis da Arma 2 - Fonte: (SILVINO JUNIOR, 2010)



(a) Projétil 1

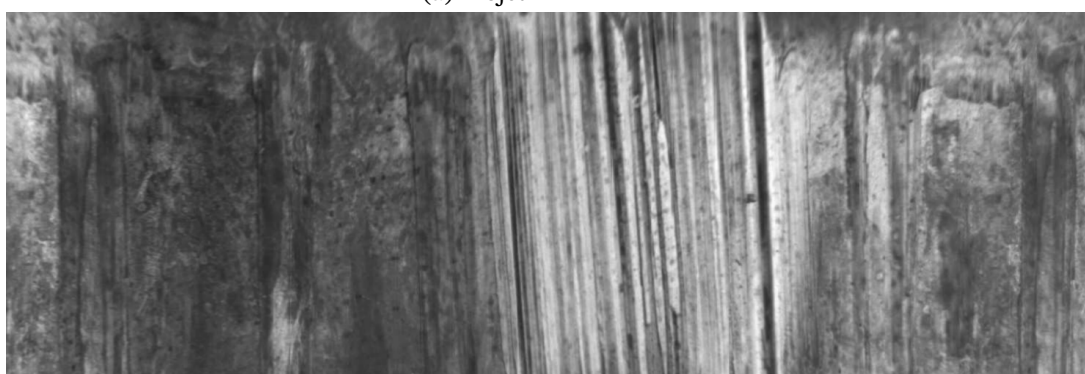


(b) Projétil 2

Figura A.119 – Projéteis da Arma 3 - Fonte: (SILVINO JUNIOR, 2010)

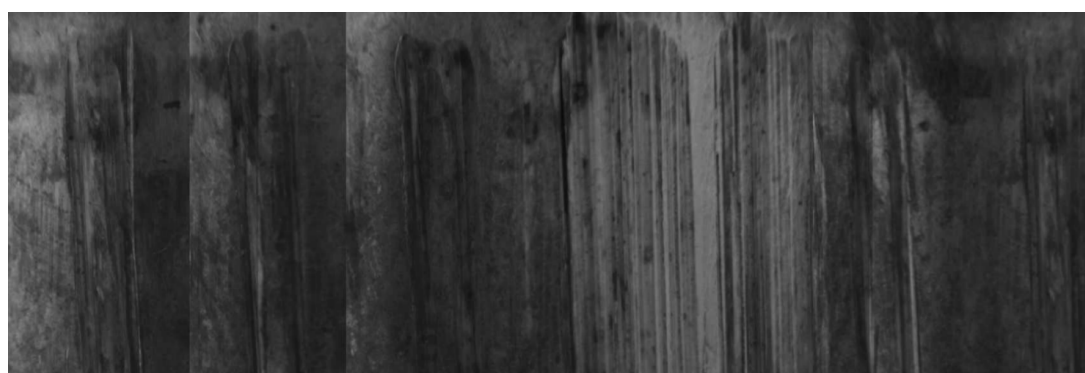


(a) Projétil 1

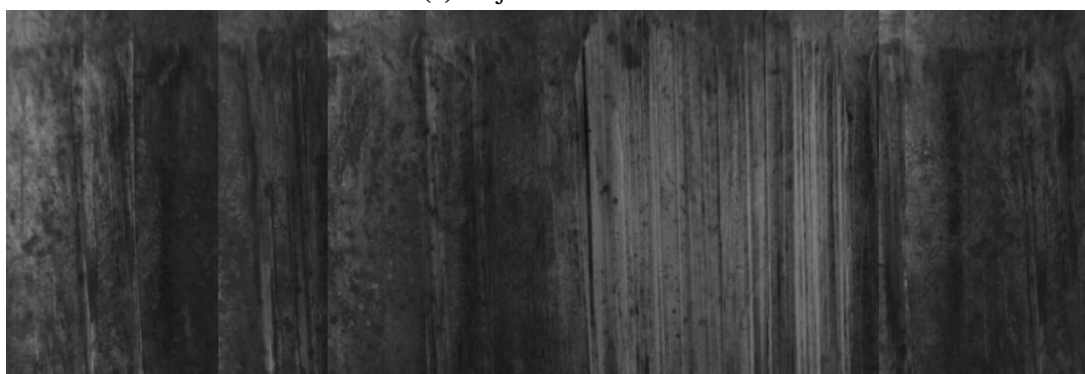


(b) Projétil 2

Figura A.120 – Projéteis da Arma 4 - Fonte: (SILVINO JUNIOR, 2010)

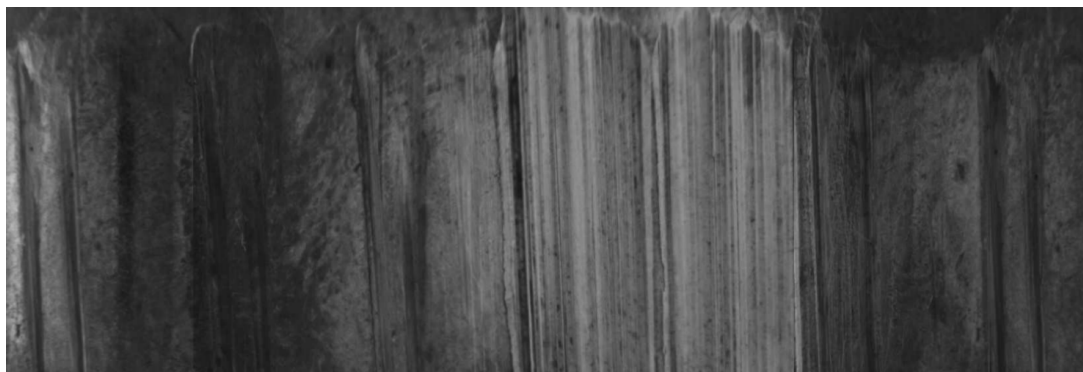


(a) Projétil 1

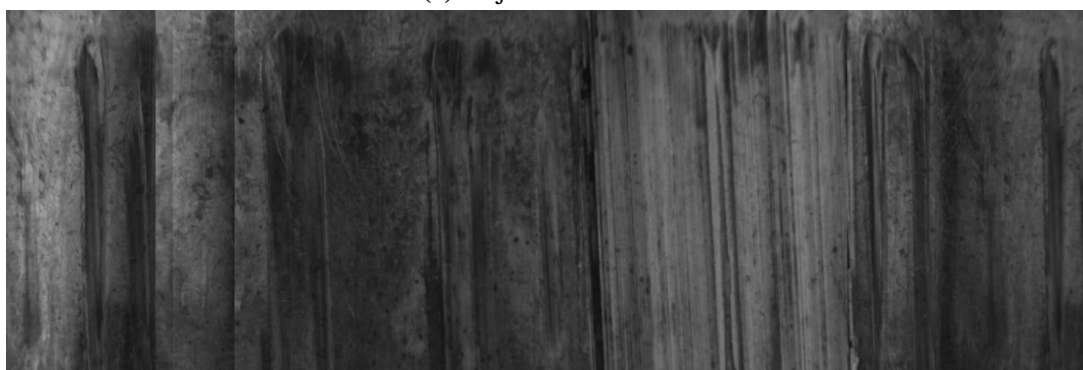


(b) Projétil 2

Figura A.121 – Projéteis da Arma 5 - Fonte: (SILVINO JUNIOR, 2010)



(a) Projétil 1



(b) Projétil 2

Figura A.122 – Projéteis da Arma 6 - Fonte: (SILVINO JUNIOR, 2010)



(a) Projétil 1



(b) Projétil 2

Figura A.123 – Projéteis da Arma 7 - Fonte: (SILVINO JUNIOR, 2010)