

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Carlos Humberto de Paula Borges

**Ajuste polinomial para refinamento do
resultado de segmentação de imagem para
cálculo de volume de ovos**

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Carlos Humberto de Paula Borges

Ajuste polinomial para refinamento do resultado de segmentação de imagem para cálculo de volume de ovos

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Mauricio Cunha Escarpinati

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Carlos Humberto de Paula Borges

Ajuste polinomial para refinamento do resultado de segmentação de imagem para cálculo de volume de ovos

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 30 de Junho de 2023:

Mauricio Cunha Escarpinati
Orientador

Daniel Duarte Abdala

Rafael Dias Araújo

Uberlândia, Brasil
2023

Agradecimentos

Primeiramente, gostaria de agradecer aos meus pais, Carlos Humberto Silva Borges e Márcia de Paula Calil Borges. Vocês foram meu porto seguro e minha fonte constante de motivação. Obrigado por todo o amor, compreensão e encorajamento que me ofereceram durante todo este período. Sem a dedicação e o apoio incondicional de vocês, eu não teria chegado tão longe. Vocês são minha inspiração e agradeço de todo coração por tudo o que fizeram por mim.

À minha amada irmã, Josiane de Paula Borges, sou imensamente grato por sua presença constante em minha vida. Além de ser minha irmã, você foi minha instrutora linguística, sempre me ajudando a aprimorar o texto deste trabalho. Sua habilidade em encontrar as palavras certas e sua paciência em explicar conceitos complexos foram inestimáveis para o sucesso deste projeto. Agradeço por suas valiosas contribuições, que foram além das expectativas e enriqueceram meu trabalho.

À minha querida namorada, Akemi Komino, meu coração transborda de gratidão. Você esteve ao meu lado em todos os momentos, oferecendo seu apoio incondicional, paciência e compreensão. Suas palavras de incentivo e seu suporte emocional foram cruciais para me manter motivado e confiante durante todo o processo. Agradeço por todas as formas possíveis de ajuda que você me ofereceu. Sua presença em minha vida é um presente precioso.

Gostaria também de expressar minha gratidão ao meu melhor amigo, Diego Alves da Silva. Em momentos em que me encontrava bloqueado em raciocínios lógicos e desafiado pela complexidade do trabalho, você esteve ao meu lado, ajudando-me a pensar com clareza e a encontrar soluções. Sua visão única e capacidade de análise foram de valor inestimável. Agradeço por suas contribuições significativas e por estar sempre presente para me ajudar.

Por fim, gostaria de agradecer ao Professor Doutor Mauricio Cunha Escarpinati, que contribuiu para o meu crescimento acadêmico e pessoal. Suas orientações, trocas de ideias e discussões enriquecedoras não foram apenas fundamentais para a conclusão deste trabalho, mas também durante a minha graduação.

"Qualquer bug suficientemente avançado é indistinguível de um recurso." Rich Kulawiec

Resumo

As indústrias avícolas e pesquisas biológicas relacionadas a ovos enfrentam constantemente a demanda por novas técnicas de precisão para obter atributos dos ovos, como o seu volume. Esses atributos são adquiridos por meio do processamento de imagens e cálculos derivados delas, buscando minimizar a intrusão no processo. Conforme a grande quantidade de processamento de imagens e cálculos são realizados, é comum ocorrerem erros e imprecisões ao processar e segmentar imagens contendo objetos, o que afeta a obtenção precisa dos dados dos ovos.

A fim de apresentar novas introspecções para mitigar esses problemas, este trabalho aborda a aplicação do ajuste de curva no cálculo de volume de ovos com a utilização de seus pontos de bordas com ruídos. Ao investigar como o ajuste de curva pode melhorar a precisão desses cálculos, é possível reduzir o impacto dos ruídos nos pontos obtidos. Para tanto, são utilizadas funções polinomiais de diferentes graus para obter o melhor ajuste para o conjunto de pontos. Assim, com a utilização de um conjunto de dados obtidos de vários ovos, e com seus volumes calculados pelo método de deslocamento de água, obtemos o desempenho do ajuste de curva comparável com o volume calculado a partir dos pontos não ajustados, o que resultará em um formato de ovo mais próximo do real, e por conseguinte a obtenção de uma ferramenta confiável de precisão do cálculo do volume para a melhoria de futuras pesquisas no controle de qualidade de ovos.

Palavras-chave: Volume de Ovos, Ruídos, Ajuste de Curva, Precisão, Pontos de Bordas, Distâncias Euclidianas, Cálculo de Volume, Segmentação de Imagem, Funções Polinomiais.

Lista de ilustrações

Figura 1 – Exemplo de de imagem em escala de cinza sobre o par de eixo x e y , onde o ponto de origem $f(0,0)$ fica no canto superior esquerdo (FILHO; NETO, 1999)	16
Figura 2 – Arquiteturas de um sistema de processamento de imagens	17
Figura 3 – Exemplo de aquisição de imagem digital. (a) Fonte de iluminação (energia). (b) Uma cena. (c) Sistema de imagem. (d) Projeção da cena no plano da imagem. (e) Imagem digitalizada. Fonte: (GONZALEZ; WOODS, 2018)	18
Figura 4 – Exemplo de ajuste de curva, onde os pontos de dados da Tabela 1 são exibidos ao longo de uma curva que representa o ajuste ideal de uma função de potência ($\sigma_y = Cd^m$) aos dados (GILAT, 2013)	21
Figura 5 – Exemplo de um gráfico que representa a relação entre tensão [<i>Stress</i>] e deformação [<i>Strain</i>] para borracha. Os marcadores vermelhos representam pontos de dados experimentais altamente precisos, enquanto a curva sólida foi obtida ao usar <i>spline</i> de interpolação. (GILAT, 2013)	22
Figura 6 – Exemplos de ajuste de curva para o mesmo conjunto de pontos de dados com a utilização de polinômios de vários graus.	23
Figura 7 – Exemplo de uma função linear [2.1] (linha reta) usada para ajustar uma curva a n pontos. Há os pontos de dados originais (x_n, y_n) e os novos pontos do ajuste $(x_n, f(x_n))$, com seus respectivos residuais r_n , que são a diferença entre y_n e $f(x_n)$	25
Figura 8 – Exemplos de código que utiliza a função <i>plot()</i> do <i>matplotlib.pyplot</i> para plotar um gráfico com eixo x e y , com apenas o eixo y definido, o que levou a gerar automaticamente os valores $x = [0, 1, 2, 3, 4]$ com base no comprimento da lista de valores y	29
Figura 9 – Exemplos de código que utiliza a função <i>plot()</i> do <i>matplotlib.pyplot</i> para plotar um gráfico com eixo x e y definidos, além de definir rótulos para cada eixo e um título para o gráfico	30
Figura 10 – Exemplo de códigos da biblioteca <i>Openpyxl</i> para criar uma nova pasta de trabalho (1), definir uma nova planilha (2), atribuir um valor "Teste" na célula A1 (3), criar uma referência para a célula A1 (4), definir um novo valor para a célula A1 através de sua referência (5), usar o valor da célula A1 (6) e salvar a nova pasta de trabalho em um diretório (7).	31
Figura 11 – Fluxograma que demonstra o funcionamento da aplicação desde a captura da imagem	36

Figura 12 – Figura de exemplos que demonstram como serão as linhas encontradas para a divisão da metade do ovo, apresentada no início da aplicação . . .	37
Figura 13 – Exemplo de seleção de ovos Bandeja 3 - Leitura 1: A aplicação sempre irá numerar os ovos em ordem crescente, de 1 para cima, conforme a quantidade selecionada	38
Figura 14 – Imagem do ovo número 2 em cada etapa de processamento : Na imagem esquerda, ele em seu formato padrão <i>RGB</i> . Na imagem do meio, ela foi transformada em <i>HSV</i> . Na imagem a direita é o resultado após aplicar a máscara e identificação da caixa delimitadora do ovo.	39
Figura 15 – Nesta figura, é apresentada uma representação visual dos pontos identificados na borda de um ovo, que compõe o conjunto $P = [P_1, P_2, P_3, P_4, P_5, P_6]$, com o objetivo de proporcionar uma melhor visualização. Cada ponto é representado pelas coordenadas (x, y) de um plano cartesiano.	40
Figura 16 – Representação gráfica de todos os pontos obtidos do contorno segmentado do ovo 1.	41
Figura 17 – Exemplo da Figura 15 com o conjunto Q de segmentos de reta, $\overline{P_i P_j}$, para todo $i \neq j$ com uma variação de 1 à 6 em i e j	42
Figura 18 – Exemplo da Figura 17 ao implementar o ovo em um plano cartesiano, com valores definidos para os pontos do eixo X e Y . Através do cálculo da distância euclidiana, foi considerado o segmento de reta $\overline{P_1 P_4}$ como a mais longa.	43
Figura 19 – Exemplo da Figura 18 após encontrar o maior segmento de reta, perpendicular a reta $\overline{A_1 A_4}$. Apenas as retas $\overline{P_2 P_6}$ e $\overline{P_3 P_5}$ estavam aptas a serem escolhidas, sendo a $\overline{P_2 P_6}$ com a maior distância euclidiana. . . .	44
Figura 20 – Figura demonstra como são definidas o novo conjunto K , que possui o tamanho da reta (eixo Y) e a distância euclidiana entre $(0, 0)$ e o ponto onde a reta desta reta fará intersecção com a reta $\overline{A_i A_f}$. Os limites do eixo X serão definidos pelo tamanho da maior reta $\overline{A_i A_f}$ encontrada no ovo, que será o valor n	45
Figura 21 – Esta figura mostra a ilustração do cone com a utilização da distância que está uma reta da outra no eixo X , que corresponde a maior reta $\overline{A_i A_f}$, e a aplicação da distância euclidiana de cada reta para definir a distância do centro do cone a parte mais externa.	46
Figura 22 – Demonstração de um conjunto K com maior número de pontos, e com o tamanho das retas definidas no eixo Y além da distância dos pontos destas retas no eixo X , que será definida pelo segmento de reta $\overline{A_i A_f}$. Esse gráfico é apenas representativa, sem utilizar valores verdadeiros. . .	47

Figura 23 – Demonstração da aplicação de ajuste de curva nos pontos das retas do conjunto K , dados pelo exemplo da Figura 22. Novos pontos de tamanhos das retas serão dados para cada ponto X de cada tamanho de reta que foi dado anterior. A linha vermelha demonstra como será o alinhamento dos novos pontos, em comparação com a linha preta que demonstra a ligação entre os pontos antigos. Esse gráfico é apenas representativa, sem utilizar valores verdadeiros.	48
Figura 24 – Função <i>disc_slices_curve_fit</i> : Utilizada para aplicar o ajuste de curva no <i>xdata</i> e <i>ydata</i> obtidos	49
Figura 25 – Funções polinomiais em forma de código. A função deve receber as coordenadas x e y de seus pontos de dados como entradas e retornar os valores y previstos para cada valor x	49
Figura 26 – Exemplo da foto da Bandeja 1, onde os ovos são numerados da posição 1 a 24, da ordem esquerda-superior para direita-inferior. Todos os ovos estão alinhados verticalmente	51
Figura 27 – Ovo 8 - Gráficos de Ajuste de Curva Polinomial pelo Método dos Quadrados Mínimos - Do gráfico no canto esquerda-superior para o canto direita-inferior : Grau 3, 5, 7, 9 e 11.	52
Figura 28 – Gráfico de erro relativo de cada ovo : Eixo X é a numeração dos ovos, enquanto que o Eixo Y é o valor de erro relativo do volume calculado dos ovos	54
Figura 29 – Ovo 1 - Distância de pontos sem ajuste e com ajuste polinomial de grau 11 da leitura 1 a 4.	55
Figura 30 – Ovo 1 - <i>Pixels</i> do ovo que foram identificados como área interna, através dos seus pontos, da leitura 1 a 4.	55

Lista de tabelas

Tabela 1 – Amostras de metais de teste, caracterizadas por diferentes tamanhos de grão e resistências a tração correspondentes. As variáveis listadas incluem o diâmetro médio do grão (d) e a resistência a tração (σ_y) de cada metal) (GILAT, 2013)	20
Tabela 2 – Erro médio quadrático (Relativo)	53
Tabela 3 – Correlação entre volume real e volume da aplicação	53

Lista de abreviaturas e siglas

API	Application Programming Interface
2D	Two-dimensional
3D	Three-dimensional
HSV	Hue (matiz), Saturation (saturação) e Value (valor)
NumPy	Numerical Python
E/S	Entrada ou Saída
SciPy	Scientific Python
WORM	Write-Once-Read-Many : Escreva uma vez, leia várias vezes
RMSE	Root Mean Squared Error : Raiz do Erro Quadrático Médio

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Organização	14
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Processamento de Imagem Digital	16
2.1.1	Aquisição	17
2.1.2	Armazenamento	18
2.1.3	Processamento	19
2.1.4	Transmissão	19
2.2	Ajuste de Curva e Interpolação	20
2.2.1	Ajuste de Curvas	20
2.2.2	Interpolação	20
2.2.3	Ajuste de curvas com a utilização de polinômios quadráticos e de ordem superior	21
2.2.4	Avaliação da qualidade de um ajuste	24
2.2.5	Regressão Linear e Não-Linear de Mínimos Quadrados	25
2.3	Tecnologias utilizadas	26
2.3.1	NumPy	26
2.3.2	SciPy	27
2.3.2.1	scipy.optimize.curve_fit	27
2.3.3	Matplotlib	28
2.3.3.1	matplotlib.pyplot	28
2.3.4	Openpyxl	29
3	TRABALHOS CORRELATOS	32
3.1	Um sistema de medição de volume de ovo baseado no Microsoft Kinect	32
3.2	Estimativa do volume do ovo com base em processamento de imagem e visão computacional	33
3.3	Deformação da nuvem de pontos 3D baseada em mínimos quadrados móveis e curvas de controle	33
3.4	Análise digital da área e volume da superfície do ovo: Efeitos do eixo longitudinal, largura máxima e peso	34
3.5	Um método de segmentação de imagem baseado em ajuste de curva	35

4	METODOLOGIA	36
4.1	Início da aplicação	36
4.2	Captura da imagem	37
4.3	Segmentação assistida dos ovos	38
4.4	Deteccção das bordas e cálculo das maiores distâncias	39
4.5	Particionamento dos troncos de cone para cálculo do volume	39
4.6	Ajuste de curva em cada particionamento	43
4.7	Geração de relatórios e imagens com resultados	48
5	RESULTADOS E DISCUSSÃO	51
5.1	Máquina e dados utilizados	51
5.2	Gráficos de distância de pontos com e sem ajuste de curva	52
5.3	Resultado de volumes	52
5.4	Discussão	56
5.4.1	Gráfico de Erro Relativo	57
6	CONCLUSÕES	58
	REFERÊNCIAS	59

1 Introdução

A importância do estudo do volume dos ovos possui implicações tanto para a avicultura quanto para pesquisas biológicas; já que a sua medida pode prever a ecodibilidade do ovo, as características de qualidade da casca, o estudo da população e da morfologia ecológica, a previsão do peso do pintinho e a análise do interior do ovo, como dito por Narushin *et al* (2005) em (NARUSHIN, 2005).

Uma das formas de se obter o volume do ovo é através do método de deslocamento de água (Princípio de Arquimedes) (KERN; COWIE, 1996; SZÉKELY; KOZMA; PITI, 1994). Contudo, por exigir medições e cálculos físicos, a utilização desse método possui algumas falhas, como o tempo que se levaria para fazê-la sobre cada ovo; o que a torna impraticável para produção em larga escala, além de ser intrusiva ao molhá-los, propensa a erros e subjetiva, de acordo com Okinda *et al* (2020) em (OKINDA *et al.*, 2020).

Em contrapartida, conforme observado por Okinda *et al* (2020) em (OKINDA *et al.*, 2020), o avanço da tecnologia de visão computacional, a qual é uma ciência que estuda a obtenção de informações de imagens digitais (KLETTE, 2014), trouxe métodos rápidos, precisos e não intrusivos para estimar o volume e outros atributos dos produtos agrícolas. Entre esses métodos, encontra-se os de Monte Carlo (SISWANTORO; PRABUWONO; ABDULAH, 2013), a reconstrução do modelo de wire-frame 3D (LEE *et al.*, 2006), a de curvatura espacial (CHALIDABHONGSE; YIMYAM; SIRISOMBOON, 2006), a técnica de lofting (GONI; PURLIS; SALVADORI, 2007), a de scanners 3D (UYAR; ERDOĞDU, 2009), a que utiliza o teorema de Pappus (SOLTANI; OMID; ALIMARDANI, 2015), a de reconstrução 3D (ZHANG *et al.*, 2016) e a de sensor de infravermelho (OKINDA *et al.*, 2020).

Este trabalho colabora com um estudo em andamento, que faz parte de uma dissertação de mestrado em desenvolvimento na Faculdade de Computação da Universidade Federal de Uberlândia. Nele, são propostos métodos computacionais para calcular o volume de ovos de galinha a partir de imagens digitais bidimensionais. O método desenvolvido utiliza distâncias euclidianas entre as bordas perpendiculares à maior dimensão do ovo, que possibilita a simulação de cortes sagitais. No entanto, devido a certas irregularidades no cálculo dessas distâncias decorrentes do processo de segmentação da imagem, acredita-se que o cálculo do volume final possa ser prejudicado pelas distorções presentes nesses pontos obtidos da borda do ovo.

Para contornar essa questão, foi proposto o uso de ajuste de curva, um procedimento no qual se utiliza de uma fórmula matemática para ajustar um determinado conjunto de pontos de dados. Ele é obtido a partir da borda do ovo, e resulta em uma

representação mais suave e contínua do contorno do ovo, com a redução do impacto dos ruídos e a melhoria da precisão do cálculo do volume (GILAT, 2013).

1.1 Objetivos

O objetivo deste trabalho é investigar como o uso do método de ajuste de curvas pode melhorar a precisão dos cálculos de volume de ovos ao reduzir o impacto dos ruídos nos pontos obtidos de suas bordas. Para isso, foram usados funções polinomiais de diferentes graus para obter o melhor ajuste para o conjunto de pontos obtidos.

O desempenho do ajuste de curva foi avaliado através da utilização de um conjunto de dados obtidos de vários ovos, com seus volumes conhecidos através do cálculo de volume por deslocamento de água. Foi realizado uma análise comparativa para avaliar os ganhos de precisão obtidos pelo ajuste de curva em relação ao volume calculado dos pontos que não foram ajustados.

É esperado que essa pesquisa contribua na precisão do cálculo de volume de ovos ao deixar os valores calculados mais próximos dos valores obtidos pelo cálculo real de volume por deslocamento de água.

1.1.1 Organização

Esse trabalho é iniciado com o capítulo de Introdução, o qual apresentou uma visão geral completa dos requisitos e objetivos deste trabalho, bem como explora as aplicações e a importância de estudar o volume do ovo.

O segundo capítulo é o de Revisão Bibliográfica, que forneceu uma explicação detalhada da abordagem de ajuste polinomial de curva de mínimos quadrados e da tecnologia usada na criação desse trabalho. Também é discutido a utilidade do ajuste de curvas para analisar os pontos que definem as bordas dos objetos em imagens, enfatizando como ele se conecta a outras linhas de investigação na área.

No terceiro capítulo, intitulado Trabalhos Correlatos, foram apresentados estudos relacionados que investigam várias abordagens e estratégias de programação para medir propriedades de ovos, bem como trabalhos que mostram como o ajuste de curvas pode ser usado para reparar erros em pontos que foram detectados em objetos de imagem. Essa revisão da literatura ajuda a identificar lacunas de pesquisa e o estado atual da arte.

Mais detalhes sobre a metodologia utilizada para produzir esse trabalho são descritos no quarto capítulo de Metodologia. É descrito o processo de desenvolvimento e implantação do sistema para que o leitor tenha uma noção clara das ações e práticas utilizadas. Esta parte busca facilitar a replicação do software criado e sua utilização em estudos futuros.

Os resultados obtidos através deste projeto são apresentados no quinto capítulo, Resultados, juntamente com uma análise e discussão aprofundadas. Os resultados são avaliados à luz dos objetivos propostos, que demonstra contribuições significativas e possíveis direções para trabalhos futuros.

As conclusões tiradas desta pesquisa são então descritas no sexto e último capítulo, Conclusões, descreve as principais descobertas, contribuições e limitações.

2 Revisão Bibliográfica

2.1 Processamento de Imagem Digital

Inicialmente é definido que uma imagem pode ser identificada como uma função bidimensional, denominada como $f(x, y)$, onde x e y representam as coordenadas espaciais em um plano e que o valor de f em qualquer par de coordenadas (x, y) corresponde à intensidade ou nível de cinza da imagem naquele ponto específico (GONZALEZ; WOODS, 2018). Assim, é possível ver uma imagem em escala de cinza sobre o par de eixo x e y , na Figura 1.

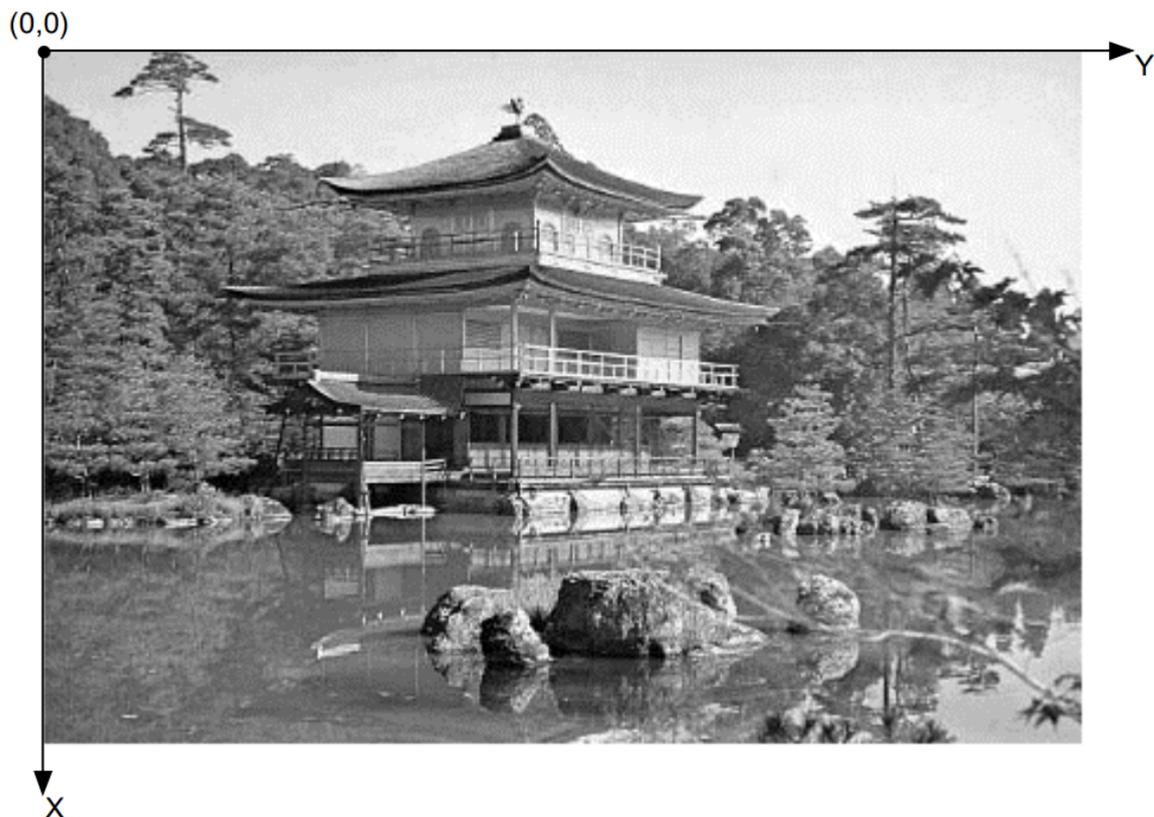


Figura 1 – Exemplo de de imagem em escala de cinza sobre o par de eixo x e y , onde o ponto de origem $f(0,0)$ fica no canto superior esquerdo (FILHO; NETO, 1999)

Já uma imagem é dita como digital quando x , y e os valores de intensidade da função f são todos finitos, em quantidades discretas (GONZALEZ; WOODS, 2018).

Um sistema de processamento de imagens consiste de vários componentes e operações, como pode ser visto na Figura 2, com o intuito de fazer a análise de uma imagem. As operações de aquisição, processamento e armazenamento serão explicadas nas seções

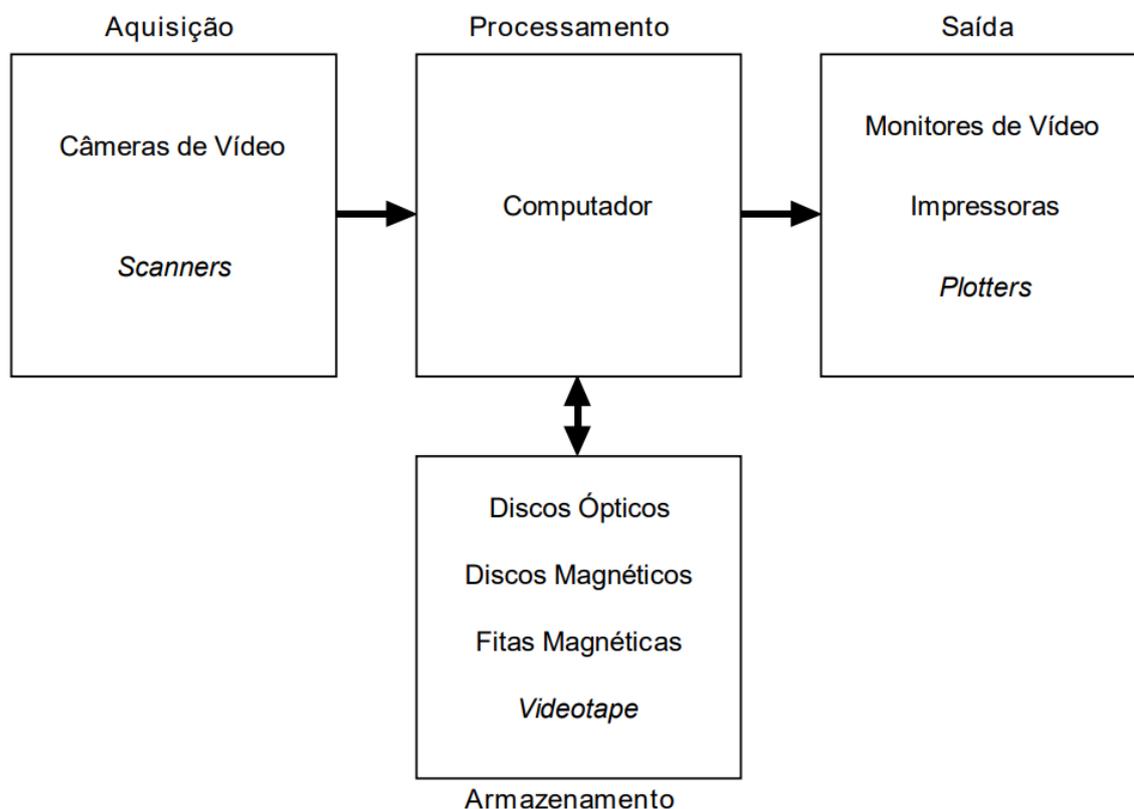


Figura 1 - Elementos de um sistema de processamento de imagens.

Figura 2 – Arquiteturas de um sistema de processamento de imagens. Fonte: (FILHO; NETO, 1999)

seguinte.

2.1.1 Aquisição

A operação de aquisição serve para transformar uma imagem em um formato digital para processamento posterior, tendo como resultado a conversão para um formato numérico que seja adequado para a operação seguinte (FILHO; NETO, 1999). Ela é dividida em duas etapas:

1. Através da detecção da energia do espectro eletromagnético, ao utilizar certos dispositivos físicos como raios X, ultravioleta, luz visível ou infravermelho. E também através de métodos tradicionais, como o uso de ultrassom ou padrões de iluminação gerados por computador (FILHO; NETO, 1999);
2. A etapa seguinte, conhecida como digitalizador, converte o sinal elétrico analógico obtido em informação digital com o uso de código binário (0s e 1s). Esse módulo muitas vezes é chamado de "placa de captura de imagens" (*frame grabber*) (FILHO; NETO, 1999).

Um exemplo de aquisição de imagem pode ser visto na Figura 3, que representa a energia de uma fonte de iluminação refletida de uma cena com a transmissão dessa energia para um sistema de imagem. Esse sistema tem como função de coletar e focalizar a energia recebida em um plano de imagem (FILHO; NETO, 1999). A matriz de sensores, localizada no plano da imagem, gera saídas proporcionais à luz integrada recebida por cada sensor. Essas saídas são processadas por circuitos digitais e analógicos, que as convertem em um sinal analógico (FILHO; NETO, 1999). Dessa forma, outra seção do sistema de imagem digitaliza o sinal analógico, que resulta em uma imagem digital (FILHO; NETO, 1999).

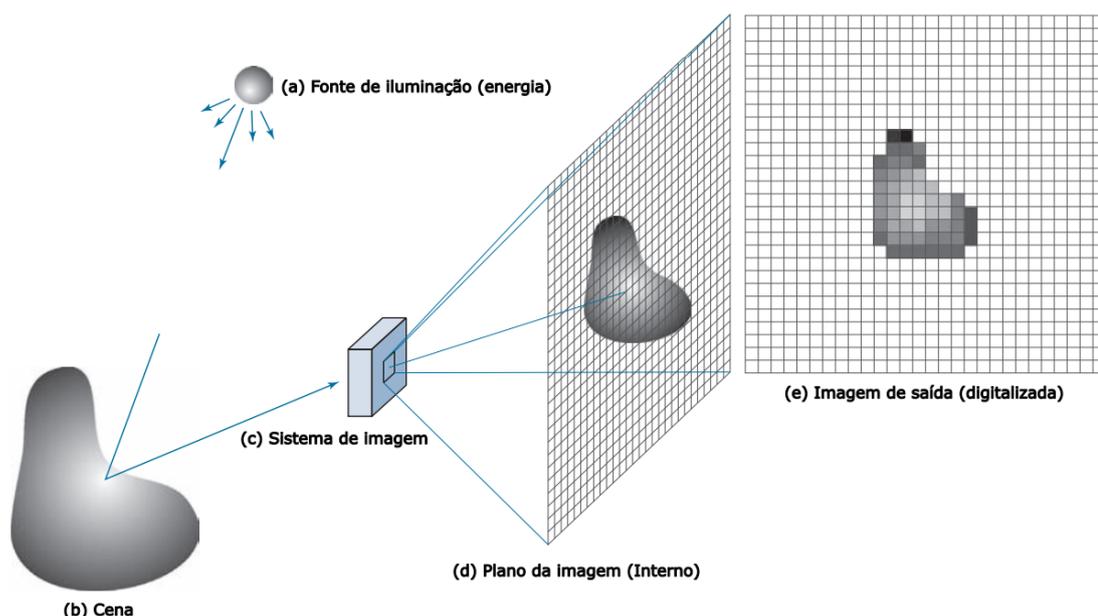


Figura 3 – Exemplo de aquisição de imagem digital. (a) Fonte de iluminação (energia). (b) Uma cena. (c) Sistema de imagem. (d) Projeção da cena no plano da imagem. (e) Imagem digitalizada. Fonte: (GONZALEZ; WOODS, 2018)

2.1.2 Armazenamento

Uma das principais dificuldades em projetar sistemas de processamento de imagem é como armazenar imagens digitais, devido a grande quantidade de bytes necessários. (FILHO; NETO, 1999). Há três formas de fazê-la:

Armazenamento de curto prazo para processamento : Pode ser feito ao usar a RAM do computador, ou através de *frame buffers*, que permitem acesso em alta velocidade e suportam operações como zoom e rolagem (FILHO; NETO, 1999).

Armazenamento em massa para recuperação rápida : Como o tempo de acesso as imagens é um fator crucial, faz-se necessário o uso de discos magnéticos ou magneto-ópticos nesta forma (FILHO; NETO, 1999). O espaço de armazenamento necessário

pode ser estimado com base na resolução da imagem e na profundidade da escala de cinza (FILHO; NETO, 1999);

Arquivamento para armazenamento de longo prazo : Envolve o armazenamento de grandes quantidades de dados na forma de imagens que raramente são acessadas, em discos tipo *WORM* como CDs, DVDs e Blu-Rays (FILHO; NETO, 1999).

2.1.3 Processamento

O processamento de imagens digitais envolve o uso de procedimentos algorítmicos que podem ser implementados por meio de um software (FILHO; NETO, 1999). Nos casos de limitações no computador que fará o processamento, pode ser necessário o uso de um hardware especializado para o processamento da imagem (FILHO; NETO, 1999).

No mercado de hardware há uma tendência crescente de venda de placas genéricas compatíveis com arquiteturas populares de microcomputadores e estações de trabalho (FILHO; NETO, 1999). Essas placas possuem um software de controle que determina qual é a funcionalidade de cada uma. As vantagens dessa abordagem incluem redução de custos, modularidade, capacidade de reutilizar componentes de software no mesmo hardware e independência de fornecedores específicos (FILHO; NETO, 1999).

Contudo, ainda há sistemas dedicados que são utilizados para tarefas específicas, como o processamento de imagens transmitidas por satélite (FILHO; NETO, 1999).

Existem muitas operações diferentes que podem ser usadas para analisar imagens; desde operações aritméticas e espaciais, análise de histogramas, operações de conjuntos e aplicação de transformações à função que compõe a figura (FILHO; NETO, 1999).

2.1.4 Transmissão

Transmitir imagens digitalizadas a longas distâncias com o uso de redes de computadores e protocolos de comunicação representa um grande desafio devido à grande quantidade de dados envolvidos (FILHO; NETO, 1999). Essa dificuldade aumenta ao transmitir sequências de vídeo em tempo real, pois exigem sincronização e consideração de fatores como canais de baixa velocidade e largura de banda limitada (FILHO; NETO, 1999). Para resolver isso, é preciso fazer o uso de técnicas de compressão e descompressão de imagem (FILHO; NETO, 1999).

As técnicas de compressão de imagem têm o objetivo de reduzir o tamanho das imagens através da remoção de informações redundantes (FILHO; NETO, 1999). O interesse na compressão de imagem existe há várias décadas, pioneiramente com técnicas analógicas e posteriormente com o surgimento de multimídia, sistemas distribuídos e várias novas tecnologias que dependem de imagens comprimidas, como videoconferência,

TV de alta definição, TV interativa e serviços de vídeo sob demanda (FILHO; NETO, 1999).

2.2 Ajuste de Curva e Interpolação

Estudos científicos e de engenharia dependem de experimentos que medem e registram quantidades físicas, que resultam em pontos de dados. Muitas vezes esses pontos são digitalizados e armazenados em computadores para estudos (GILAT, 2013). Para representar e analisar tais pontos com eficiência, é empregado o ajuste de curva sobre eles.

2.2.1 Ajuste de Curvas

O ajuste de curva é um procedimento em que uma equação matemática é usada para regular um determinado conjunto de pontos de dados (GILAT, 2013). Esse procedimento é necessário quando os pontos de dados exibem erros ou dispersão, o que geralmente ocorre em medições experimentais (GILAT, 2013).

Sendo assim, define-se uma função que ajuste os pontos de dados em geral, de forma que não precise fornecer o valor exato em cada ponto, mas sim ajustar os dados como um todo (GILAT, 2013). Um exemplo de ajuste de curva pode ser visto na Figura 4, onde a curva não corresponde exatamente a nenhum dos pontos de dados da Tabela 1, mas reflete a tendência geral dos dados.

Tabela 1 – Amostras de metais de teste, caracterizadas por diferentes tamanhos de grão e resistências a tração correspondentes. As variáveis listadas incluem o diâmetro médio do grão (d) e a resistência a tração (σ_y) de cada metal (GILAT, 2013)

d(mm)	0.005	0.009	0.016	0.025	0.040	0.062	0.085	0.110
σ_y(MPa)	205	150	135	97	89	80	70	67

O ajuste da curva também é útil para avaliar a validade das equações propostas que representam os dados e determinar os valores dos coeficientes ou parâmetros (GILAT, 2013).

2.2.2 Interpolação

Nesta pesquisa foi necessário medir os valores entre os pontos de dados que foram obtidos. Para isso, foi utilizado um método chamado Interpolação (GILAT, 2013).

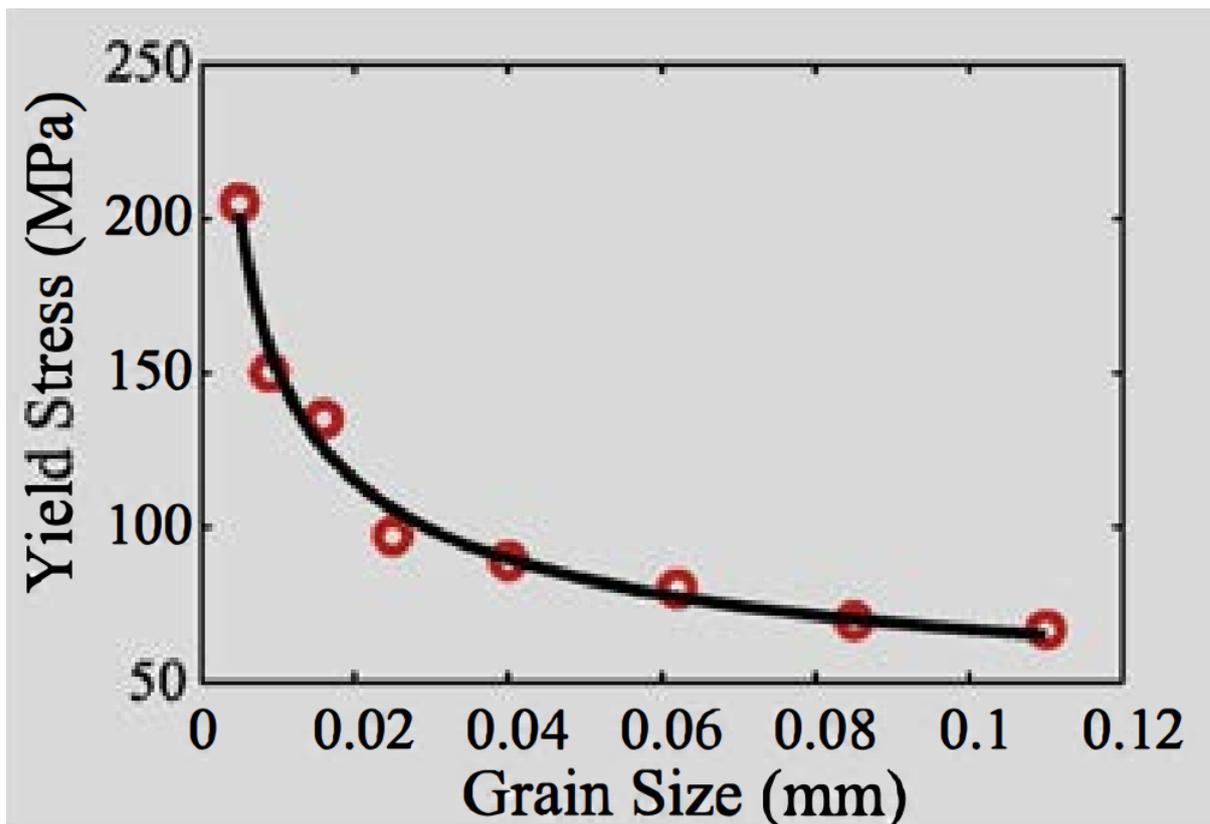


Figura 4 – Exemplo de ajuste de curva, onde os pontos de dados da Tabela 1 são exibidos ao longo de uma curva que representa o ajuste ideal de uma função de potência ($\sigma_y = Cd^m$) aos dados (GILAT, 2013)

O objetivo é encontrar um polinômio, como o da Equação 2.2, que corresponda aos valores exatos nos pontos dados para calcular os valores dentro desse intervalo (GILAT, 2013).

Quando a quantidade de pontos de dados é pequena, um único polinômio pode ser suficiente para interpolar todo o intervalo. Porém, quando se trata de um número maior de pontos, são utilizados diferentes polinômios nos intervalos entre os pontos, o que é conhecido como *spline* de interpolação (GILAT, 2013).

Um exemplo de interpolação pode ser visto na Figura 5, onde os pontos dos dados são exatamente percorridos pela curva; o que também oferece estimativas precisas para os valores entre eles (GILAT, 2013).

2.2.3 Ajuste de curvas com a utilização de polinômios quadráticos e de ordem superior

Para o propósito desta tese de graduação, será usado o método dos mínimos quadrados para fazer o ajuste não-linear das curvas, com o uso de polinômios quadráticos e de ordem superior.

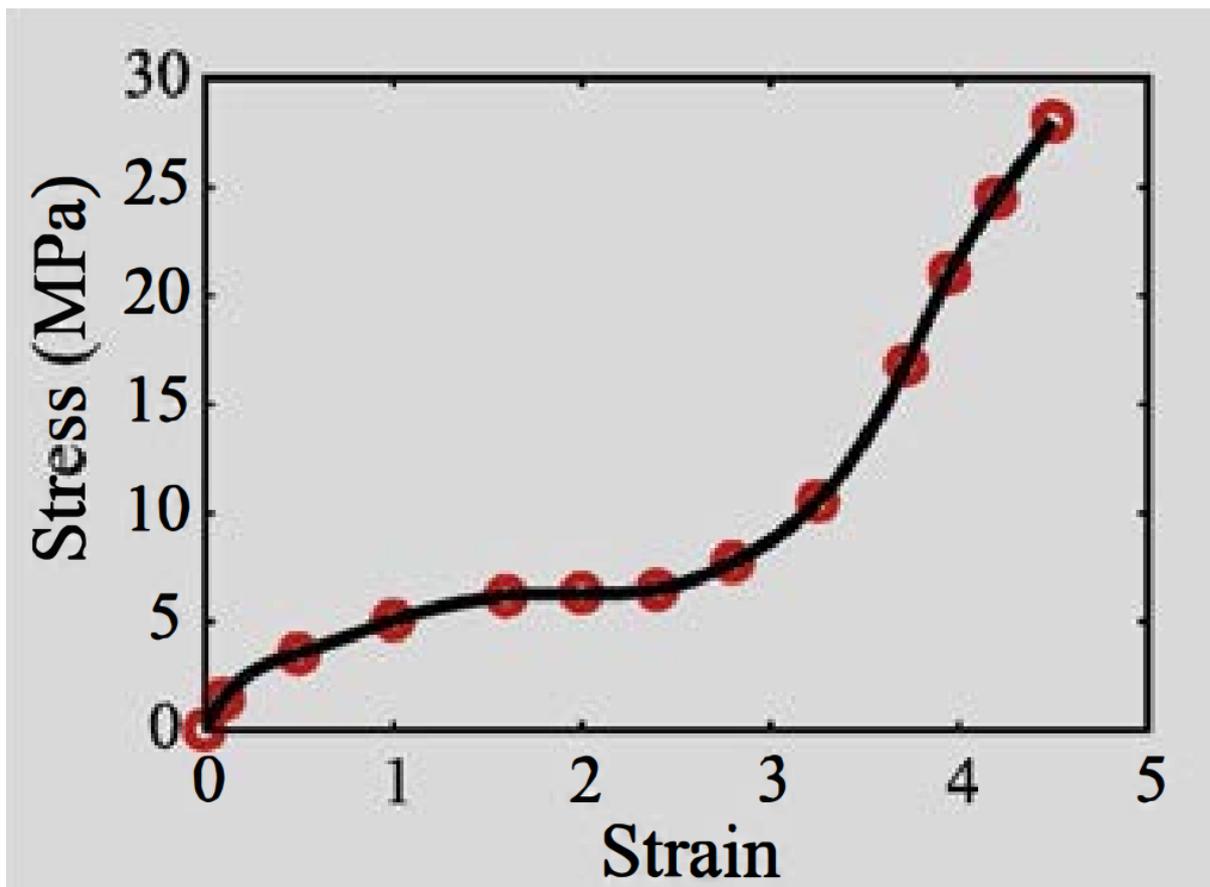


Figura 5 – Exemplo de um gráfico que representa a relação entre tensão [*Stress*] e deformação [*Strain*] para borracha. Os marcadores vermelhos representam pontos de dados experimentais altamente precisos, enquanto a curva sólida foi obtida ao usar *spline* de interpolação. (GILAT, 2013)

O ajuste pode ser feito sobre os pontos com funções polinomiais de diferentes graus. A equação polinomial de primeiro grau 2.1 é usada para ajustar os pontos dados, que criará um ajuste linear entre os pontos da melhor forma possível.

$$y = a_1x + a_0 \quad (2.1)$$

O ajuste será feito após definir os valores de seus respectivos coeficientes como números reais, neste caso sendo a_1 e a_0 , para que possam resultar no menor erro possível quando os dados forem substituídos na Equação 2.1.

Ao aumentar o grau da função polinomial para acima de 1, ela aumentará a sua quantidade de termos conforme o grau escolhido:

$$f(x) = a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 \quad (2.2)$$

Onde o valor de n , que é um número inteiro não negativo, é o grau escolhido do

polinômio, todos os polinômios com o grau acima de 1 são não-lineares. A curva então se torna mais flexível e pode capturar padrões mais complexos nos dados à medida que o grau aumenta (GILAT, 2013).

A Figura 6 demonstra o ajuste de curva com polinômios de ordens diferentes para o mesmo conjunto de pontos de dados, com o aumento da curvatura e a aproximação dos pontos.

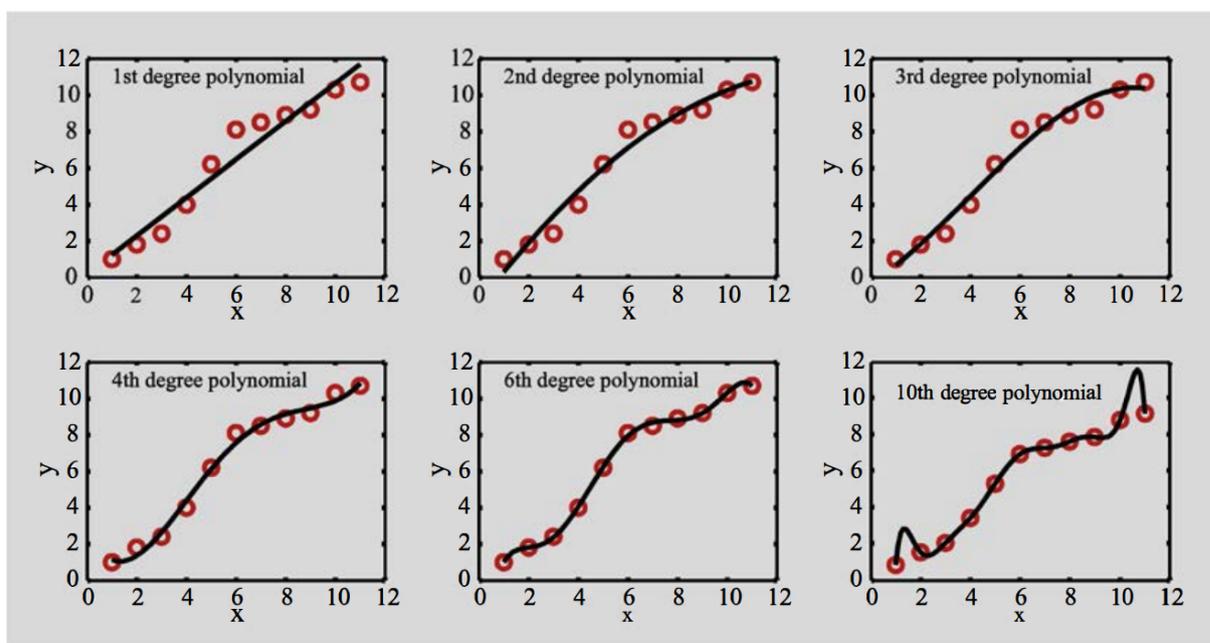


Figura 6 – Exemplos de ajuste de curva para o mesmo conjunto de pontos de dados com a utilização de polinômios de vários graus.

No entanto, existem limitações ao utilizar polinômios de ordem superior para o ajuste de curva. Um polinômio de grau alto pode passar por todos os pontos de dados precisamente quando há muitos deles, mas pode desviar significativamente entre alguns devido o aumento de erros e imprecisões pegas (GILAT, 2013); repare esse fenômeno na exibição da Figura 6 do polinômio de décimo grau. A tendência geral entre pontos adjacentes pode não ser seguida ainda que represente com precisão os pontos de dados (GILAT, 2013).

Portanto, não é aconselhável usar polinômios de ordem superior para interpolação. Por outro lado, um polinômio de ordem superior pode ser útil quando valores precisos estão disponíveis e o objetivo é representar os dados com precisão (GILAT, 2013).

Ao se obter diversas formas de se fazer o ajuste, é preciso diferenciar entre eles qual será o mais eficiente. Será necessário medir a qualidade do ajuste e utilizar um método matemático para determinar os valores dos coeficientes $a_n, a_{n-1}, \dots, a_1, a_0$ que forneçam o melhor ajuste (GILAT, 2013).

2.2.4 Avaliação da qualidade de um ajuste

Será necessário utilizar uma fórmula que calcula um número para quantificar a concordância geral entre os pontos e medir a qualidade de um ajuste, além de uma fórmula próxima para fazer o ajuste, como a equação 2.1 (GILAT, 2013).

Essa tarefa será dividida em duas etapas :

1. Comparar duas funções de ajuste para os mesmos pontos;
2. Determinar os coeficientes da função escolhida de ajuste.

Ao iniciar a primeira parte, deverá ser descoberto primeiramente os erros residuais, que são a variação entre cada ponto de dados e o valor da função de aproximação. Eles serão utilizados para calcular o erro total para todos os pontos (GILAT, 2013).

A fórmula para encontrar o erro residual de um único ponto é definida na Equação 2.3, onde y_i é o valor do ponto no eixo y e o valor da função $f(x_i)$ que foi utilizada para fazer a aproximação dos pontos de dados :

$$r_i = y_i - f(x_i) \quad (2.3)$$

Um exemplo da obtenção dos erros residuais pode ser visto na Figura 7.

Para determinar a eficácia da função escolhida, como as equações polinomiais, e fazer a aproximação de pontos de dados, usa-se o cálculo do erro total (E) com os residuais (GILAT, 2013).

Há várias formas para se calcular o erro total (GILAT, 2013), mas o método que possui melhores resultados, além de também ser utilizado nesta tese, é o da soma dos quadrados dos residuais; como pode ser visto na Equação 2.4. Essa forma resulta em residuais sempre positivos, que retira a possibilidade de residuais positivos e negativos se anularem, além de permitir que residuais maiores possam impactar ainda mais o erro total (GILAT, 2013).

$$E = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n [y_i - f(x_i)]^2 \quad (2.4)$$

A Equação 2.4 também será utilizada para encontrar os coeficientes $a_n, a_{n-1}, \dots, a_1, a_0$ da equação de aproximação que foi escolhida. Para isso, será utilizada uma técnica chamada de Regressão Linear e Não-Linear de Mínimos Quadrados.

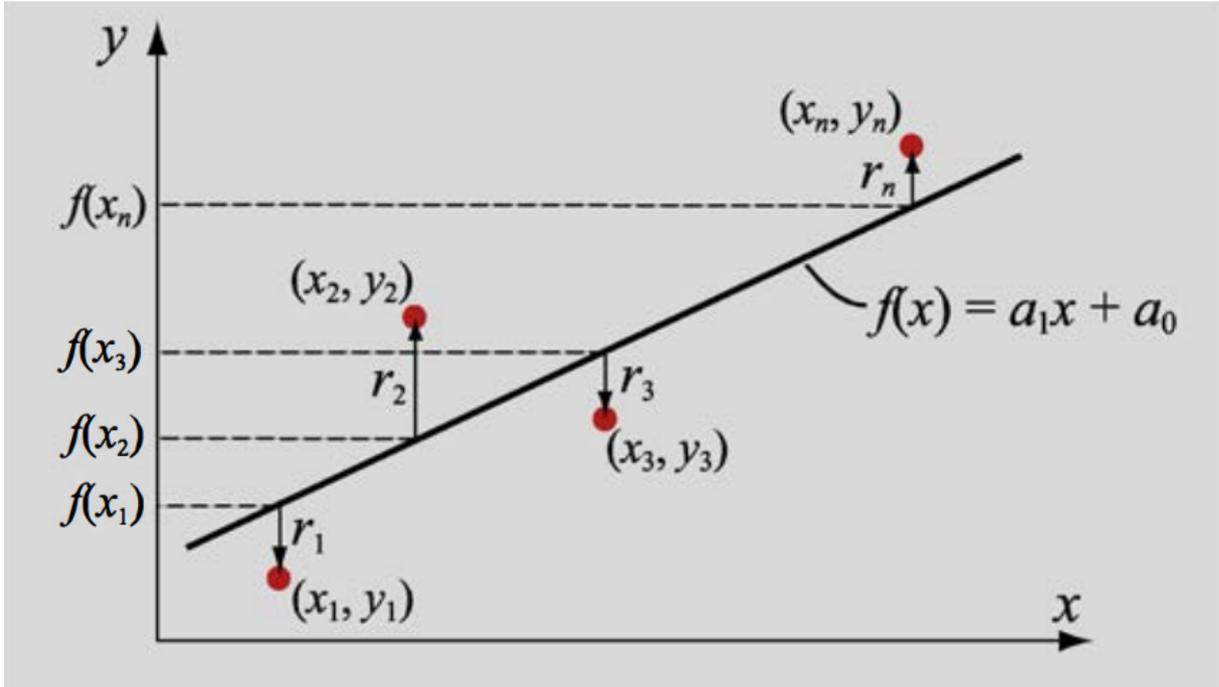


Figura 7 – Exemplo de uma função linear [2.1] (linha reta) usada para ajustar uma curva a n pontos. Há os pontos de dados originais (x_n, y_n) e os novos pontos do ajuste $(x_n, f(x_n))$, com seus respectivos residuais r_n , que são a diferença entre y_n e $f(x_n)$.

2.2.5 Regressão Linear e Não-Linear de Mínimos Quadrados

Em um processo conhecido como regressão linear ou não-linear de mínimos quadrados, os coeficientes $a_n, a_{n-1}, \dots, a_1, a_0$ de uma equação polinomial 2.2 de grau n escolhida serão determinados para que a função melhor se adeque aos pontos de dados que serão ajustados. O menor erro total, conforme determinado pelo cálculo dos quadrados dos resíduos de acordo com a Equação 2.4, é o que se chama de melhor ajuste (GILAT, 2013).

A Equação 2.4 atinge o seu menor valor ao tornar as derivadas parciais de E , em relação a cada variável a_0 até a_n , iguais a zero. Assim, podemos resolver um conjunto de $m + 1$ equações lineares para encontrar os coeficientes (GILAT, 2013).

Como exemplo, usaremos uma equação polinomial de grau 2 para fazer o ajuste, como mostrado na Equação 2.5 :

$$f(x) = a_2x^2 + a_1x + a_0 \quad (2.5)$$

Quando as derivadas parciais em relação a a_0, a_1 e a_2 são tomadas e definidas como zero, o seguinte resultado é obtido:

$$\frac{\partial E}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_2x^2 - a_1x - a_0) = 0 \quad (2.6)$$

$$\frac{\partial E}{\partial a_1} = -2 \sum_{i=1}^n (y_i - a_2 x^2 - a_1 x - a_0) x_i = 0 \quad (2.7)$$

$$\frac{\partial E}{\partial a_2} = -2 \sum_{i=1}^n (y_i - a_2 x^2 - a_1 x - a_0) x_i^2 = 0 \quad (2.8)$$

Um sistema de três equações lineares para os coeficientes a_0 , a_1 e a_2 é representado pelas equações (2.6) a (2.8) e pode ser reescrito da seguinte forma:

$$n a_0 + \left(\sum_{i=1}^n x_i \right) a_1 + \left(\sum_{i=1}^n x_i^2 \right) a_2 = \sum_{i=1}^n y_i \quad (2.9)$$

$$\left(\sum_{i=1}^n x_i \right) a_0 + \left(\sum_{i=1}^n x_i^2 \right) a_1 + \left(\sum_{i=1}^n x_i^3 \right) a_2 = \sum_{i=1}^n x_i y_i \quad (2.10)$$

$$\left(\sum_{i=1}^n x_i^2 \right) a_0 + \left(\sum_{i=1}^n x_i^3 \right) a_1 + \left(\sum_{i=1}^n x_i^4 \right) a_2 = \sum_{i=1}^n x_i^2 y_i \quad (2.11)$$

Ao resolver o sistema das equações (2.9)-(2.11), será obtido os valores dos coeficientes a_0 , a_1 e a_2 que serão empregados no cálculo da equação 2.5, sendo possível ajustar os pontos (x_i, y_i) (GILAT, 2013).

2.3 Tecnologias utilizadas

Nesta seção, serão apresentadas as bibliotecas em *Python* que foram utilizadas nesse trabalho para facilitar o desenvolvimento e a execução eficiente das etapas de manipulação, transformação e análise das imagens, contribuindo para o seu objetivo.

2.3.1 NumPy

NumPy é um pacote *Python* fundamental para computação científica que fornece um objeto de vetor multidimensional (*ndarray*) e uma variedade de funções para executar operações de vetores eficientes (HARRIS et al., 2020). Os vetores *NumPy*, ao contrário dos vetores padrões do *Python*, têm um tamanho fixo e precisam que os itens sejam do mesmo tipo de dados (HARRIS et al., 2020); isso permite que operações matemáticas e estatísticas avançadas em grandes conjuntos de dados sejam executadas com um melhor desempenho (HARRIS et al., 2020).

Como muitas bibliotecas científicas do *Python* dependem de vetores *NumPy*, o conhecimento do *NumPy* é necessário para trabalhar com software científico e matemático (HARRIS et al., 2020).

O *NumPy* oferece suporte a operações matemáticas, manipulação de formas, classificação, E/S, álgebra linear, estatísticas e outros recursos (HARRIS et al., 2020). A integração perfeita do *NumPy* com outras bibliotecas *Python*, como *SciPy* e *Matplotlib*, aumenta sua versatilidade e cria um ecossistema robusto para computação científica e análise de dados.

2.3.2 SciPy

SciPy é uma biblioteca *Python* robusta para computação científica que fornece uma coleção abrangente de rotinas numéricas e ferramentas para modelar e resolver problemas científicos (VIRTANEN et al., 2020).

O *SciPy* suporta otimização, integração, interpolação, problemas de autovalor, equações algébricas, equações diferenciais e uma variedade de outras classes de problemas por meio de um conjunto diversificado de algoritmos. Também inclui estruturas de dados avançadas, como matrizes esparsas e árvores k-dimensionais (VIRTANEN et al., 2020).

O *SciPy* faz uso dos recursos do *NumPy* (HARRIS et al., 2020) como a base sobre a qual as bibliotecas científicas de alto nível são construídas, o que oferece estruturas de dados de matriz eficazes e operações numéricas (VIRTANEN et al., 2020).

No projeto em questão, foi empregado o módulo *scipy.optimize* com ênfase na função *curve_fit*, a qual foi utilizada para realizar o ajuste de curva nos pontos adquiridos.

2.3.2.1 scipy.optimize.curve_fit

A função *curve_fit* no *SciPy* é uma ferramenta para fazer o ajuste de curvas que permite a estimativa de parâmetros otimizados para um determinado modelo matemático com base nos dados observados (VIRTANEN et al., 2020). Ele minimiza a soma residual dos quadrados entre os pontos de dados e as previsões do modelo, o que gera um melhor ajuste (VIRTANEN et al., 2020).

Utilizar *curve_fit* requer duas entradas essenciais: a função alvo a ser ajustada e uma estimativa inicial dos parâmetros. A função deve ser definida para receber a variável independente como primeiro argumento, seguida dos parâmetros que precisam ser estimados. Após a execução, *curve_fit* retorna os parâmetros otimizados, bem como a matriz de covariância, que oferece informações valiosas sobre as incertezas associadas aos parâmetros estimados. Essa combinação de parâmetros otimizados e matriz de covariância fornece aos pesquisadores informações valiosas para avaliar a confiabilidade e a importância do modelo ajustado (VIRTANEN et al., 2020).

Ao empregar por padrão o algoritmo *Levenberg-Marquardt* (MORE, 1978; MORE; GARBOW; HILLSTROM, 1980) para a otimização não-linear de mínimos quadrados, o *curve_fit* lida com uma ampla gama de funções e tipos de dados. O algoritmo ajusta

iterativamente os parâmetros do modelo para minimizar a soma dos quadrados dos resíduos com a obtenção do ajuste ideal para os dados observados; o que torna o *curve_fit* particularmente valioso quando os métodos de regressão linear são insuficientes devido às relações complexas ou não-lineares entre as variáveis (VIRTANEN et al., 2020).

2.3.3 Matplotlib

A biblioteca *Python Matplotlib* possibilita aos usuários plotar matrizes em 2D. Embora o objetivo original do *Matplotlib* era de imitar a funcionalidade dos comandos gráficos do *MATLAB*, ele se desenvolveu desde então em uma ferramenta autônoma, que segue as convenções da comunidade *Python* para a criação de visualização. A fim de garantir um desempenho eficaz, mesmo com grandes conjuntos de dados, ele fornece uma abordagem orientada a objetos e é frequentemente usado em conjunto com o *NumPy* e outras extensões (HUNTER, 2007).

Existem três partes conceitualmente distintas na base de código *Matplotlib* :

- O módulo *matplotlib.pylab* fornece um conjunto de funções por meio da interface *pylab* que permite aos usuários criar gráficos ao utilizar um código muito semelhante ao que é usado para criar figuras no *MATLAB*. Ao simplificar o processo de criação de gráficos essa interface traz mais facilidade para os usuários, que já estão familiarizados com o *MATLAB* (HUNTER, 2007);
- Em segundo lugar, o *front-end matplotlib* ou *API matplotlib* é composto de classes que gerenciam e produzem gráficos, linhas, plotagens, texto e outros componentes visuais. Essa camada de abstração gerencia os principais recursos do *Matplotlib* e é independente do formato ou dispositivo de saída (HUNTER, 2007);
- Por fim, os *back-ends* no *Matplotlib* são renderizadores dependentes de dispositivo que convertem a representação do *front-end* em diferentes formatos de saída, como cópia impressa ou dispositivos de exibição (HUNTER, 2007).

Abaixo será explicado sobre o módulo do *Matplotlib* utilizado pelo projeto : *matplotlib.pyplot*.

2.3.3.1 matplotlib.pyplot

O módulo *matplotlib.pyplot* consiste em várias funções que oferecem uma interface de estilo de comando para criar gráficos. Essas funções incluem aquelas para criar uma figura, especificar uma área de plotagem, desenhar linhas ou curvas, adicionar rótulos e muito mais (HUNTER, 2007).

A função `plot()` é uma das mais usadas do módulo `pyplot`. Quando você atribui um vetor à função `plot()`, `matplotlib` a interpreta como uma sequência de valores y e gera os valores x correspondentes para o usuário. Como os intervalos do *Python* começam com 0, os valores x são gerados por padrão como $[0, 1, 2, 3, \dots]$. É possível plotar os valores de y em relação aos seus índices correspondentes (HUNTER, 2007).

No exemplo de código na Figura 8, a função `plot()` gera automaticamente os valores $x = [0, 1, 2, 3, 4]$ com base no comprimento da lista de valores y . O gráfico resultante na mesma figura mostra os valores de y em relação aos seus índices correspondentes no eixo x .

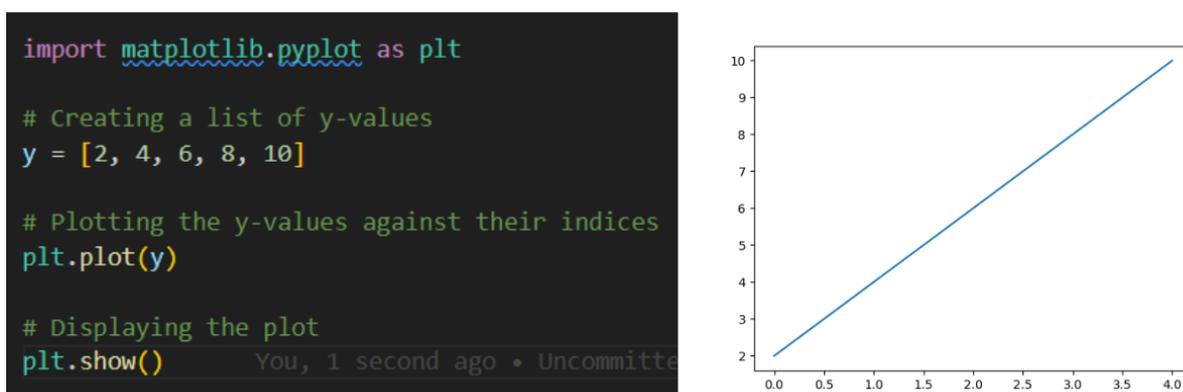


Figura 8 – Exemplos de código que utiliza a função `plot()` do `matplotlib.pyplot` para plotar um gráfico com eixo x e y , com apenas o eixo y definido, o que levou a gerar automaticamente os valores $x = [0, 1, 2, 3, 4]$ com base no comprimento da lista de valores y .

No exemplo mostrado na Figura 9, a função `plot()` é utilizada com dois argumentos: uma lista de valores x e uma lista de valores y . Ao conectar os pontos correspondentes nos pares (x, y) , a função gera um gráfico de linha. Além disso, personalizações são aplicadas para aprimorar a representação visual do enredo. Os rótulos são adicionados aos eixos x e y ao usar as funções `xlabel` e `ylabel`, respectivamente, enquanto um título é atribuído ao gráfico ao usar a função `title()`.

2.3.4 Openpyxl

A biblioteca *Openpyxl* é uma ferramenta *Python* para trabalhar com arquivos do *Excel* que permite automatizar tarefas de planilhas, como a de criar, ler, modificar e salvar pastas.

Um arquivo do *Excel*, também conhecido como pasta de trabalho, consiste em uma ou mais planilhas. Cada planilha compõe uma grade de células organizadas em linhas e colunas. Dados de todos os tipos, que inclui fórmulas, texto, datas e números, são armazenados nas células. Cálculos baseados em valores de células são possíveis com

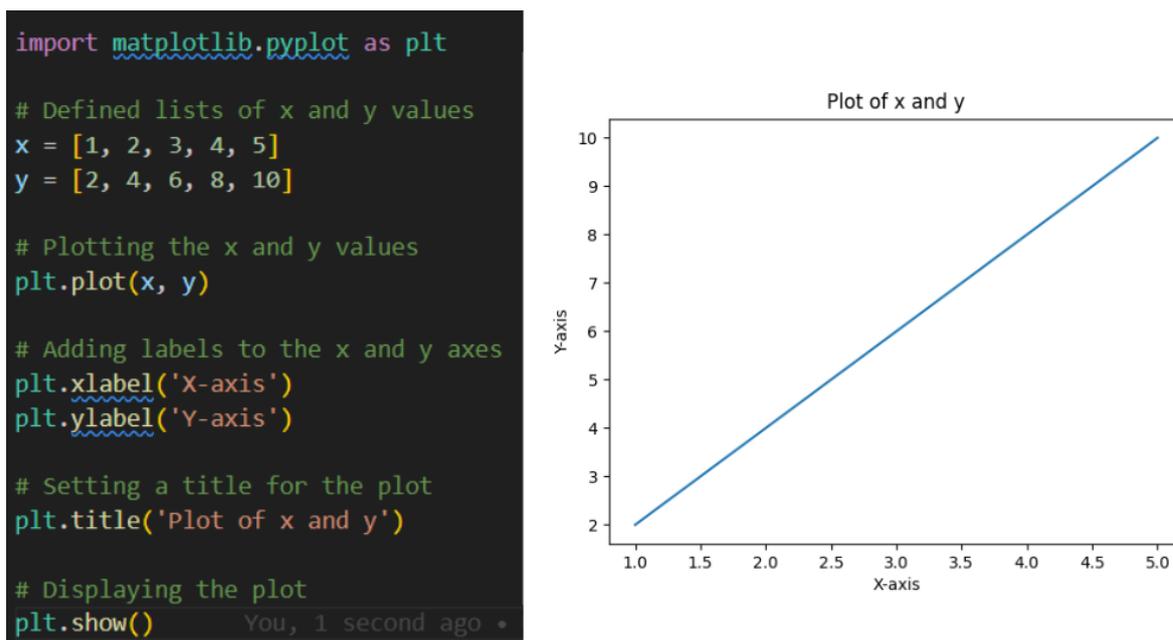


Figura 9 – Exemplos de código que utiliza a função `plot()` do `matplotlib.pyplot` para plotar um gráfico com eixo x e y definidos, além de definir rótulos para cada eixo e um título para o gráfico

fórmulas. Outros componentes, como cabeçalhos, rodapés, estilos, formatação e validação de dados, podem estar presentes em arquivos do *Excel*. Em muitos campos diferentes, eles são comumente usados para organização, análise e visualização de dados.

Para se trabalhar com uma nova pasta de trabalho com *openpyxl*, ela deve ser criada com a utilização da função `Workbook()`.

Uma vez criada a pasta de trabalho, pode-se adicionar novas planilhas a ela. Essa nova pasta de trabalho é produzida com uma única planilha denominada "*Sheet*" por padrão. É possível alterar o nome ou adicionar novas planilhas conforme necessário.

Com uma planilha declarada, pode-se manipular células individuais dentro dela. As células são referenciadas ao fazer o uso de uma combinação da letra da coluna e do número da linha. Com a referência da célula obtida, seu valor é lido com o atributo "*value*" e pode ser modificado para lhe atribuir um novo valor. A referência das células também pode ser obtida ao empregar índices numéricos. Pode-se visualizar os códigos das últimas explicações na Figura 10.

```
from openpyxl import Workbook

# 1 - Criar nova pasta de trabalho
workbook1 = Workbook()

# 2 - Criar nova planilha dentro da pasta
worksheet = workbook1.create_sheet("Nova Planilha")

# 3 - Definir um valor "Teste" na célula A1
worksheet["A1"] = "Teste"

# 4 - Criar uma referência para a célula A1
celulaA1 = worksheet["A1"]

# 5 - Definir um novo valor para a célula A1 através de sua referência
celulaA1.value = "Teste2"

# 6 - Usar o valor da célula A1
print(celulaA1.value)

# 7 - Salvar o arquivo Excel
workbook1.save("C:/teste.xlsx")
```

Figura 10 – Exemplo de códigos da biblioteca *Openpyxl* para criar uma nova pasta de trabalho (1), definir uma nova planilha (2), atribuir um valor "Teste" na célula A1 (3), criar uma referência para a célula A1 (4), definir um novo valor para a célula A1 através de sua referência (5), usar o valor da célula A1 (6) e salvar a nova pasta de trabalho em um diretório (7).

3 Trabalhos Correlatos

Este capítulo se concentra na revisão de trabalhos correlatos que fizeram contribuições significativas para o campo do ajuste de curva.

3.1 Um sistema de medição de volume de ovo baseado no Microsoft Kinect

Essa pesquisa feita por Ting On Chan *et al* (2018) em (CHAN *et al.*, 2018) apresenta um novo protótipo de sistema metrológico para medir com precisão o volume de ovos de aves, que é importante na avicultura e na pesquisa ornitológica. O sistema faz uso de uma câmera de alcance *3D*, notadamente o *Microsoft Kinect* (Versão 2), além de um algoritmo de pós-processamento para nuvens de pontos.

O volume do ovo é determinado diretamente ao manejar os parâmetros de sua forma, previstos e obtidos com o uso do método dos mínimos quadrados (CHAN *et al.*, 2018). Para fazer isso, modelos geométricos de ovos recém-criados em um espaço *3D* devem ser ajustados às nuvens de pontos dos ovos que o *Kinect* capturou (CHAN *et al.*, 2018). Este método também permite que o sistema determine simultaneamente a posição, a orientação e as características da forma do ovo (CHAN *et al.*, 2018).

Quatro conjuntos de experimentos foram realizados para avaliar a funcionalidade e o desempenho do sistema, que compara os volumes previstos com os adquiridos ao lidar com um scanner a laser e o método tradicional de deslocamento de água (CHAN *et al.*, 2018).

Os resultados mostram que a abordagem sugerida é simples, viável e confiável, com uma precisão média de 93,3% no cálculo dos volumes de ovos em comparação com os volumes de referência.

Essa pesquisa fornece uma perspectiva complementar e amplia o conhecimento sobre técnicas e métodos computacionais utilizados na medição precisa de volumes de ovos através de imagens 2D obtidas com equipamentos diferentes, o que pode contribuir para a obtenção de resultados mais confiáveis e precisos na aplicação da dissertação de mestrado, ao se utilizar diferentes fontes de obtenção de imagens.

3.2 Estimativa do volume do ovo com base em processamento de imagem e visão computacional

O estudo feito por Cedric Okinda *et al* (2020) em (OKINDA *et al.*, 2020) apresenta uma abordagem baseada em imagens de profundidade e em análise de curvatura de contorno dos ovos identificados, para calcular o volume de ovos de galinha. Em uma linha de classificação ele analisa duas configurações de ovos: um único ovo sem oclusão e um grupo de ovos com oclusão parcial em cenários simples e complexos. Essa abordagem oferece uma maneira distinta de obter os contornos de múltiplos ovos em diferentes posições em um plano, contribuindo para o aprimoramento do trabalho atual ao demonstrar diferentes formas de se obter os contornos.

Com o emprego da análise de curvatura de contorno e os algoritmos de centro de círculo M mais próximo k , os ovos ocluídos são segmentados.

Foram treinados e testados 13 modelos de regressão com base em características coletadas de imagens de ovos individuais. A técnica de Regressão Exponencial do Processo Gaussiano superou todos os outros modelos, com um *RMSE* de $1,175 \text{ cm}^3$ e um R^2 de 0,984. O mesmo modelo calculou volumes para ovos parcialmente ocluídos com um *RMSE* de $1,080 \text{ cm}^3$ para oclusão simples e $1,294 \text{ cm}^3$ para oclusão complexa.

Ambos os trabalhos abordam a estimativa do volume de ovos de galinha por meio do processamento de imagens e segmentação. Essa abordagem oferece uma maneira distinta de obter os contornos de múltiplos ovos em diferentes posições em um plano, contribuindo para a compreensão e aprimoramento do processo de estimativa de volume desses ovos.

3.3 Deformação da nuvem de pontos 3D baseada em mínimos quadrados móveis e curvas de controle

Neste estudo feito por Rui Wang *et al* (2013) em (WANG *et al.*, 2013), um método de deformação com a aplicação de mínimos quadrados móveis e curvas de controle é apresentado para modelos de nuvens de pontos 3D. A manipulação da curva é usada para controlar o processo de deformação.

Com base nos dados da topologia, os pontos críticos são identificados e a interpolação *spline* cúbica é usada para ajustar as curvas a esses pontos. Através do ajuste de contorno, o método sugerido manipula com precisão o contorno distorcido de um objeto.

Ao contrário dos controles por pontos, as transformações rígidas, que são usadas localmente, oferecem um controle mais preciso. Além disso, essa abordagem pode ser usada para deformações em qualquer dimensão.

Os resultados dos experimentos mostram que a deformação da nuvem de pontos 3D e o ajuste de informações de contorno podem ser implementados com sucesso e produzir resultados de deformação precisos.

Ambos os trabalhos demonstram o uso de técnicas avançadas para manipular e ajustar informações de contorno de diferentes objetos, visando obter resultados precisos em diferentes contextos. Dessa forma, o trabalho mencionado complementa esse estudo ao fornecer diferentes compreensões sobre o uso de métodos de deformação e ajuste de contornos em dados tridimensionais, que podem ser relevantes para aprimorar a precisão da estimativa de volume de ovos de galinha.

3.4 Análise digital da área e volume da superfície do ovo: Efeitos do eixo longitudinal, largura máxima e peso

O estudo feito por Mohammad Sedghi *et al* (2013) em (SEDGHI; GHADERI, 2023) empregou a tecnologia de análise de imagem digital (*IA*) para medir com precisão o volume do ovo (V) e a área de superfície (*textitS*) de galinhas poedeiras. Ao comparar o V adquirido pelo *IA* com medições feitas usando deslocamento de água, a precisão da abordagem *IA* foi avaliada. Essa comparação produziu uma correlação de determinação (R^2) de 0,98 entre os dois métodos.

O estudo também analisou a aplicabilidade de equações publicadas anteriormente para prever S e V , com os dados coletados pelo *IA*. Com base no comprimento (L) e na largura máxima (B), as equações apresentadas por Carter (CARTER, 1975), Paganelli *et al.* (PAGANELLI; OLSZOWKA; AR, 1974) e Narushin (NARUSHIN, 1997) prognosticaram o ovo S com bastante precisão ($R^2 > 0,839$). Semelhante a isso, com L e B como entradas, as equações fornecidas por Carter (CARTER, 1975), Ayupov (AYUPOV, 1976) e Narushin (NARUSHIN, 1994; NARUSHIN, 1997; NARUSHIN, 2005) ofereceram previsões precisas para o ovo V .

O estudo também investigou o uso de modelos de redes neurais artificiais (RNA), regressão linear múltipla (RLM) e modelos de regressão polinomial (RP) para criar novas equações para estimar o volume do ovo e a área de superfície com base no peso do ovo, L e B . Ao contrário da maioria das fórmulas publicadas, os resultados mostraram que o peso sozinho não era uma variável de entrada útil. Mesmo assim, os modelos recém-criados com base em L e B foram bem-sucedidos em prever corretamente o V e S de amostras de ovos.

Essa pesquisa oferece informações relevantes para esse estudo, ao explorar técnicas semelhantes e o uso de inteligência artificial para aprimorar a medição do volume de ovos de galinha. Ambos os estudos utilizam a técnica de criar cones de troncos de partes dos

ovos e somá-los para calcular o volume. No entanto, o estudo de Sedghi *et al.* (SEDGHI; GHADERI, 2023) aplica essa abordagem com o uso de inteligência artificial para melhorar os ângulos entre as distâncias dos pontos das bordas dos ovos, resultando em medições precisas.

3.5 Um método de segmentação de imagem baseado em ajuste de curva

O estudo feito por Wei Shen *et al* (2009) em (SHEN; WU; TU, 2009) propôs um algoritmo com um método automatizado para segmentar o tabuleiro do jogo *Snooker*. Assim como a aplicação da dissertação de mestrado, esse estudo também buscar obter os pontos em um plano cartesiano das bordas de uma mesa de *Snooker* e aplica o ajuste de curva para reduzir os ruídos, o qual oferece uma perspectiva similar a esse estudo, ao explorar técnicas de segmentação e ajuste de curva em imagens.

Inicialmente, o algoritmo começa com a extração das arestas ao usar a segunda derivada separável de filtros gaussianos orientáveis nas direções horizontal e vertical. Esses filtros combinam vários filtros básicos diferentes que fornecem filtragem direcional. Cada filtro base é aplicado à imagem de entrada $f(x,y)$ e a imagem resultante $g(x,y)$ é produzida pela soma ponderada das saídas.

As regiões candidatas do quadro são diferenciadas por segmentação de cores. Para combinar pontos de fundo com objetos, é utilizada a dilatação da imagem, que ajuda na extração precisa dos pixels das bordas. Em seguida, com base no valor de pico dos pixels de borda e na região da placa candidata, os pixels de borda candidatos são escolhidos. Por fim, o ajuste de curva é usado para gerar pixels de borda reais e a placa é segmentada com base nos resultados do ajuste de curva.

4 Metodologia

A aplicação principal deste trabalho funciona de maneira local, em que é utilizado um único código-fonte desenvolvido na linguagem de programação *Python*, juntamente com várias bibliotecas.

O funcionamento da aplicação segue a ordem descrita no fluxograma ilustrado na Figura 11.

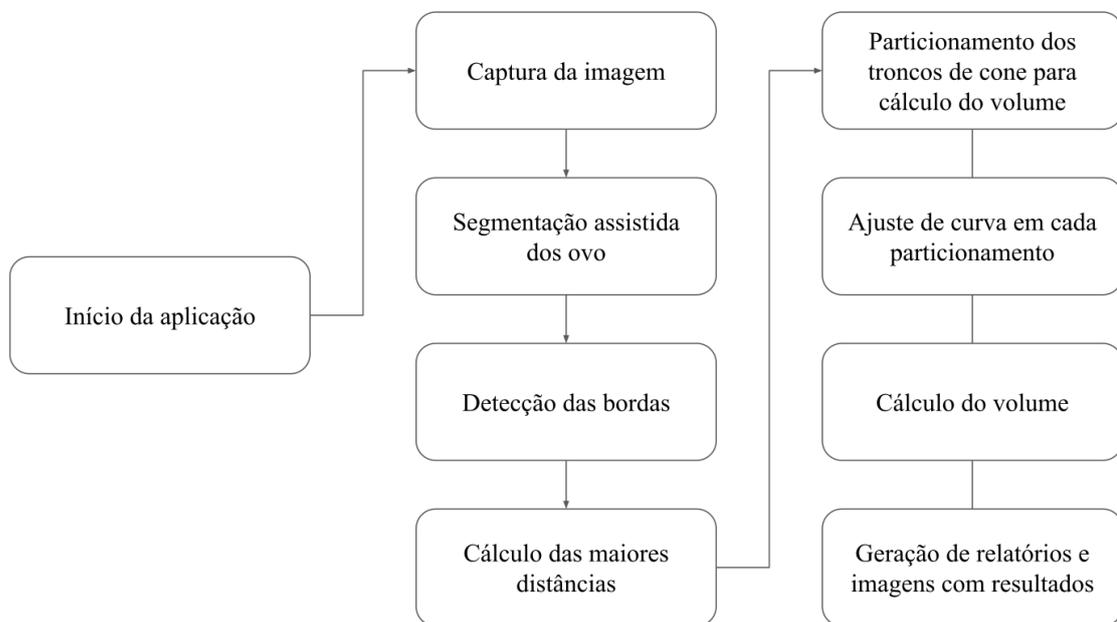


Figura 11 – Fluxograma que demonstra o funcionamento da aplicação desde a captura da imagem

4.1 Início da aplicação

Para iniciar a aplicação, é necessário executar o arquivo *main.py* na versão *Python 3* por meio de uma linha de comando.

Após a execução do comando escolhido pelo usuário, uma janela será aberta, conforme exemplo mostrado na Figura 12, que ilustra como serão as medidas a serem calculadas a partir da imagem digital do ovo.

Algumas opções de seleção, ao pressionar as teclas de atalho, foram implementadas no sistema para permitir a iteração do usuário, são elas:

Q Pressionado para sair do programa

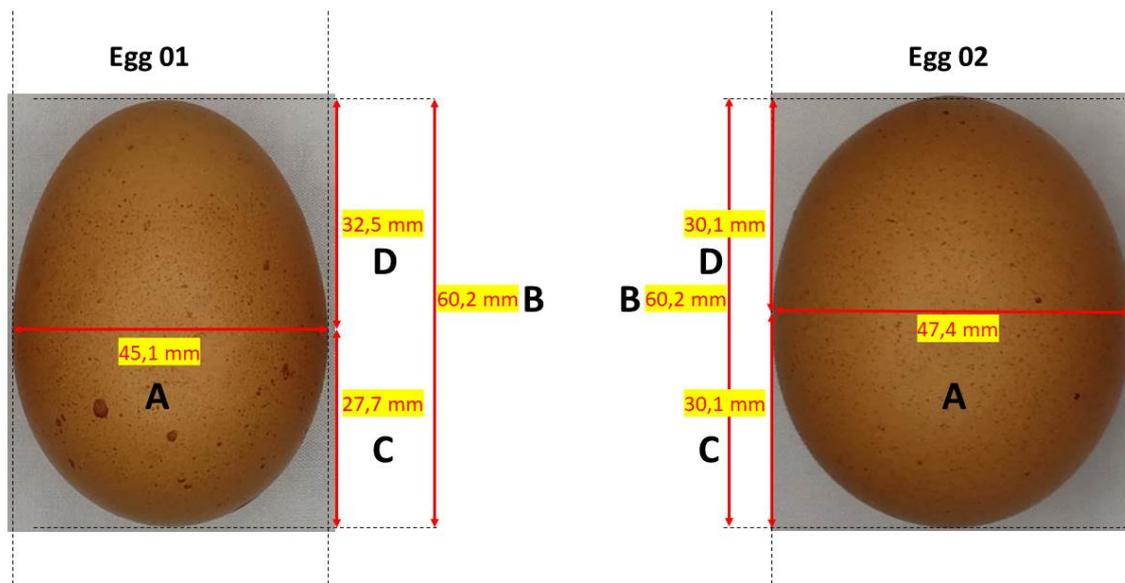


Figura 12 – Figura de exemplos que demonstram como serão as linhas encontradas para a divisão da metade do ovo, apresentada no início da aplicação

- D Pressionado para excluir a última linha construída
- P Pressionado para ativar o processamento de imagem
- C Pressionado para capturar uma nova imagem de uma câmera conectada
- F Pressionado para ler a imagem do arquivo
- X Pressionado para capturar a calibração do *pixFactor*
- V Pressionado para ativar o modo de vídeo
- S Pressionado para salvar a imagem em um arquivo

4.2 Captura da imagem

Para abrir uma imagem com uma bandeja de ovos, o usuário a seleciona na aplicação ao pressionar a tecla *F*. Em seguida, cria-se linhas e colunas para especificar quais ovos serão processados, pois é através delas que a aplicação conseguirá separar as imagens dos ovos para segmentação da imagem escolhida. Ao pressionar o botão esquerdo do *mouse*, uma linha é criada, e ao pressionar o botão direito do *mouse*, uma coluna é criada. O resultado da seleção dos ovos e a ordem em que foram escolhidos pode ser visualizado na Figura 13.

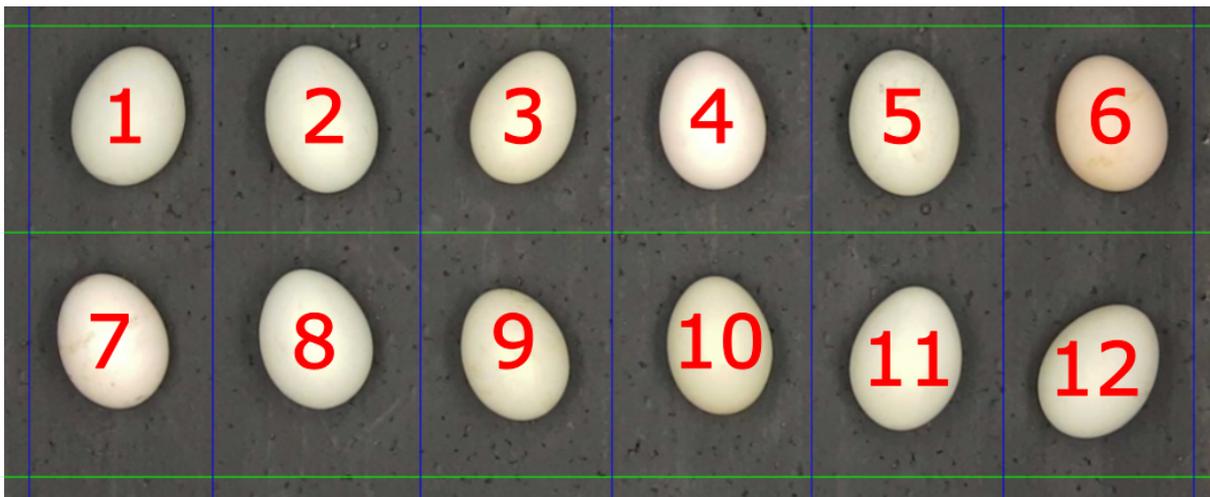


Figura 13 – Exemplo de seleção de ovos Bandeja 3 - Leitura 1: A aplicação sempre irá numerar os ovos em ordem crescente, de 1 para cima, conforme a quantidade selecionada

A aplicação iniciará o processamento com base no fator de *pixel* escolhido para a tela atual. O valor padrão do fator de *pixel* é definido próximo a 0.250417 . Esse valor afeta a precisão da medição dos volumes dos ovos e pode ser ajustado para se adequar às necessidades do usuário e à resolução da imagem.

4.3 Segmentação assistida dos ovos

No início do processamento da imagem selecionada, a aplicação realiza o redimensionamento da imagem para otimizar o tempo de processamento, o que resulta na redução de seu tamanho.

Em seguida, a imagem é convertida para o espaço de cores **HSV** para criar uma máscara binária que separa objetos ou regiões específicas do restante da imagem. Nessa máscara, os *pixels* que se enquadram nos intervalos de limite especificados são definidos como 1 (branco), o que denota sua inclusão no ovo, enquanto os *pixels* fora dos intervalos de limite são definidos como 0 (preto). A caixa delimitadora que contém o ovo é encontrada e os *pixels* dentro dela são ajustados para refinar a segmentação. Por fim, a segmentação modificada é retornada como resultado do processo. Cada etapa de transformação da imagem do ovo pode ser visto na Figura 14.

Essa abordagem de segmentação por máscara binária e refinamento da caixa delimitadora viabiliza uma identificação e isolamento mais precisos dos ovos na imagem, o que possibilita a análise mais precisa das bordas dos ovos selecionados.

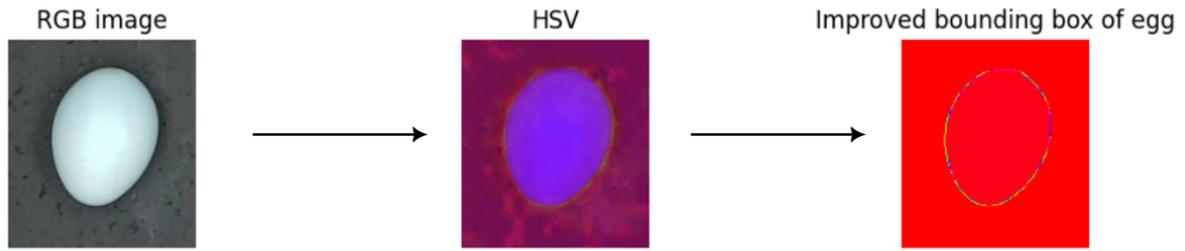


Figura 14 – Imagem do ovo número 2 em cada etapa de processamento : Na imagem esquerda, ele em seu formato padrão *RGB*. Na imagem do meio, ela foi transformada em *HSV*. Na imagem a direita é o resultado após aplicar a máscara e identificação da caixa delimitadora do ovo.

4.4 Detecção das bordas e cálculo das maiores distâncias

Uma vez segmentado é realizada a identificação dos pontos de borda para cada ovo que teve sua caixa aprimorada, a qual criará um novo conjunto P , formado pelos n pontos $P_i(x_i, y_i)$, onde n corresponde ao número total de pontos da borda do ovo e uma variação de 1 a n em i . Na Figura 15 pode ser visto um exemplo simplificado da borda de um ovo composta por 6 pontos após o processo de segmentação. A Figura 16 possui os pontos reais obtidos de um ovo.

Um novo conjunto Q é formado com todos os segmentos de reta $\overline{P_iP_j}$, para todos os pontos P_i e P_j do conjunto P , onde i e j variam de 1 a n e $i \neq j$. Na Figura 17, é apresentado um exemplo simplificado dos segmentos de reta formados entre os pontos que compõem a borda do ovo, onde o conjunto P seria formado por $\{\overline{P_1P_2}, \overline{P_1P_3}, \overline{P_1P_4}, \overline{P_1P_5}, \overline{P_1P_6}, \overline{P_2P_1}, \overline{P_2P_3}, \overline{P_2P_4}, \overline{P_2P_5}, \overline{P_2P_6}, \dots, \overline{P_6P_4}, \overline{P_6P_5}\}$.

A distância A , representada na Figura 18 pela distância entre os pontos P_1 e P_4 , foi dada pelo segmento $\overline{A_kA_p}$, sendo este o maior segmento de reta existente no conjunto Q , definido pelo cálculo da distância euclidiana conforme equação 4.1 (STEWART, 2006).

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4.1)$$

Já a distância B , representada na Figura 19, será dada pelo maior segmento $\overline{B_iB_j}$ dentre aqueles pertencentes ao conjunto Q que sejam perpendiculares à $\overline{A_kA_p}$.

4.5 Particionamento dos troncos de cone para cálculo do volume

Uma vez obtido o segmento $\overline{A_kA_p}$, um novo conjunto U será formado pelos m pontos U_i na forma (x, d) , onde:

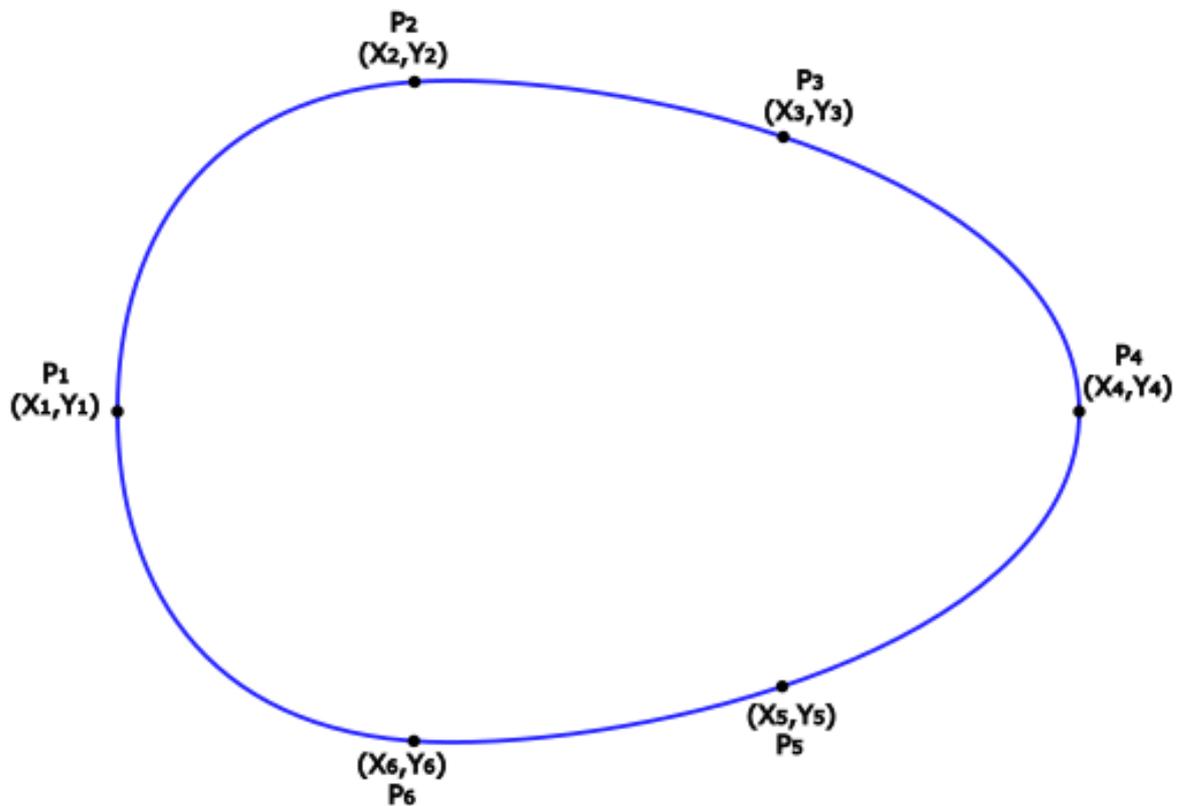


Figura 15 – Nesta figura, é apresentada uma representação visual dos pontos identificados na borda de um ovo, que compõe o conjunto $P = [P_1, P_2, P_3, P_4, P_5, P_6]$, com o objetivo de proporcionar uma melhor visualização. Cada ponto é representado pelas coordenadas (x, y) de um plano cartesiano.

- d será o tamanho de cada segmento de reta $\overline{Q_i Q_j}$, pertencente à Q e que seja perpendicular à $\overline{A_k A_p}$;
- x será a distância euclidiana entre o ponto $A_k(x_k, y_k)$ e o ponto $V_u(x_u, y_u)$ formado pela intersecção entre $\overline{A_k A_p}$ e $\overline{Q_i Q_j}$ para cada $\overline{Q_i Q_j}$ pertencente à Q que seja perpendicular à $\overline{A_k A_p}$;
- É importante destacar que os elementos $(0,0)$ e $(n, 0)$ farão parte de U , onde n representa o tamanho de $\overline{A_i A_f}$;

A representação gráfica do conjunto U em um plano cartesiano é mostrada na Figura 20.

Após a definição de U , os seus elementos são ordenados de forma crescente, com a consideração do valor de x de cada (x_i, d_i) presente em U .

Por considerar que o segmento de reta $\overline{A_i A_f}$ representa o eixo maior do ovo e que, conforme sua anatomia, qualquer corte sagital a este eixo irá compor um círculo perfeito.

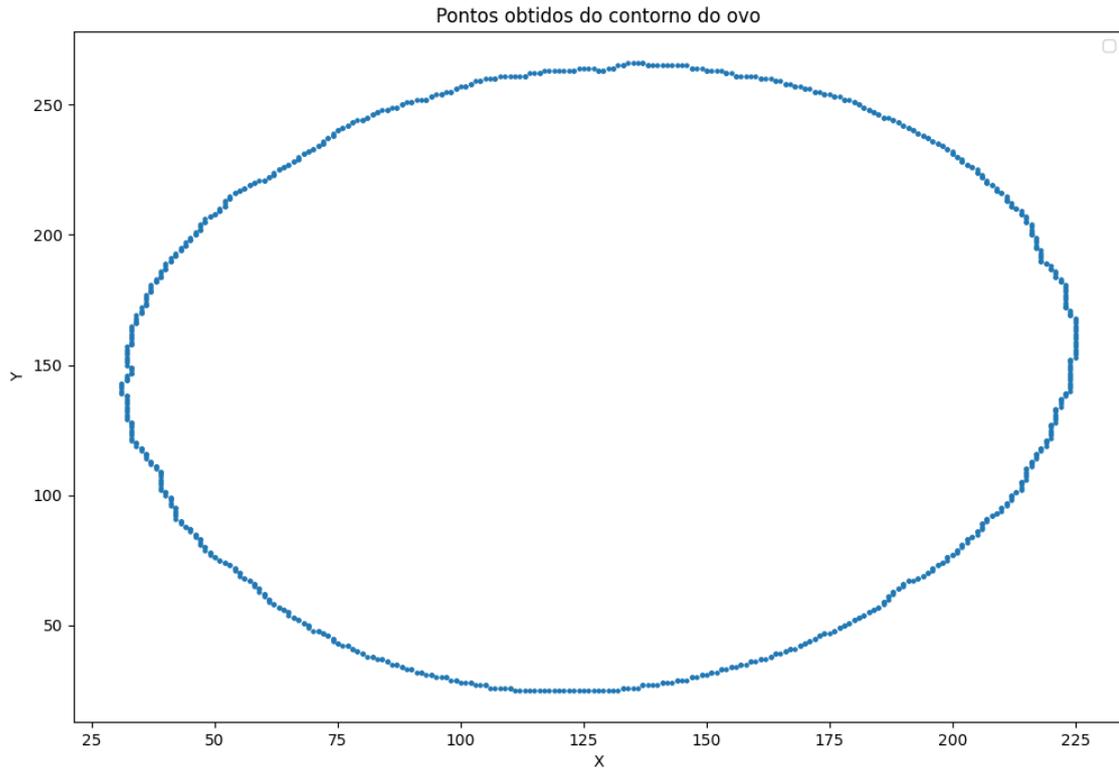


Figura 16 – Representação gráfica de todos os pontos obtidos do contorno segmentado do ovo 1.

Seu volume então é dado por uma integral circular, como definido na equação 4.2.

$$V = \int_0^\pi \int_1^n U \, dx \, d\theta \quad (4.2)$$

$$V = \frac{\pi h}{3} \cdot (R^2 + Rr + r^2) \quad (4.3)$$

Dada a natureza do trabalho anterior, os valores dos elementos do conjunto P são obtidos de forma discreta. Portanto, para o cálculo da equação 4.2 foi adotado um algoritmo baseado em métodos de integração numérica, mais especificamente no método do trapézio (PRESS et al., 2007), onde as formas utilizadas como referência para o cálculo da integral circular numérica foram troncos de cone, através da equação 4.3. Através da Figura 21 observa-se a criação e a utilização dos tais troncos de cone ao manusear a distância entre as retas dos pontos no conjunto K . O desenho de um trapézio é formado ao ligar as distâncias $\frac{D_1}{2}$, $\frac{D_2}{2}$, X_1 e X_2 , assim como também acontece na Figura 20 em que o trapézio formado, o que justificaria a utilização desse método.

A partir deste ponto foi identificado um problema: os valores de distância calculados dos pontos da borda apresentam certas variações, como pode ser visto na Figura 22, devido a possíveis fatores:

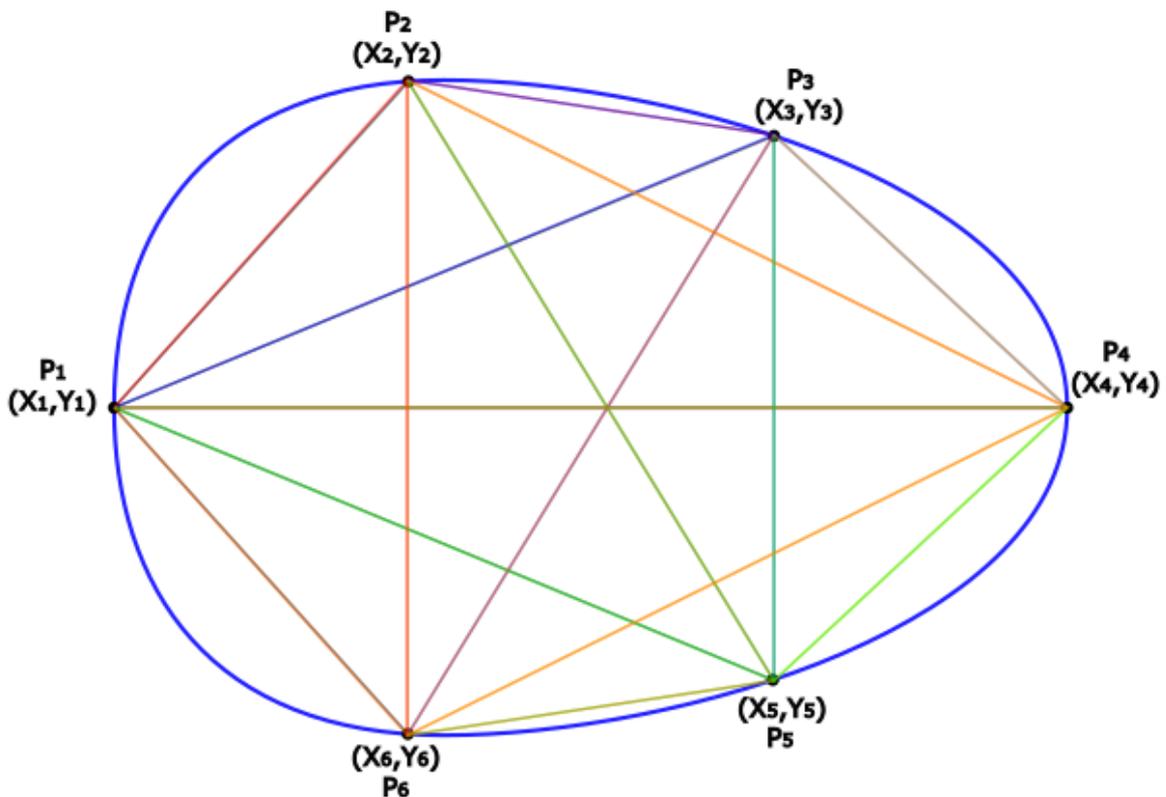


Figura 17 – Exemplo da Figura 15 com o conjunto Q de segmentos de reta, $\overline{P_i P_j}$, para todo $i \neq j$ com uma variação de 1 à 6 em i e j .

1. Erros de medição: pequenas imprecisões nas medições podem resultar em variações nos valores de distância.
2. A fórmula utilizada para calcular a distância euclidiana entre os pontos contém raízes quadradas e aritmética de pontos flutuantes, aos quais podem introduzir pequenos erros numéricos.
3. Ruídos na imagem: imperfeições na imagem capturada ou no processo de segmentação, que podem introduzir pequenos desvios nos pontos da borda e, então, prejudicar as medições de distância.
4. Variações na forma do ovo: a forma do ovo pode não ser perfeitamente simétrica, devido a deformações naturais que causam as variações nas distâncias calculadas em diferentes partes da borda.

O objetivo deste trabalho é aplicar um processo de ajuste polinomial para determinar qual curva melhor se ajustaria à nuvem dos pontos pertencentes ao conjunto K da Figura 22. O emprego do polinômio minimiza o efeito do ruído gerado na curva pelos processos de aquisição, segmentação, entre outros.

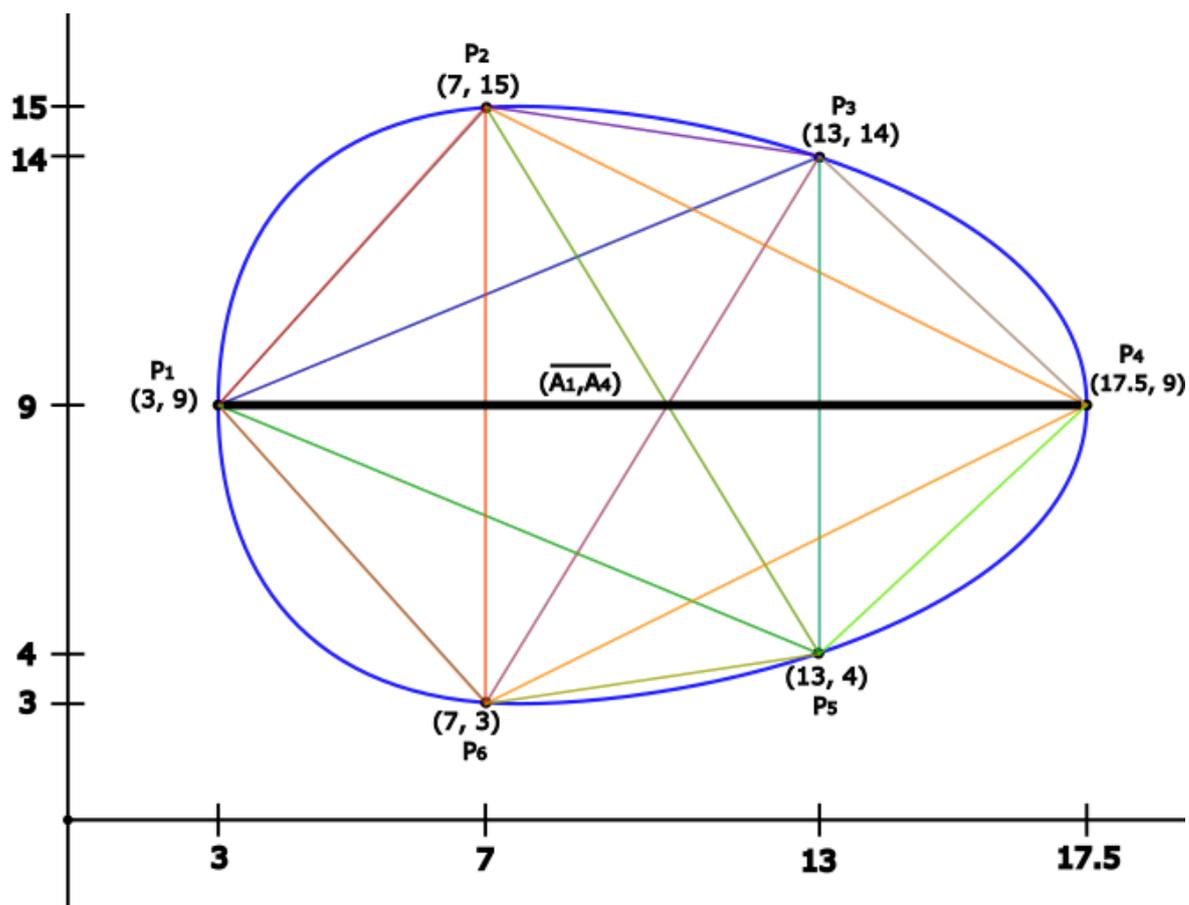


Figura 18 – Exemplo da Figura 17 ao implementar o ovo em um plano cartesiano, com valores definidos para os pontos do eixo X e Y . Através do cálculo da distância euclidiana, foi considerado o segmento de reta $\overline{P_1P_4}$ como a mais longa.

4.6 Ajuste de curva em cada particionamento

A abordagem utilizada neste trabalho para minimizar os efeitos desse ruído foi a de empregar técnicas de ajuste de curva (KIUSALAAS, 2013), como mencionado anteriormente. Ao ajustar uma curva à nuvem de pontos formada pelo valor dos diâmetros, a curva resultante será suavizada, com pontos mais alinhados, o que aumentará a precisão e representação do comportamento real dos dados (KIUSALAAS, 2013). Na Figura 23 a curva está sem ruídos.

O ajuste de curva tem como objetivo encontrar um modelo matemático que capture com precisão o comportamento dos dados, o que possibilitará a extração de informações relevantes e fazer previsões (KIUSALAAS, 2013). É importante ressaltar que é incomum que um modelo matemático corresponda precisamente aos dados experimentais devido a fatores como erros de medição e variações aleatórias.

Foi então criada a função *disc_slices_curve_fit* com a intenção de aplicar o ajuste de curva nos dados do conjunto K . Essa função foi implementada para suportar diferentes graus de polinômios, que permitem a realização do ajuste com maior flexibilidade. O

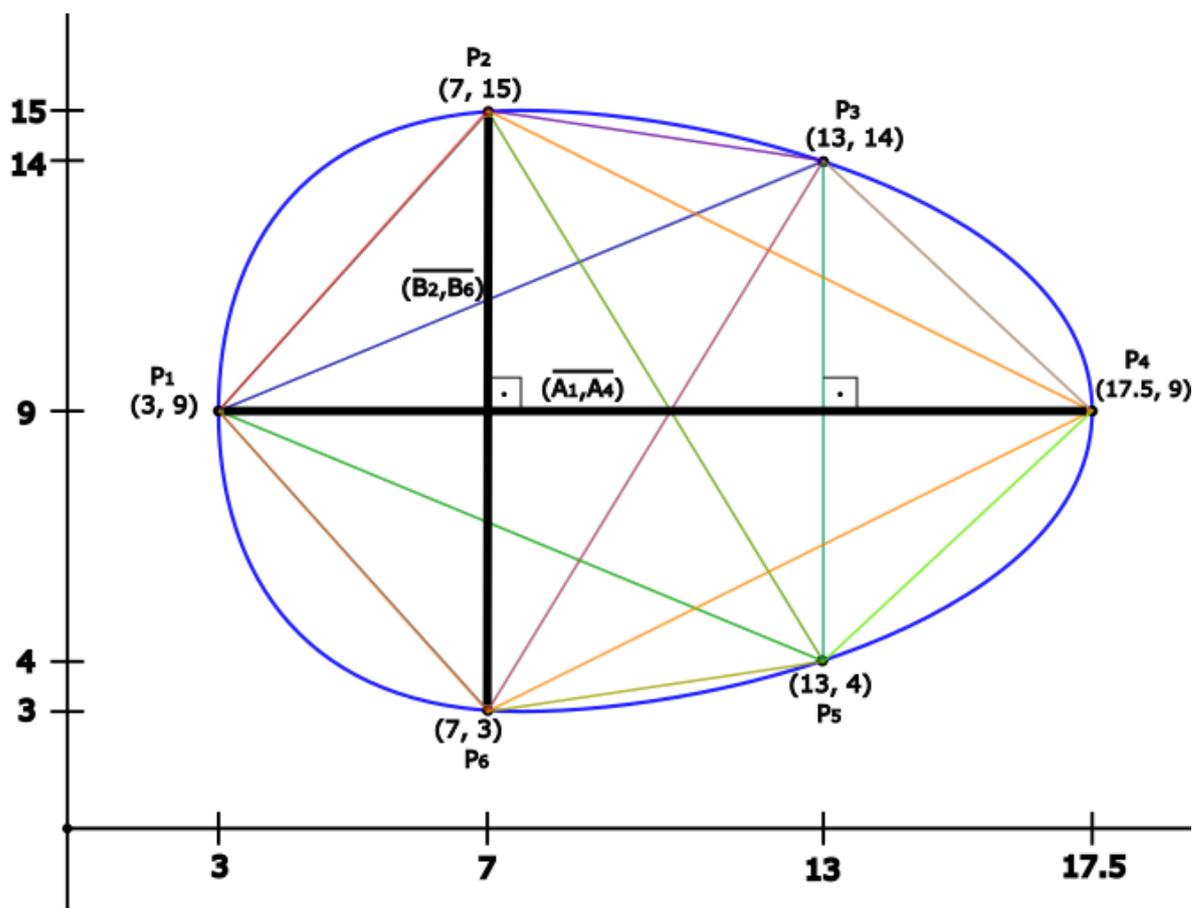


Figura 19 – Exemplo da Figura 18 após encontrar o maior segmento de reta, perpendicular a reta $\overline{A_1A_4}$. Apenas as retas $\overline{P_2P_6}$ e $\overline{P_3P_5}$ estavam aptas a serem escolhidas, sendo a $\overline{P_2P_6}$ com a maior distância euclidiana.

código desta função criada pode ser vista na Figura 24.

A função recebe como entrada os vetores $ydata$, que contém os tamanhos das retas d , e $xdata$, que abrange os pontos de interseção com o segmento de reta $\overline{A_iA_f}$, ambos obtidos do conjunto K de ovos. Em seguida, utiliza-se a biblioteca `scipy.optimize` e a função `curve_fit` para aplicar o ajuste de curva polinomial nos pontos do ovo, com diferentes graus de polinômios (3, 5, 7, 9 e 11). As funções polinomiais podem ser vistas na Figura 25.

A função usa a biblioteca `curve_fit` do módulo `scipy.optimize` (COMMUNITY, 2023) para realizar o ajuste de curva polinomial. Essa função tem como objetivo aplicar uma função polinomial pré-definida em nosso código a um conjunto de dados específico ($xdata$ e $ydata$). Ela utiliza o método dos mínimos quadrados não-lineares para ajustar uma curva aos dados, com a finalidade de encontrar os parâmetros da função que minimizam a soma dos quadrados dos resíduos entre os dados e a função polinomial escolhida (NUMPY, 2022b).

Ao considerar todos os aspectos mencionados anteriormente, foi decidido utilizar

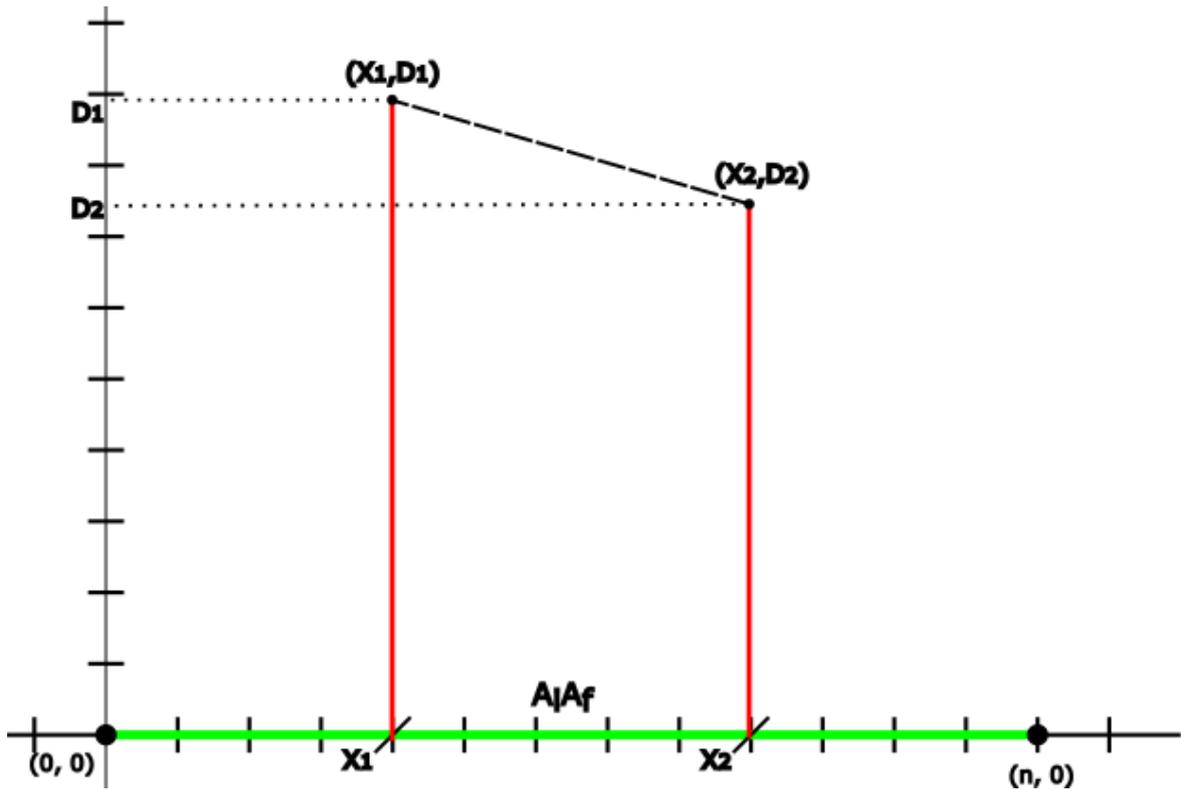


Figura 20 – Figura demonstra como são definidas o novo conjunto K , que possui o tamanho da reta (eixo Y) e a distância euclidiana entre $(0, 0)$ e o ponto onde a reta desta reta fará intersecção com a reta $\overline{A_i A_f}$. Os limites do eixo X serão definidos pelo tamanho da maior reta $\overline{A_i A_f}$ encontrada no ovo, que será o valor n .

equações polinomiais de forma geral nos graus 3, 5, 7, 9 e 11 para realizar o ajuste de curva. Na escolha do intervalo entre os graus dos ajustes polinomiais, optou-se por um salto de dois graus, com o intuito de minimizar o esforço computacional e também por ser uma diferença negligenciável entre ajustes consecutivos de grau 1. Essa escolha nos permite alcançar um equilíbrio entre a precisão do ajuste e a complexidade computacional.

A complexidade do ajuste aumentou consideravelmente ao ultrapassar o grau 11, o que resultou na percepção da aplicação apresentar-se mais lenta durante o seu uso e que impactaria de forma negativa a experiência do usuário, caso fosse necessário fazer dezenas de processamento de uma só vez. Portanto, foi decidido não utilizar graus superiores a 11 para garantir a eficiência e a usabilidade da aplicação.

Ressalte-se que escolha do grau da equação polinomial é um compromisso entre a capacidade de modelar adequadamente os dados e evitar o fenômeno chamado de sobreajuste (*overfitting*). O sobreajuste ocorre quando a função se ajusta muito bem aos dados de treinamento, mas não há uma boa generalização para novos dados. Portanto, é fundamental encontrar um equilíbrio que forneça um ajuste preciso aos dados existentes e permita previsões confiáveis para novos conjuntos de dados (NUMPY, 2022b).

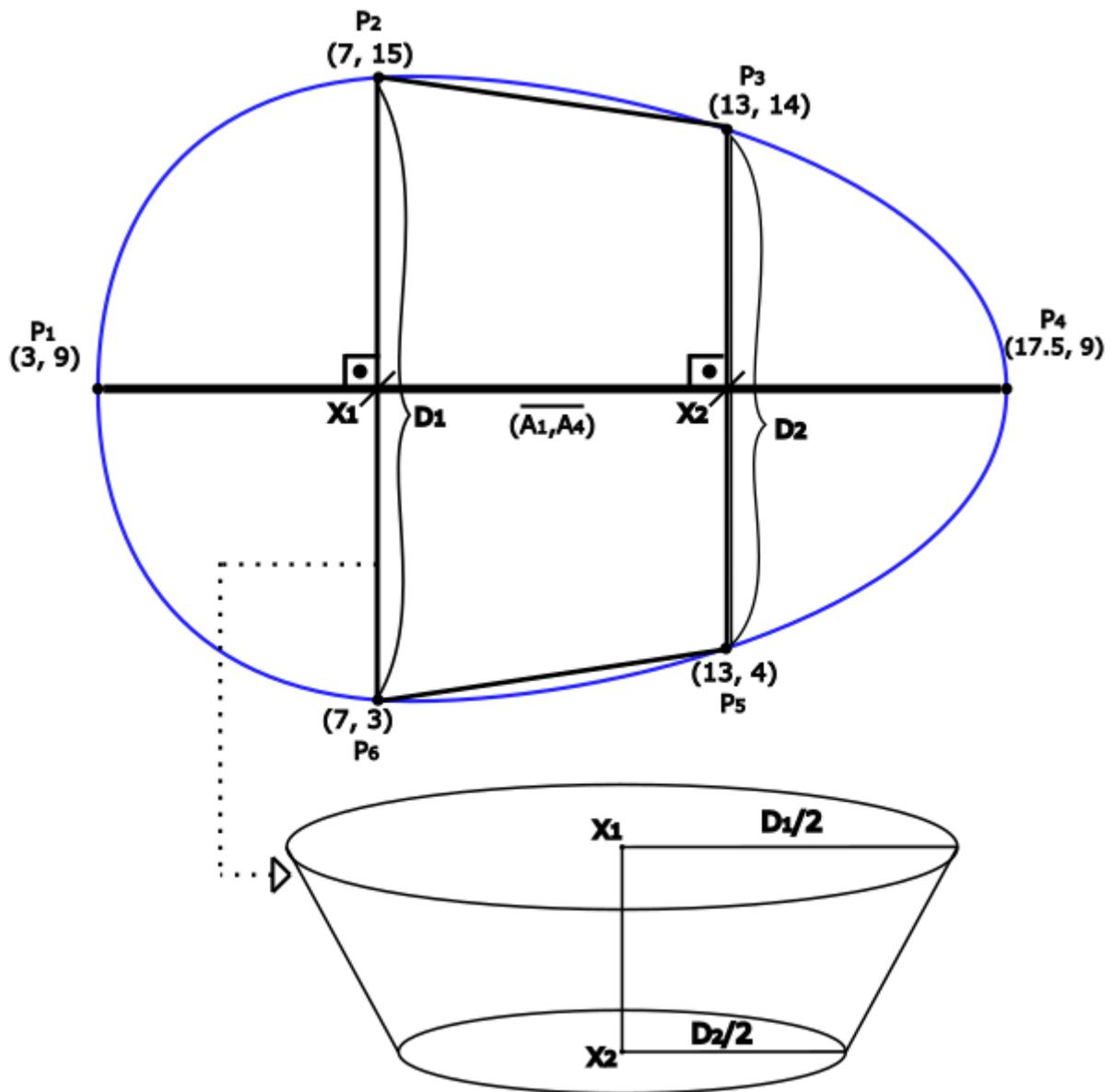


Figura 21 – Esta figura mostra a ilustração do cone com a utilização da distância que está uma reta da outra no eixo X , que corresponde a maior reta $\overline{A_i A_f}$, e a aplicação da distância euclidiana de cada reta para definir a distância do centro do cone a parte mais externa.

Além de calcular os parâmetros ajustados, ela utiliza o resultado de covariância retornado pela função *curve_fit* para criar a nova variável *perr*. A matriz de covariância contém informações sobre a incerteza dos parâmetros ajustados pelo método de mínimos quadrados. A partir dessa matriz, é possível extrair as estimativas de erro para cada parâmetro.

A variável *perr* é criada com base nos desvios padrão dos parâmetros, que correspondem à raiz quadrada dos elementos diagonais da matriz de covariância para obter seus valores absolutos.

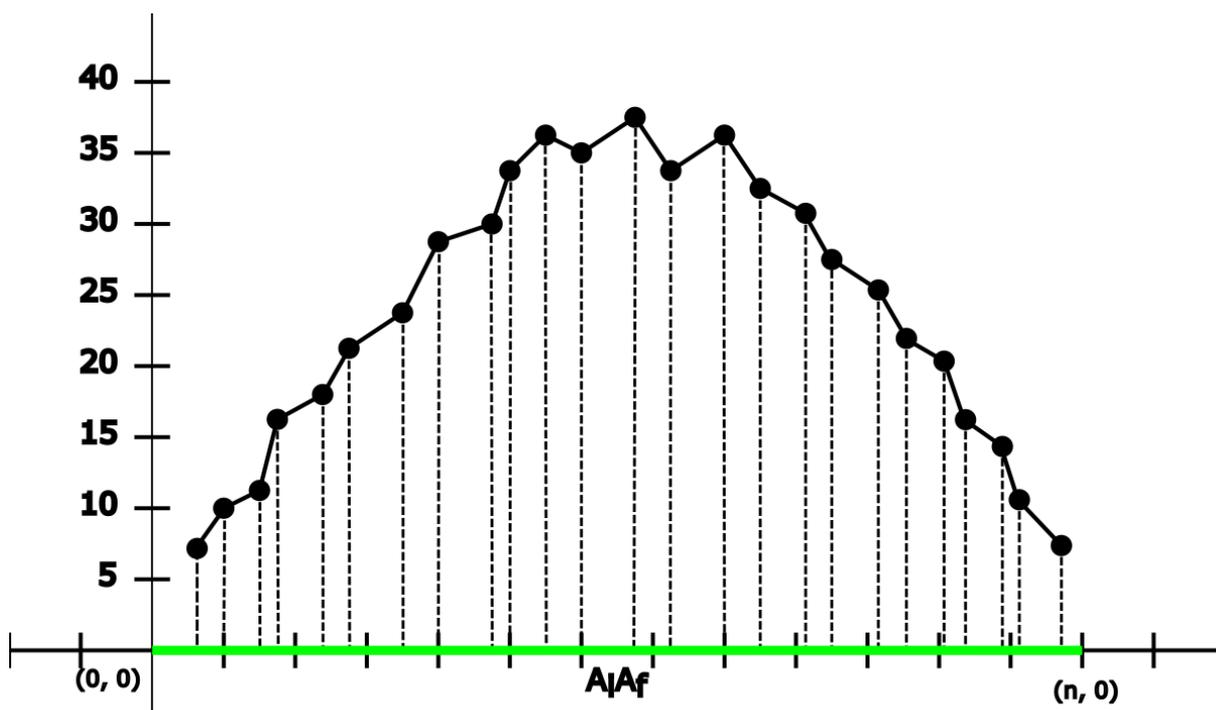


Figura 22 – Demonstração de um conjunto K com maior número de pontos, e com o tamanho das retas definidas no eixo Y além da distância dos pontos destas retas no eixo X , que será definida pelo segmento de reta $\overline{A_i A_f}$. Esse gráfico é apenas representativa, sem utilizar valores verdadeiros.

A função *np.diag* é aplicada à matriz de covariâncias retornada pelo ajuste de curva para extrair os elementos diagonais (NUMPY, 2022a), que correspondem às variâncias dos parâmetros ajustados. Ao utilizar *np.diag* nessa matriz, será obtido uma representação direta das variâncias, que indica a dispersão dos valores ajustados em relação à média. Dessa forma, pode-se analisar a precisão dos parâmetros e avaliar a confiabilidade dos resultados obtidos pelo ajuste de curva.

A raiz quadrada é aplicada à variância para obter o desvio padrão, que é uma medida de dispersão dos valores em relação à média.

Assim, a matriz de covariância é utilizada para fornecer informações sobre a precisão dos parâmetros ajustados e permitir uma análise mais robusta e confiável dos resultados obtidos pelo ajuste de curva polinomial.

Por fim, a função retorna os parâmetros ajustados para cada grau de polinômio com seus erros, sendo possível fazer a análise comparativa e a seleção do melhor ajuste com eles.

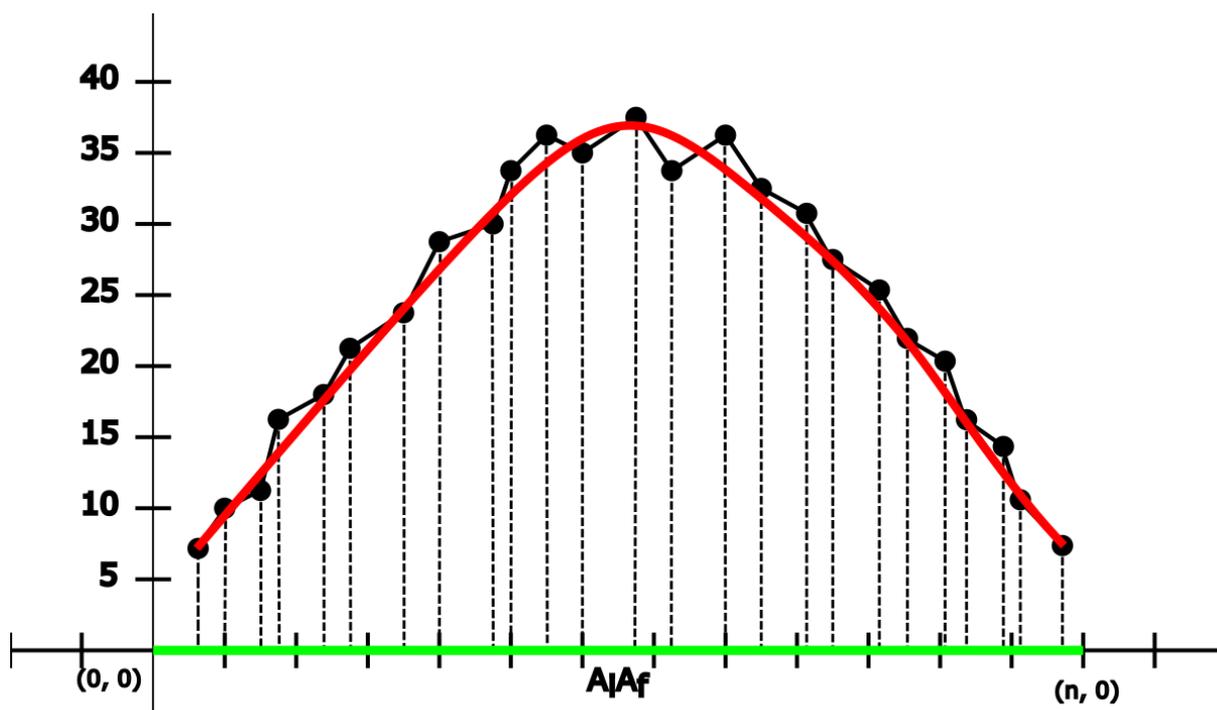


Figura 23 – Demonstração da aplicação de ajuste de curva nos pontos das retas do conjunto K , dados pelo exemplo da Figura 22. Novos pontos de tamanhos das retas serão dados para cada ponto X de cada tamanho de reta que foi dado anterior. A linha vermelha demonstra como será o alinhamento dos novos pontos, em comparação com a linha preta que demonstra a ligação entre os pontos antigos. Esse gráfico é apenas representativa, sem utilizar valores verdadeiros.

4.7 Geração de relatórios e imagens com resultados

Com os resultados em mãos, foi decidido criar uma nova função para gerar novas planilhas no formato do *Excel* (.xlsx) para futuras análises externas, que serão feitas fora do código.

A função `create_sheet` é responsável por criar uma planilha *Excel*, através da biblioteca `openpyxl` (GAZONI, 2023), que armazena os resultados de volume e área obtidos a partir do ajuste de curva. Ela recebe como argumentos o caminho do arquivo da planilha, o nome do ovo, os resultados de volume, bem como os erros de ajuste de curva.

Primeiro, a função verifica se o argumento `path_file_name` é uma instância do tipo `Path` (da biblioteca `pathlib` (FOUNDATION, 2023)). Em seguida, ela define o nome da planilha a ser criada com base no diretório pai do arquivo e o nome do ovo. A função também verifica se o arquivo da planilha já existe. Se existir, o arquivo é carregado; caso contrário, uma nova planilha é criada.

A função percorre as planilhas existentes no arquivo e verifica se a planilha correspondente ao nome do ovo já existe. Se não existir, uma nova planilha é criada com os

```

def disc_slices_curve_fit(x_data, y_data, polynomial_fit_degree_functions):
    resultDiscSlices = {}
    resultCurveError = {}
    for poly_fit_function in polynomial_fit_degree_functions:
        # Criar a curva de ajuste polinomial do grau escolhido na lista
        optimized_parameters, covariance = curve_fit(poly_fit_function, x_data, y_data)

        y_data_fit = poly_fit_function(x_data, *optimized_parameters)

        # plotando o gráfico do ajuste com os os valores de coeficiente otimizados
        plt.cla()
        plt.plot(x_data, y_data, 'b-', label='Sem ajuste (antigo)')
        plt.plot(x_data, y_data_fit, 'r-', label=f'Com ajuste: {poly_fit_function.__name__}')
        plt.suptitle(f'{{_path_file_name.name}} - Ovo {egg_num}', fontweight='bold')
        plt.title(f'Comparação: Sem ajuste X C/ ajuste ({{poly_fit_function.__name__}})')
        plt.xlabel("Distância X")
        plt.ylabel("Pontos de distância Y")
        plt.legend()

        plt.savefig(Path(egg_folder_fit_plot_path, f'plot_{{poly_fit_function.__name__}}'))

        # erro do ajuste polinomial
        perr = np.sqrt(np.diag(covariance))
        print(f'Error of the curve fit - {poly_fit_function.__name__}: {perr}\n')

        # Convert lists (x_data, y_data_fit) to dictionary and append it to the result list as tuple
        resultDiscSlices[poly_fit_function] = dict(zip(x_data, y_data_fit))
        resultCurveError[poly_fit_function] = perr

    return [resultDiscSlices, resultCurveError]

```

Carlos Humberto de Paula Borges, last month

Figura 24 – Função `disc_slices_curve_fit`: Utilizada para aplicar o ajuste de curva no `xdata` e `ydata` obtidos

```

def polynomial_curv_3(x, a, b, c, d):
    return a*x**3 + b*x**2 + c*x + d

def polynomial_curv_5(x, a, b, c, d, e, f):
    return a*x**5 + b*x**4 + c*x**3 + d*x**2 + e*x + f

def polynomial_curv_7(x, a, b, c, d, e, f, g, h):
    return a*x**7 + b*x**6 + c*x**5 + d*x**4 + e*x**3 + f*x**2 + g*x + h

def polynomial_curv_9(x, a, b, c, d, e, f, g, h, i, j):
    return a*x**9 + b*x**8 + c*x**7 + d*x**6 + e*x**5 + f*x**4 + g*x**3 + h*x**2 + i*x + j

def polynomial_curv_11(x, a, b, c, d, e, f, g, h, i, j, k, l):
    return a*x**11 + b*x**10 + c*x**9 + d*x**8 + e*x**7 + f*x**6 + g*x**5 + h*x**4 + i*x**3 +
    j*x**2 + k*x + l

```

Figura 25 – Funções polinomiais em forma de código. A função deve receber as coordenadas x e y de seus pontos de dados como entradas e retornar os valores y previstos para cada valor x

cabeçalhos adequados para armazenar os resultados de volume e área, bem como os erros para cada grau de polinômio.

Em seguida, é realizada a busca ou criação da linha correspondente ao ovo na

planilha. Se a linha não existir, é adicionada uma nova linha com o nome do ovo.

A partir daí, os resultados de volume e área, bem como os erros de ajuste de curva, são inseridos nas células correspondentes da planilha. A variável *current_min_col* é utilizada para acompanhar a coluna atual na qual os valores são inseridos. Os valores do resultado sem ajuste de curva são inseridos nas colunas 2 e 3. Para os resultados com ajuste de curva, o volume é inserido na coluna atual, a área é inserida na coluna seguinte e os erros são inseridos na coluna seguinte a essa. Por fim, a planilha é salva no caminho especificado.

Com a planilha criada, a aplicação prossegue com as etapas anteriores, que incluem a geração de imagens dos ovos com suas áreas de cálculo identificadas, bem como as linhas de partição sobrepostas a elas. Além disso, a aplicação também cria um arquivo de texto que contém os resultados obtidos, os quais podem ser utilizados para fins externos à própria aplicação e possibilitará análises e processamentos adicionais. Essas funcionalidades adicionais agregam valor ao trabalho, além de permitir uma visualização mais completa e fornecer dados de saída para utilização em outros contextos.

5 Resultados e Discussão

5.1 Máquina e dados utilizados

Para realizar os testes da aplicação foi utilizado um computador com sistema operacional Windows 11, com um processador Intel(R) Core(TM) i7-8550U, 20gb de memória RAM DDR4 e uma placa de vídeo NVIDIA GeForce MX150, a qual não foi utilizada para fazer o processamento das imagens.

Neste estudo, foram utilizadas fotografias de três bandejas de ovos, com um total de 60 ovos. A primeira e a segunda bandeja contêm 24 ovos cada, enquanto a terceira bandeja contém apenas a metade dos ovos das bandejas anteriores. Para cada bandeja, foram capturadas quatro fotografias, todas com o mesmo tamanho e distância, que apresentarão os ovos em diferentes posições.

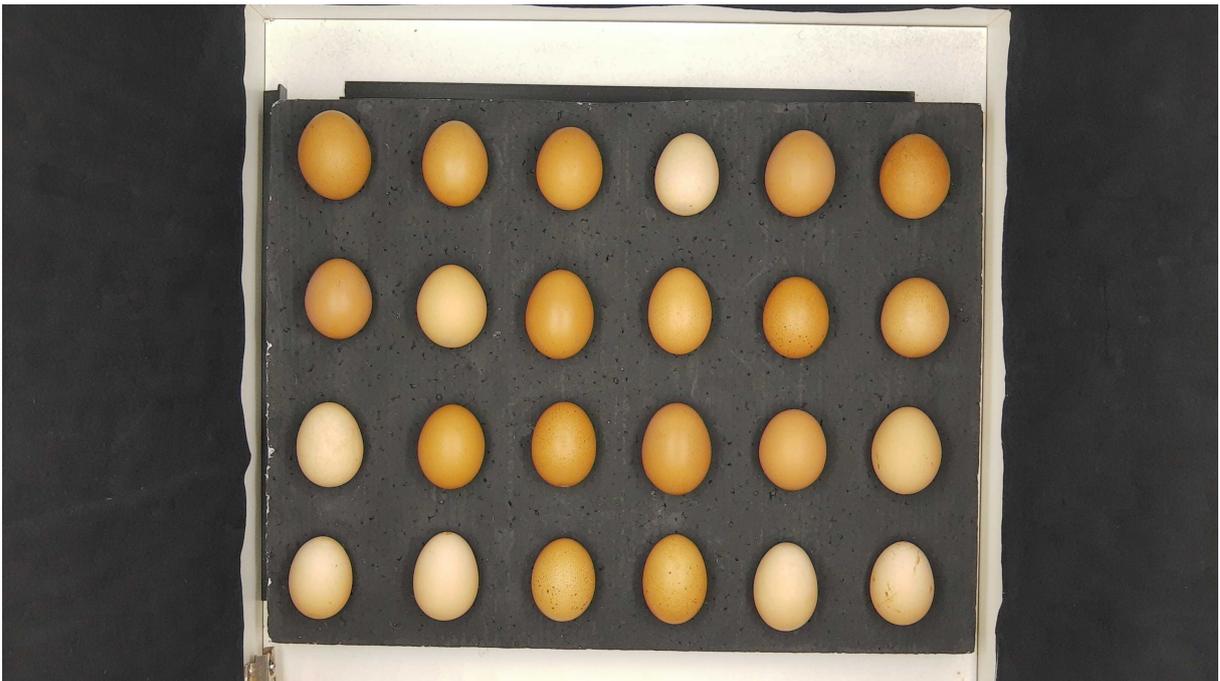


Figura 26 – Exemplo da foto da Bandeja 1, onde os ovos são numerados da posição 1 a 24, da ordem esquerda-superior para direita-inferior. Todos os ovos estão alinhados verticalmente

O objetivo de realizar quatro leituras em cada bandeja é avaliar a precisão da técnica de cálculo digital do volume do ovo. Busca-se obter resultados consistentes entre as quatro leituras, com variações mínimas no volume das metades selecionadas. A inversão das leituras tem o propósito de capturar o contorno da outra metade do ovo e realizar o cálculo correspondente para obter uma estimativa completa do volume do ovo.

5.2 Gráficos de distância de pontos com e sem ajuste de curva

Foram gerados gráficos comparativos das distâncias euclidianas e perpendiculares obtidas para os 60 ovos analisados. O eixo x é representado como *Distância X*, que são os valores obtidos de *distP1*, enquanto que o *Pontos de Distância Y* representa os valores da variável *distance*. A variação gráfica dos pontos obtidos por meio do ajuste de curva foi essencialmente a mesma para todos os ovos. Portanto, como exemplo representativo, será apresentado um único gráfico do ovo número 8 para destacar a utilidade do ajuste de curva aplicado aos resultados.

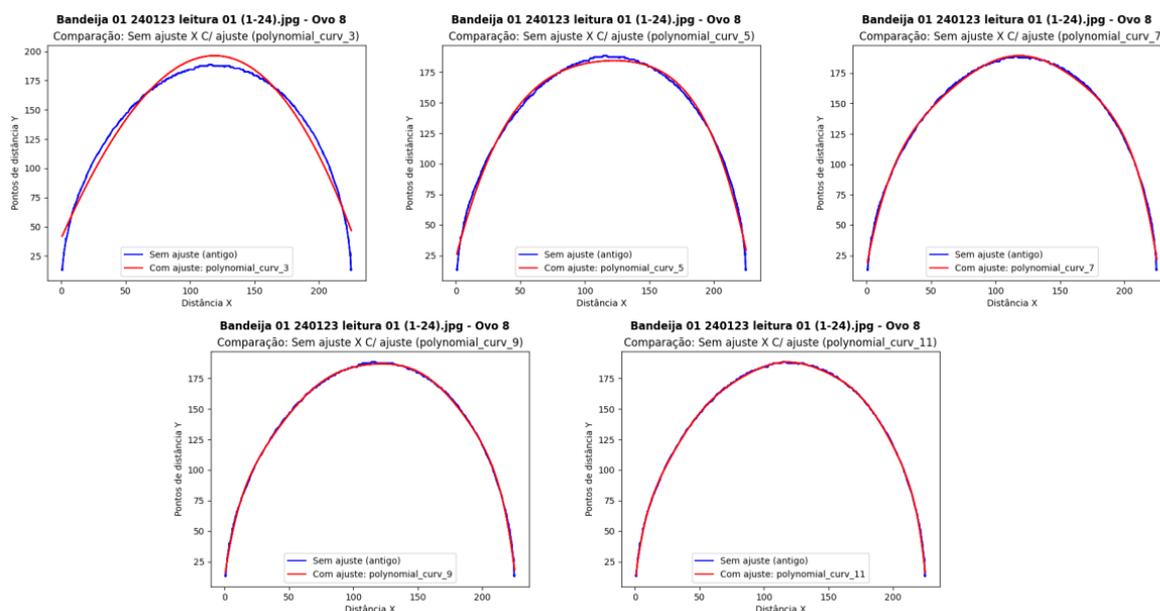


Figura 27 – Ovo 8 - Gráficos de Ajuste de Curva Polinomial pelo Método dos Quadrados Mínimos - Do gráfico no canto esquerda-superior para o canto direita-inferior : Grau 3, 5, 7, 9 e 11.

Ao analisar os gráficos da Figura 27, é evidente que o ajuste de curva polinomial de grau 11 apresentou uma maior proximidade entre os pontos calculados e os pontos obtidos sem ajuste. Essa tendência se mantém consistente para todos os ovos, independentemente da leitura realizada.

5.3 Resultado de volumes

Foram obtidos os volumes de cada um dos 60 ovos por meio de medidas físicas reais, através do método de deslocamento de Arquimedes com provetas, os quais serão denominados como *Volume Real* na aplicação. Com o verdadeiro volume de cada ovo, pode-se calcular os erros para cada tipo de volume resultante. Esses erros são calculados da seguinte forma:

$$ErroSemAjuste = \sqrt{(VolumeRealOvo * VolumeSoftwareOvo)^2}$$

$$ErroPolinomio = \sqrt{(VolumeRealOvo * \frac{VolumeMedioLeituras}{10})^2}$$

Com base nos testes realizados, obtiveram-se 5 tipos diferentes de volumes para cada um dos 60 ovos: sem ajuste, Grau 3, 5, 7, 9 e 11. Para uma melhor análise dos resultados desses testes, foram calculadas duas médias: o Erro Médio Quadrático Absoluto, ao utilizar os erros calculados com as fórmulas de *ErroSemAjuste* e *ErroPolinomio*, e o Erro Médio Quadrático Relativo que pode ser calculado através do erro relativo obtido na fórmula abaixo :

$$ErroQuadraticoRelativo = \frac{ErroComOuSemAjuste}{VolumeRealOvo}$$

Tabela 2 – Erro médio quadrático (Relativo)

	Funções de Ajuste de Curva					
	Sem Ajuste	Pol. Grau 3	Pol. Grau 5	Pol. Grau 7	Pol. Grau 9	Pol. Grau 11
Erro Médio	0,08411400	0,08490796	0,07692743	0,07899440	0,07873288	0,07873612
Variância	0,00170706	0,00109843	0,00103041	0,00105002	0,00105702	0,00105899

A Tabela 2 apresenta os valores do erro médio quadrático relativos e da variância para diferentes funções de ajuste de curva. Segue a explicação dos elementos da tabela:

1. A primeira linha abaixo dos cabeçalhos apresenta os valores correspondentes ao erro médio quadrático relativo. Essa medida representa a média dos erros ao quadrado, sem levar em consideração o sinal dos erros. Nesse contexto, ela indica a diferença relativa entre os volumes calculados e os volumes reais dos ovos. Os valores são apresentados com uma precisão de quatro casas decimais e utilizarão a vírgula como separador decimal.
2. A segunda linha abaixo dos cabeçalhos apresenta os valores correspondentes à variância dos erros. A variância é uma medida estatística que indica a dispersão dos valores em relação à média; nesse contexto, representa a variabilidade dos erros em relação aos volumes reais dos ovos. Assim como o erro médio, os valores também são apresentados com uma precisão de quatro casas decimais e utilizarão a vírgula como separador decimal.

Tabela 3 – Correlação entre volume real e volume da aplicação

Correlações					
Sem Ajuste	Pol. Grau 3	Pol. Grau 5	Pol. Grau 7	Pol. Grau 9	Pol. Grau 11
0,9116	0,9221	0,9235	0,9249	0,9246	0,9251

A Tabela 3 descreve os valores de correlação entre o volume verdadeiro dos ovos e o volume calculado pela aplicação para diferentes funções de ajuste de curva, em que

a correlação é uma medida estatística que indica a força e a direção da relação linear entre duas variáveis. Nesse contexto, a correlação representa o desempenho dos volumes calculados pela aplicação que se relacionam com os volumes verdadeiros dos ovos. O valor da correlação varia de -1 a 1, em que 1 indica uma correlação positiva perfeita, 0 indica ausência de correlação e -1 indica uma correlação negativa perfeita. Os valores da tabela são apresentados com uma precisão de quatro casas decimais e utilizará a vírgula como separador decimal.

Para se obter uma análise mais detalhada do cálculo individual de volume e erro de cada ovo, foi criado o gráfico da Figura 28. Neste gráfico, alguns ovos apresentaram valores de erro que se distanciaram da curva estabelecida pelos ajustes polinomiais. Idealmente, esperava-se que esses valores estivessem mais próximos dos erros absolutos e relativos, obtidos através dos ajustes polinomiais.

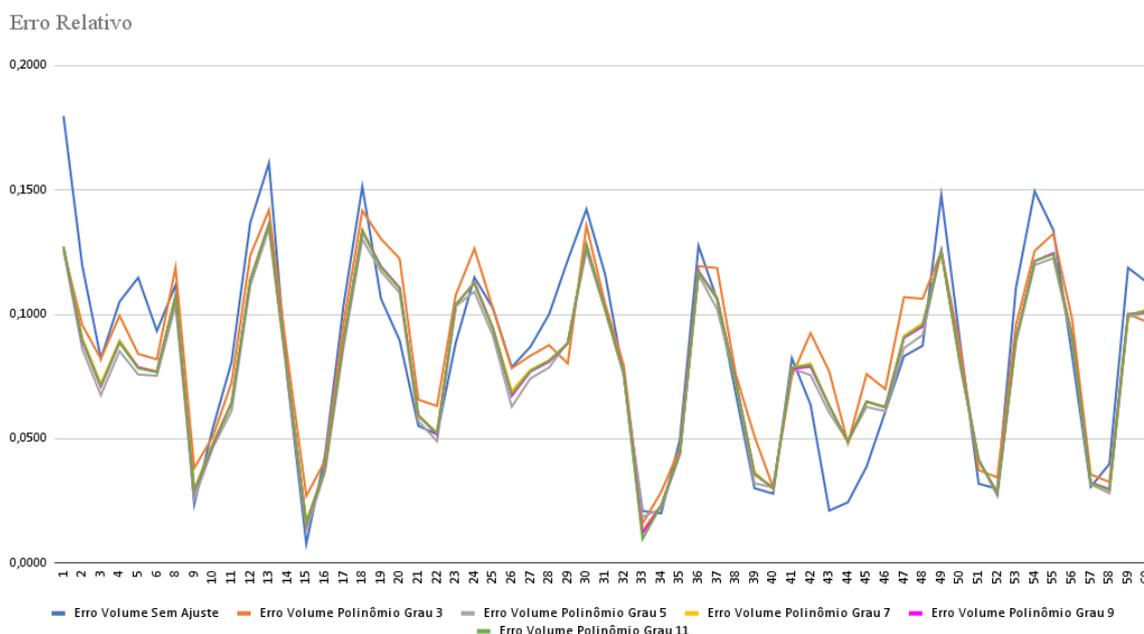


Figura 28 – Gráfico de erro relativo de cada ovo : Eixo X é a numeração dos ovos, enquanto que o Eixo Y é o valor de erro relativo do volume calculado dos ovos

Os ovos identificados que se destacaram, ou seja, que apresentaram valores de erro absoluto e relativo fora do padrão estabelecido, foram os de números 19, 42, 43, 44 e 45. Essa divergência pode indicar a presença de fatores adicionais que não foram considerados pelo modelo polinomial utilizado, como falhas ou maiores ruídos no processamento das imagens, os quais podem influenciar no cálculo do volume desses ovos específicos.

Ao analisar as Figuras 29 e 30, observa-se que os dados obtidos para o ovo 1 nas quatro leituras apresentaram uma variação visualmente perceptível, a ponto de afetar consideravelmente os pontos calculados pela aplicação de extração de pontos euclidianos

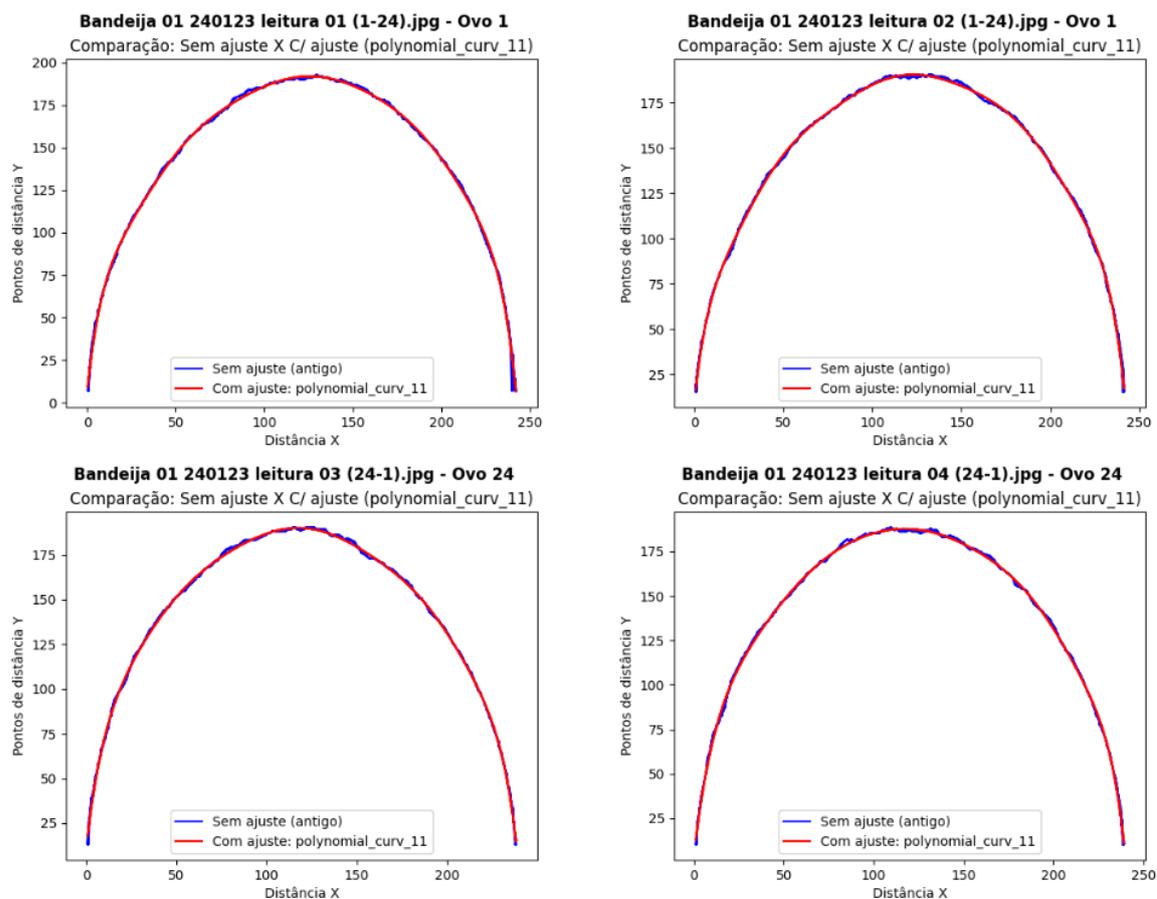


Figura 29 – Ovo 1 - Distância de pontos sem ajuste e com ajuste polinomial de grau 11 da leitura 1 a 4.

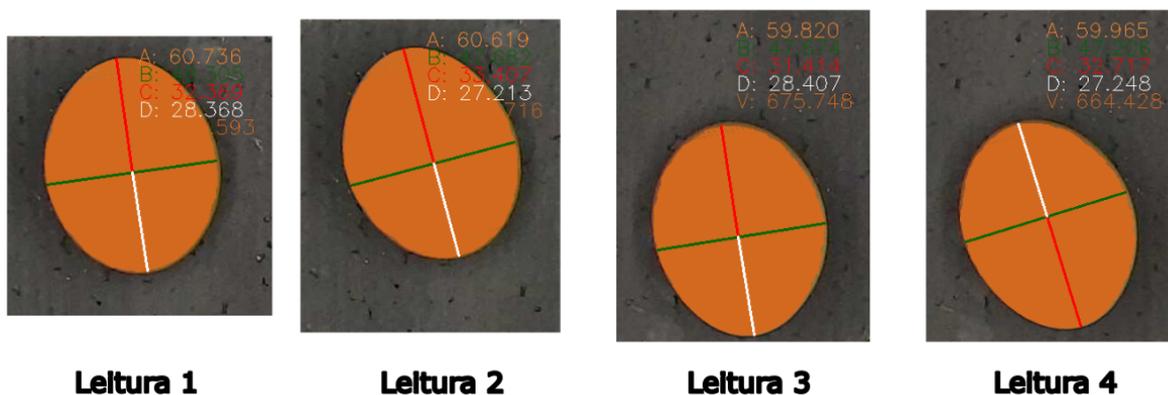


Figura 30 – Ovo 1 - *Pixels* do ovo que foram identificados como área interna, através dos seus pontos, da leitura 1 a 4.

e perpendiculares. No entanto, foi possível observar que o ajuste de curva polinomial de grau 11 foi capaz de suavizar esses pontos ao alinhá-los de forma mais consistente com os pontos base da borda do ovo obtidos pela aplicação.

Essa tendência de suavização dos pontos por meio do ajuste de curva também foi

observada em outros ovos que apresentaram diferenças significativas nos erros médios de seus ajustes. Isso evidencia a capacidade do ajuste de curva de grau 11 em harmonizar e aproximar os pontos calculados aos pontos de referência da borda do ovo, obtidos pela aplicação utilizada.

É importante ressaltar que a diferença nos erros médios observada nos ovos está diretamente relacionada à variabilidade dos dados coletados durante as leituras. Ao ocorrer uma maior dispersão dos pontos, como no caso do ovo 1, o ajuste de curva se torna essencial para mitigar essas variações e obter resultados mais consistentes e confiáveis.

Dessa forma, os resultados obtidos demonstram que o ajuste de curva polinomial de grau 11 desempenha um papel fundamental na redução das discrepâncias entre os pontos calculados e os pontos base da borda do ovo, ao proporcionar uma representação mais precisa da geometria do objeto estudado. Assim, reitera-se a importância do uso adequado de técnicas de ajuste de curva para melhorar a exatidão e a confiabilidade dos resultados em estudos que envolvam análises geométricas de objetos semelhantes.

5.4 Discussão

Com base nos testes realizados, ao analisar a Tabela 2, constatou-se que o erro médio quadrático relativo e a variância foram menores no cálculo do volume dos ovos do ajuste de curva com polinômio de grau 5.

Isso indica que, para esse conjunto de dados, a função de ajuste que utiliza um polinômio de grau 5 foi mais eficiente em reproduzir os volumes verdadeiros dos ovos, se considerar a relação relativa entre os erros. Os demais polinômios de graus 3, 7, 9 e 11 também tiveram valores de erro médio relativamente próximos e indicam uma performance semelhante entre eles. Contudo, o polinômio de grau 5 se destacou ao apresentar o menor erro médio relativo.

Já a variância indica que os resultados obtidos por meio desse método de ajuste foram mais consistentes e apresentaram menor dispersão em relação aos volumes verdadeiros dos ovos. Essa consistência sugere que os erros individuais tenderam a se aproximar uns dos outros e proporcionam maior previsibilidade e confiabilidade nos resultados obtidos.

No entanto, ao observar a tabela de correlação (Tabela 3), percebeu-se que o polinômio de grau 11 apresentou o maior valor de correlação. Essa correlação mais elevada indica que o modelo de ajuste com o polinômio de grau 11 foi mais adequado para capturar as características específicas de cada ovo, no que resulta em uma redução mais consistente das discrepâncias entre os volumes calculados e os verdadeiros volumes. Essa maior correlação é esperada, uma vez que o polinômio de grau 11 possui um desenho mais suave e próximo aos pontos verdadeiros, como observado nos gráficos do ovo 1 (29).

Portanto, com base nos resultados obtidos, pode-se concluir que o cálculo do volume dos ovos sem ajuste de curva apresentou o menor erro médio quadrático absoluto. No entanto, ao considerar o erro médio quadrático relativo e a variância dos erros, o polinômio de grau 5 foi a função de ajuste mais eficiente. Além disso, o polinômio de grau 11 demonstrou a maior correlação e menor erro total, o que indica uma melhor capacidade de reproduzir os volumes verdadeiros dos ovos de forma precisa. Essas conclusões são relevantes para a escolha da função de ajuste mais adequada na estimativa do volume dos ovos, quando se considera as diferentes métricas de avaliação.

5.4.1 Gráfico de Erro Relativo

Durante a análise do gráfico 28 dos ovos individuais, foi observado que os ovos 19, 42, 43, 44 e 45 se desviaram da curva esperada. Uma inspeção visual das imagens desses ovos revelou que o processamento das leituras apresentava mais erros, que resultou em distâncias euclidianas menores para os pontos sem ajuste em comparação com os pontos ajustados. Isso indica que os residuais de cada ponto no cálculo do ajuste foram predominantemente positivos, ao contrário do esperado, que era reduzir o volume do ovo para se aproximar do valor real.

6 Conclusões

O presente estudo abordou o problema da estimativa do volume de ovos por meio do uso de pontos com ruídos que definem as bordas das cascas. O objetivo foi desenvolver uma proposta que oferecesse uma abordagem eficiente e precisa para melhorar calcular o volume dos ovos de galinha, através do uso do ajuste de curva.

Os principais resultados obtidos indicaram que o uso de polinômios de grau 5 e 11 no ajuste de curva proporcionou melhores resultados. O polinômio de grau 5 demonstrou menor erro médio quadrático relativo e variância, o que indica maior eficiência em reproduzir os volumes verdadeiros dos ovos. Por outro lado, o polinômio de grau 11 apresentou maior correlação, sugerindo uma melhor capacidade de capturar as características específicas de cada ovo.

No entanto, algumas limitações foram identificadas no estudo. Foi observado que certos ovos apresentaram desvios em relação à curva esperada, indicando erros no processamento das leituras e na obtenção dos pontos de contorno. Essa limitação ressalta a importância de aprimorar a precisão do processamento de imagens e definir limites aceitáveis para o erro médio quadrático relativo do volume dos ovos em estudos futuros.

É válido ressaltar que este trabalho de conclusão de curso foi desenvolvido com base em uma pesquisa de mestrado em andamento, a qual não estabeleceu os limites aceitáveis para o erro médio quadrático relativo do volume dos ovos. Portanto, sugere-se que estudos futuros incluam a definição desses limites, bem como busquem aprimorar a precisão do processamento das imagens adquiridas, visando obter pontos de contorno com menor margem de erro. Além disso, a realização de validações experimentais adicionais é recomendada para verificar a eficácia dos métodos de estimativa utilizados.

Referências

- AYUPOV, F. On the egg mathematical model. *Advanced scientific-industrial experience in poultry breeding. Express Information*, v. 9, p. 14–6, 1976. Citado na página 34.
- CARTER, T. C. The hen's egg: Estimation of shell superficial area and egg volume, using measurements of fresh egg weight and shell length and breadth alone or in combination. *British Poultry Science*, Taylor & Francis, v. 16, n. 5, p. 541–543, 1975. Disponível em: <<https://doi.org/10.1080/00071667508416224>>. Citado na página 34.
- CHALIDABHONGSE, T.; YIMYAM, P.; SIRISOMBOON, P. 2d/3d vision-based mango's feature extraction and sorting. p. 1–6, 2006. Disponível em: <<https://doi.org/10.1109/ICARCV.2006.345248>>. Citado na página 13.
- CHAN, T. O. et al. An egg volume measurement system based on the microsoft kinect. *Sensors*, v. 18, n. 8, 2018. ISSN 1424-8220. Disponível em: <<https://doi.org/10.3390/s18082454>>. Citado na página 32.
- COMMUNITY, T. S. *scipy.optimize.curve_fit - SciPy v1.10.1 Manual*. 2023. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html>. Citado na página 44.
- FILHO, O.; NETO, H. *Processamento digital de imagens*. Brasport, 1999. ISBN 8574520098. Disponível em: <<https://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>>. Citado 6 vezes nas páginas 6, 16, 17, 18, 19 e 20.
- FOUNDATION, P. S. *pathlib — Object-oriented filesystem paths*. 2023. Disponível em: <<https://docs.python.org/3/library/pathlib.html>>. Citado na página 48.
- GAZONI, C. C. E. *openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files*. 2023. Disponível em: <<https://openpyxl.readthedocs.io/en/stable/>>. Citado na página 48.
- GILAT, A. *Numerical Methods for Engineers and Scientists, 3rd Edition: Third Edition*. Wiley Global Education, 2013. ISBN 9781118803042. Disponível em: <<https://www.wiley.com/en-us/Numerical+Methods+for+Engineers+and+Scientists:+An+Introduction+with+Applications+Using+MATLAB,+3rd+Edition-p-9781118554937>>. Citado 10 vezes nas páginas 6, 9, 14, 20, 21, 22, 23, 24, 25 e 26.
- GONI, S. M.; PURLIS, E.; SALVADORI, V. O. Three-dimensional reconstruction of irregular foodstuffs. *Journal of Food Engineering*, v. 82, n. 4, p. 536–547, 2007. ISSN 0260-8774. Disponível em: <<https://doi.org/10.1016/j.jfoodeng.2007.03.021>>. Citado na página 13.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. Pearson, 2018. ISBN 9780133356724. Disponível em: <<https://books.google.com.br/books?id=0F05vgAACAAJ>>. Citado 3 vezes nas páginas 6, 16 e 18.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, 9 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>. Citado 2 vezes nas páginas 26 e 27.

- HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. Citado 2 vezes nas páginas 28 e 29.
- KERN, M. D.; COWIE, R. J. The size and shape of eggs from a welsh population of pied flycatchers: Testing hoyt's use of egg dimensions to ascertain egg volume (tamaño y forma de los huevos de una población de ficedula hypoleuca: Poniendo a pruebas la fórmula de hoyt para determinar el volumen de los huevos). *Journal of Field Ornithology*, [Association of Field Ornithologists, Wiley], v. 67, n. 1, p. 72–81, 1996. ISSN 02738570, 15579263. Disponível em: <<http://www.jstor.org/stable/4514084>>. Citado na página 13.
- KIUSALAAS, J. *Numerical Methods in Engineering with Python 3*. Cambridge University Press, 2013. (Numerical Methods in Engineering with Python 3). ISBN 9781107033856. Disponível em: <<https://doi.org/10.1017/CBO9781139523899>>. Citado na página 43.
- KLETTE, R. *Concise Computer Vision - An Introduction into Theory and Algorithms*. Springer, 2014. 1-413 p. (Undergraduate Topics in Computer Science). ISBN 978-1-4471-6319-0. Disponível em: <<http://dx.doi.org/10.1007/978-1-4471-6320-6>>. Citado na página 13.
- LEE, D.-J. et al. Area and volume measurements of objects with irregular shapes using multiple silhouettes. *Optical Engineering*, SPIE, v. 45, n. 2, p. 027202, 2006. Disponível em: <<https://doi.org/10.1117/1.2166847>>. Citado na página 13.
- MORÉ, J. J. The levenberg-marquardt algorithm: Implementation and theory. In: WATSON, G. A. (Ed.). *Numerical Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978. p. 105–116. ISBN 978-3-540-35972-2. Citado na página 27.
- MORE, J. J.; GARBOW, B. S.; HILLSTROM, K. E. User guide for minpack-1. [in fortran]. 8 1980. Disponível em: <<https://www.osti.gov/biblio/6997568>>. Citado na página 27.
- NARUSHIN, V. Express method for determination of egg morphological parameters. In: . United Kingdom Branch, World's Poultry Science Association, 1994. (Proceedings 9th European Poultry Conference: Glasgow 7-12th August 1994, v. 2). ISBN 9780952360223. Disponível em: <https://www.temjournal.com/content/101/TEMJournalFebruary2021_171_176.pdf>. Citado na página 34.
- NARUSHIN, V. Non-destructive measurements of egg parameters and quality characteristics. *World's Poultry Science Journal*, Cambridge University Press on behalf of World's Poultry Science Association, v. 53, n. 2, p. 141–153, 1997. Disponível em: <<https://doi.org/10.1079/WPS19970014>>. Citado na página 34.
- NARUSHIN, V. Egg geometry calculation using the measurements of length and breadth. *Poultry Science*, v. 84, n. 3, p. 482–484, 2005. ISSN 0032-5791. Disponível em: <<https://doi.org/10.1093/ps/84.3.482>>. Citado 2 vezes nas páginas 13 e 34.
- NUMPY, D. *numpy.diag — NumPy v1.23 Manual*. 2022. Disponível em: <<https://numpy.org/doc/1.23/reference/generated/numpy.diag.html>>. Citado na página 47.
- NUMPY, D. *numpy.linalg.norm — NumPy v1.23 Manual*. 2022. Disponível em: <<https://numpy.org/doc/1.23/reference/generated/numpy.linalg.norm.html>>. Citado 2 vezes nas páginas 44 e 45.

OKINDA, C. et al. Egg volume estimation based on image processing and computer vision. *Journal of Food Engineering*, v. 283, p. 110041, 2020. ISSN 0260-8774. Disponível em: <<https://doi.org/10.1016/j.jfoodeng.2020.110041>>. Citado 2 vezes nas páginas 13 e 33.

PAGANELLI, C. V.; OLSZOWKA, A.; AR, A. The avian egg: Surface area, volume, and density. *The Condor*, American Ornithological Society, v. 76, n. 3, p. 319–325, 1974. ISSN 00105422, 19385129. Disponível em: <<https://doi.org/10.2307/1366345>>. Citado na página 34.

PRESS, W. H. et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3. ed. Cambridge University Press, 2007. ISBN 0521880688. Disponível em: <<https://dl.acm.org/doi/10.5555/1403886>>. Citado na página 41.

SEDGHI, M.; GHADERI, M. Digital analysis of egg surface area and volume: Effects of longitudinal axis, maximum breadth and weight. *Information Processing in Agriculture*, v. 10, n. 2, p. 229–239, 2023. ISSN 2214-3173. Disponível em: <<https://doi.org/10.1016/j.inpa.2022.01.003>>. Citado 2 vezes nas páginas 34 e 35.

SHEN, W.; WU, L.; TU, L. A curve fitting based image segmentation method. p. 71–75, 2009. Disponível em: <<https://doi.org/10.1109/IACSIT-SC.2009.86>>. Citado na página 35.

SISWANTORO, J.; PRABUWONO, A. S.; ABDULAH, A. Volume measurement of food product with irregular shape using computer vision and monte carlo method: A framework. *Procedia Technology*, v. 11, p. 764–770, 2013. ISSN 2212-0173. 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013. Disponível em: <<https://doi.org/10.1016/j.protcy.2013.12.256>>. Citado na página 13.

SOLTANI, M.; OMID, M.; ALIMARDANI, R. Egg volume prediction using machine vision technique based on pappus theorem and artificial neural network. *Journal of Food Science and Technology*, v. 52, n. 5, p. 3065–3071, May 2015. ISSN 0975-8402. Disponível em: <<https://doi.org/10.1007/s13197-014-1350-6>>. Citado na página 13.

STEWART, J. *Cálculo Volume 1*. [S.l.]: São Paulo: Pioneira Thomson Learning, 2006. Citado na página 39.

SZÉKELY, T.; KOZMA, J.; PITI, A. The volume of snowy plover eggs (el volumen de los huevos de charadrius alexandrinus). *Journal of Field Ornithology*, [Association of Field Ornithologists, Wiley], v. 65, n. 1, p. 60–64, 1994. ISSN 02738570, 15579263. Disponível em: <<http://www.jstor.org/stable/4513895>>. Citado na página 13.

UYAR, R.; ERDOĞDU, F. Potential use of 3-dimensional scanners for food process modeling. *Journal of Food Engineering*, v. 93, n. 3, p. 337–343, 2009. ISSN 0260-8774. Disponível em: <<https://doi.org/10.1016/j.jfoodeng.2009.01.034>>. Citado na página 13.

VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020. Disponível em: <<https://doi.org/10.1038/s41592-019-0686-2>>. Citado 2 vezes nas páginas 27 e 28.

WANG, R. et al. 3d point cloud deformation based on moving least squares and control curves. p. 338–341, 2013. Disponível em: <<https://doi.org/10.1049/cp.2013.1984>>. Citado na página 33.

ZHANG, W. et al. A novel method for measuring the volume and surface area of egg. *Journal of Food Engineering*, v. 170, p. 160–169, 2016. ISSN 0260-8774. Disponível em: <<https://doi.org/10.1016/j.jfoodeng.2015.08.025>>. Citado na página 13.