

APLICAÇÃO DA VISÃO COMPUTACIONAL PARA ANÁLISE DIMENSIONAL DE FIOS TREFILADOS

Guilherme Marim da Silva – UNIFAE – guilhermemarim@gmail.com

Wiliam Regone – UNIFAE – wiliamregone@fae.br

Resumo

A visão computacional é composta por ferramentas de processamento de imagens que auxiliam na extração de informações e reconhecimento de padrões de um objeto de interesse através de um conjunto hardware e software. O presente trabalho busca desenvolver um sistema aplicando tais ferramentas para processar e extrair informações de diâmetro de fios trefilados e compará-los com a medição convencional. O sistema foi desenvolvido utilizando a linguagem de programação Python, em sua terceira versão, e as bibliotecas OpenCV, PIL/Pillow e NumPy. O fluxo comum à diversas aplicações da visão computacional seguem as etapas de aquisição da imagem, pré-processamento, segmentação, extração de características, reconhecimento de padrões e resultados. Para validar a eficiência do sistema foram utilizados objetos conhecidos do nosso cotidiano. Após a validação, pôde-se obter valores semelhantes entre as medidas obtidas pelo sistema em relação com a medição convencional de fios trefilados de diferentes bitolas, o que demonstrou alto grau de aplicabilidade no controle de qualidade durante o processo de trefilação. Embora os resultados tenham sido satisfatórios, notou-se a existência de fatores que podem afetar a precisão das medidas e devem, portanto, ser compensados durante o processo de captura de imagem.

Palavras-chave

Visão computacional; Análise dimensional; Python; Fios trefilados; Processamento de imagens

1 Introdução

A evolução de ferramentas computacionais vem possibilitando cada vez mais o uso de novas ferramentas nas específicas áreas da engenharia, que auxiliam na automação de processos, velocidade no processamento de dados e reconhecimento de padrões (PEIXOTO et al., 2015). Em particular, a visão computacional é definida pela ciência que estuda e desenvolve tecnologias que permitem que máquinas processem imagens (BALLARD & BROWN, 1982), extraíndo informações e reconhecendo padrões do objeto de interesse através de um conjunto hardware e software (BARELLI, 2018).

Para viabilizar a visão computacional, podem ser utilizados hardwares como sensores e câmeras responsáveis pela captura da cena. O algoritmo tem a finalidade de fazer a aquisição dessa imagem, já capturada e convertida de um formato analógico para um formato digital,

fazer o processamento da informação, segmentar os objetos de interesse e extrair informações necessárias para análise ou classificação (RUDEK et al., 2001). Algumas das ferramentas de visão computacional podem ser aplicadas em diversas áreas, como na engenharia mecânica em processos de conformação.

Dentre os processos de conformação mecânica, a trefilação é um dos mais utilizados pela indústria. Esse processo industrial consiste na redução do diâmetro de um fio e respectivo aumento no comprimento do material. A partir da trefilação obtêm-se produtos como arames, barras finas, fios e tubos de pequeno diâmetro.

Com o auxílio da visão computacional, o presente trabalho busca criar um sistema para reconhecer e extrair informações dimensionais de fios de seção transversal, obtidos através de um processo de trefilação. Para isso, foi utilizada a linguagem de programação Python 3 e as bibliotecas OpenCV, PIL/Pillow e NumPy. As bibliotecas OpenCV e PIL/Pillow contêm centenas de classes e métodos para leitura, processamento de imagens, segmentação, extração de informações, análise de objetos e reconhecimento de padrões para o desenvolvimento da aplicação. A biblioteca NumPy possui ferramentas numéricas para executar operações matriciais.

1.1 Justificativa

Na literatura, as aplicações de visão computacional em processos de conformação mecânica são pouco usuais. Desta forma, este trabalho busca apresentar uma ferramenta inovadora que seja capaz de reconhecer e extrair informações dimensionais de fios trefilados, sendo uma alternativa no processo de controle de qualidade, que pode resultar em redução de custo de produção e aumento na precisão das medidas do objeto de estudo.

1.2 Objetivo geral

Aplicar os métodos da visão computacional para obtenção do diâmetro de fios trefilados e comparar com as medidas extraídas através de outras ferramentas.

1.3 Objetivos específicos

- Capturar a imagem de um fio trefilado;
- Realizar o pré-processamento da imagem;
- Aplicar binarização e operações morfológicas;
- Separar os objetos de interesse na figura;

- Extrair informações dimensionais dos objetos segmentados;
- Comparar os valores obtidos através da medição convencional com os dados do sistema de visão computacional;
- Utilizar controles estatísticos para melhor compreensão dos resultados.

2. Referencial Teórico

2.1 Características de uma imagem digital

Segundo Feliciano et al. (2005) e Barelli (2018), uma imagem é composta por uma matriz de linhas e colunas que recebe valores de pixel (*picture element*) inteiros positivos, indicando a intensidade de luz em cada posição (i,j) dessa matriz. O pixel é o menor ponto que forma uma imagem digital, e o conjunto deles distribuídos entre as linhas e colunas formam a imagem completa. Um pixel pode receber valores que variam de 0 a 255 de acordo com a sua intensidade luminosa, sendo 0 o nível mais baixo de luz, a cor preta, 255 o nível mais alto de luz, a cor branca, e os valores intermediários representando os tons da escala cinza.

As imagens podem ser classificadas como binárias, tons de cinza e coloridas. Para imagens binárias temos uma matriz bidimensional que armazena apenas valores de 0 ou 255 para cada pixel, ou seja, somente preto ou branco. As que possuem apenas tons de cinza seguem o mesmo princípio das binárias, porém, o pixel pode receber um valor que varia entre 0 a 255 (Figura 1) (BRADSKI & KAEHLER, 2008; BARELLI, 2018).

0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255
0	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255

Figura 1. Escala de cores em tons de cinza. Fonte: Autor.

A imagem colorida é constituída de três matrizes bidimensionais, também chamadas de canais, em que cada canal representa uma das cores, vermelho, verde e azul, e recebem variações de 0 a 255 para cada pixel (BRADSKI & KAEHLER, 2008; BARELLI, 2018). Esse

padrão é conhecido como RGB (*Red, Green, Blue*), embora também exista o padrão HSV (*Hue, Saturation, Value*), que permite um grande grau de separação entre a cor e a iluminação, sendo muito utilizado na segmentação de objetos em imagens. Seguindo o padrão RGB, as cores mais comuns são:

- Branco - (255,255,255);
- Azul - (0,0,255);
- Vermelho - (255,0,0);
- Verde - (0,255,0);
- Amarelo - (255,255,0);
- Magenta - (255,0,255);
- Ciano - (0,255,255);
- Preto - (0,0,0).

2.2 Ferramentas de captura de uma imagem

Os principais sensores utilizados na captura de imagens digitais são o CCD (*Charge Coupled Device*) e CMOS (*Complementary Metal-Oxide Semiconductor*) (RUDEK et al., 2001). De acordo com Barelli (2018), os sensores CCD usam uma matriz de capacitores fotossensíveis, também conhecidos por células, que permitem capturar a luz do ambiente, converter em uma imagem digital e armazená-la na memória da câmera.

Cada capacitor é responsável por receber uma intensidade luminosa que representará um pixel na imagem, e a partir dessa intensidade, um conversor analógico-digital (A/D) é usado para transformar o valor da tensão elétrica em um valor binário. Considerando apenas a intensidade luminosa que incide sobre cada célula de um único CCD, tem-se apenas pixels em tons de cinza. Entretanto, a captura de imagens coloridas tornou-se possível por meio da utilização de três sensores CCD, cada um dedicado a um canal de cor, isto é, as três cores do padrão RGB.

Os sensores CMOS seguem o mesmo princípio do CCD, transformando a luz em uma tensão elétrica, no entanto, não utilizam um conversor (A/D). Essa tarefa fica responsável pelos transistores presentes nos capacitores fotossensíveis (BARELLI, 2018).

Além desses dois sensores apresentados, existem outros que não necessitam de luz visível para formar uma imagem digital. Os sensores infravermelhos são capazes de gerar imagens através da captação do calor, e os sensores ultrassônicos detectando as vibrações ultrassônicas. Esses também possuem muitas aplicações nos estudos de visão computacional.

2.3 Visão computacional

De acordo com Bradski & Kaehler (2008), a visão computacional é o processo de modelagem e reprodução da visão humana realizada por um conjunto hardware e software. Os dados de entrada a serem considerados podem incluir informações relacionados ao contexto em que a câmera está inserida, como o lugar ou a distância que ela está posicionada na cena. Ainda nas palavras de Bradski & Kaehler (2008), a relação da visão computacional com a visão humana é apresentada da seguinte forma:

Como somos criaturas visuais, é fácil sermos enganados ao pensar que as tarefas de visão computacional são fáceis. Quão difícil pode ser encontrar, digamos, um carro quando você está olhando para ele em uma imagem? Suas intuições iniciais podem ser bastante enganosas. O cérebro humano divide o sinal da visão em muitos canais, que transmitem diferentes tipos de informação para o cérebro. Seu cérebro tem um sistema de atenção que identifica, de uma maneira dependente da tarefa, partes importantes de uma imagem para examinar enquanto suprime o exame de outras áreas. Há um *feedback* massivo no fluxo visual que ainda é pouco compreendido. Há amplas contribuições associativas dos sensores de controle muscular e de todos os outros sentidos que permitem ao cérebro recorrer a associações cruzadas feitas a partir dos anos de vivência da pessoa. Os *loops* de *feedback* no cérebro remontam a todos os estágios do processamento, incluindo os próprios sensores de hardware (os olhos), que controlam mecanicamente a iluminação através da íris e sintonizam a recepção na superfície da retina. (BRADSKI & KAEHLER, 2008)¹

A visão computacional busca simular o funcionamento do cérebro humano, utilizando ferramentas como modelos matemáticos e algoritmos para receber estímulos através da visão (imagens), processar imagens realizando associações e transmitir essas informações. Com o avanço da tecnologia, principalmente na computação, esse recente campo da ciência pode nos ajudar com inúmeras contribuições nas mais distintas áreas da medicina, biologia, física e engenharia, extraíndo informações e reconhecendo padrões que antes levariam muito tempo para serem processados nos meios convencionais.

Para facilitar a metodologia e aplicação, os sistemas de visão computacional possuem um fluxo comum a diversas aplicações, em que seguem as etapas de aquisição da imagem, pré-processamento, segmentação, extração de características, reconhecimento de padrões e resultados. Esse fluxo pode ser analisado na Figura 2.

¹ Tradução livre de Bradski & Kaehler (2008, p.2)



Figura 2. Fluxo da visão computacional. Fonte: Autor.

2.4 Linguagem de programação Python

A linguagem Python foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), como sucessor da linguagem ABC (BORGES, 2014). Atualmente é mantida pela *Python Software Foundation* (PSF) e possui uma licença de código aberto.

Python vem crescendo em várias áreas da ciência e sendo adotada por muitos pesquisadores. Essa adoção se dá, devido ao fato dela ser uma linguagem expressiva, em que a complexidade de se traduzir o raciocínio para um algoritmo é diminuída, e também por se tratar de um software livre (MENEZES, 2010).

Em pesquisas científicas, o raciocínio na maioria dos casos é complexo. Python trata alguns detalhes importantes na programação, como alocação de memória e gerenciamento de recursos de maneira automática e competente, permitindo que os pesquisadores se concentre exclusivamente com o estudo em questão, e não com detalhes técnicos da linguagem. Todo esse gerenciamento interno possibilita fluidez e agilidade nos trabalhos científicos.

Outra característica fundamental de Python é sua alta legibilidade. Com isso, permite-se uma melhor compreensão de um código escrito há algum tempo, o que faz com que bibliotecas das mais diversas aplicações já criadas na linguagem sejam implementadas constantemente em novos projetos. Assim, tem-se uma evolução de novos programas criados a partir de algoritmos anteriores.

2.5 Biblioteca OpenCV

A biblioteca OpenCV foi originalmente desenvolvida pela Intel em 2000 e idealizada para manipular e processar imagens digitais, permitindo a popularização da visão computacional a usuários, programadores e engenheiros das mais diversas áreas (BRADSKI & KAEHLER, 2008).

Essa biblioteca possui código aberto, sendo totalmente livre para o uso comercial e acadêmico, e está dividida em cinco grandes funções que são: processamento de imagens, análise estrutural, análise de movimento e rastreamento de objetos, reconhecimento de padrões e calibração de câmera, e reconstrução 3D (MARENGONI & STRINGHINI, 2009; MORDVINTSEV & RAHMAN K, c2013).

2.6 PIL/Pillow

A biblioteca PIL (*Python Imaging Library*) possui métodos e funções para manipulação de imagens e suporta alguns formatos como PNG, TIFF, BMP, EPS e GIF. Posteriormente, foi atualizada para a biblioteca Pillow com o objetivo de suportar a terceira versão da linguagem Python (LUNDH & ELLIS, 2002; LUNDH & CLARK, c2015).

2.7 NumPy

NumPy é uma biblioteca usada para realizar cálculos em arranjos/vetores multidimensionais (matrizes). Ela fornece um conjunto de ferramentas para uso em álgebra linear, transformadas de Fourier e números aleatórios (NUMPY DEVELOPERS, c2019).

Imagens são construídas de matrizes bidimensionais, sendo assim, seus recursos são fundamentais para se trabalhar no processamento de imagens, tornando as operações e manipulações envolvendo figuras mais rápidas e eficientes. Sua sintaxe se assemelha muito com o software MatLab®, porém com toda expressividade da linguagem Python.

2.8 Trefilação

Na conformação mecânica, o processo de trefilação é um dos mais utilizados pela indústria na fabricação de arames, barras cilíndricas, fios e tubos de pequeno diâmetro (BUTTON, 2001; SOARES, 2012).

Uma grande variedade de materiais metálicos pode ser conformada por trefilação, desde que apresentem a capacidade de serem deformados a frio (conformabilidade plástica) e que os parâmetros de processamento envolvidos (lubrificante, velocidade, redução por passe, número de passes, etapas de recozimento) sejam escolhidas adequadamente (BUTTON, 2001). As mais variadas classes de aços, ligas de cobre e ligas de alumínio, metais nobres como o ouro, a prata e a platina podem ser trefiladas para a obtenção de diversas geometrias de produtos (ALTAN et al., 1999; HELMAN & CETLIN; 1983).

Há uma série de normas relacionadas com as propriedades de produtos trefilados, como tolerâncias geométricas e dimensionais, rugosidade superficiais e propriedades mecânicas. A especificação das propriedades de um produto trefilado dependerá das exigências de sua utilização e dependerá da definição adequada do material de partida, acabamento superficial, diâmetro do fio, etc.

A trefilação é definida como um processo de fabricação por deformação plástica, em que a matéria-prima ou fio-máquina é tracionado ao passar por uma ferramenta cônica chamada de fieira. Esse processo resulta na redução da área da seção transversal do fio e aumento do comprimento, permanecendo assim o mesmo volume, alterando as características dimensionais da peça (SOUZA, 2011; NUNES, 2012; SOARES, 2012). A operação é comumente realizada a frio, com alteração das propriedades mecânicas do material do fio.

A redução do diâmetro da peça se dá através da deformação plástica permanente do material, causada pela reação de compressão realizada pela ferramenta no metal (fieira), enquanto o fio sofre a ação da força de tração (ALTAN et al., 1999; HELMAN & CETLIN; 1983; SOUZA, 2011; NUNES, 2012; SOARES, 2012).

Por sua vez, do outro lado da fieira, há um mecanismo responsável por aplicar a força de tração para que ocorra o trabalho da redução de diâmetro. Esse mecanismo é chamado de Cabeçote de tração, e está exemplificado na Figura 3:

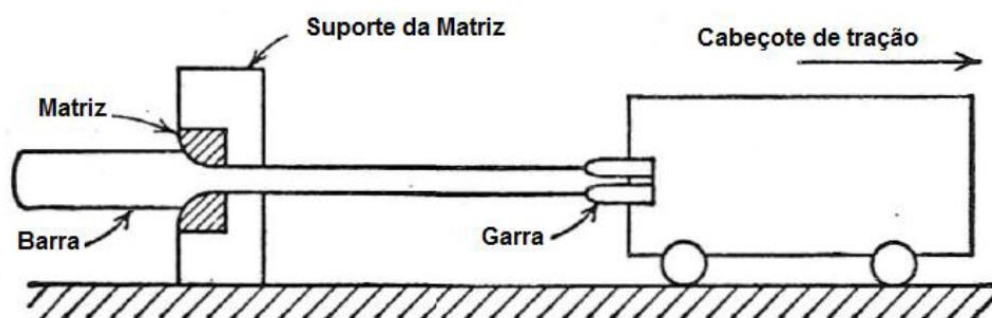


Figura 3. Ilustração esquemática do processo de trefilação. Fonte: (DIETER, 1981).

A geometria de seção transversal mais simples e mais comumente encontrada em produtos trefilados é a seção circular característica de barras, arames e fios, sendo que a diferenciação entre cada um deles se deve aos diâmetros característicos e à qualidade geral obtida na trefilação. Assim, as barras teriam diâmetros superiores a 5 mm, os fios grossos (ou arames) com diâmetros entre 2 e 5 mm, os fios médios entre 0,5 e 2 mm, os finos entre 0,15 e 0,50 mm e finalmente, os fios capilares com diâmetros menores que 0,15 mm.

Na Figura 4 pode-se observar um “zoom” esquemático do processo de trefilação. O diâmetro inicial (D_i) representa as características da matéria-prima, ou seja, antes da deformação iniciar, que afetam o comportamento do material na zona de deformação e as propriedades do produto obtido. Também são importantes a qualidade superficial e o tratamento de superfície anterior ao processo de conformação. Após a trefilação temos o diâmetro final (D_f) que representa as características do produto conformado, principalmente as propriedades mecânicas e superficiais e sua qualidade dimensional e geométrica.

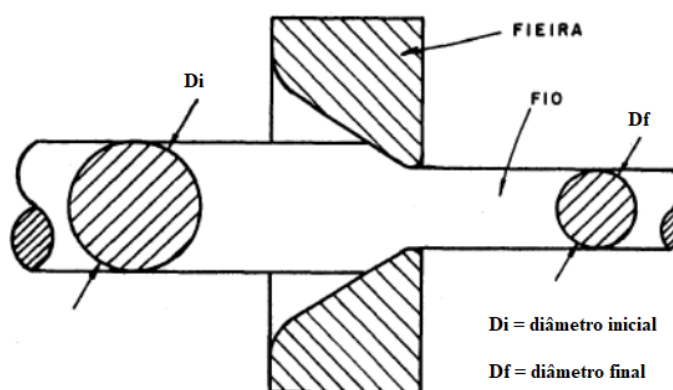


Figura 4. Esquema de fieira e fio sendo trefilado. Fonte: (BUTTON, 2001).

Para este estudo o objetivo é desenvolver um programa para medir continuamente o valor do diâmetro final (D_f), pois essa característica define o uso desse produto após a conformação. Inicialmente esse sistema de medição será feito com uma sequência de imagens obtidas por fotografias para eliminar as vibrações que ocorrem durante o processo, a iluminação insuficiente, imagem desfocadas, entre algumas perturbações que podem ocorrer no momento do processo.

3 Metodologia

3.1 Aquisição da imagem

Nesta etapa devem ser considerados dois importantes elementos para se obter uma imagem com qualidade aprovável, que facilitará nas etapas posteriores. O primeiro elemento trata-se dos equipamentos a serem utilizados no ambiente da captura, como câmeras e sistemas de iluminação. O segundo elemento é o algoritmo que faz a leitura e gerencia as próximas ações a serem realizadas (RUDEK et al., 2001).

Existem várias câmeras que podem ser utilizadas para fazer a captura. Nas câmeras digitais os sensores CCD ou CMOS são responsáveis por registrar as imagens e codificá-las em

dados digitais de bits e bytes que serão transformados em um arquivo digital (RUDEK et al., 2001; BARELLI, 2018). A partir desse arquivo gerado, é realizado então a etapa de pré-processamento.

3.2 Pré-processamento

Após a aquisição da imagem, com o arquivo digital armazenado e carregado pelo algoritmo, é desempenhado a etapa de pré-processamento para melhorar a qualidade da imagem. É realizado correção de iluminação, contraste, e ruídos com o auxílio de operações aritméticas, geométricas e aplicação de filtros (RUDEK et al., 2001; BRADSKI & KAEHLER, 2008; BARELLI, 2018).

Os ruídos gerados no processo anterior podem ser provenientes de diferentes fontes e os principais fatores são: sensores utilizados na captura da cena, iluminação do ambiente, condições climáticas no momento da aquisição da imagem e a posição relativa entre o objeto de interesse e a câmera. São considerados ruídos, não somente interferências no sinal de captura da imagem, mas também qualquer interferência que atrapalhe a interpretação ou reconhecimento dos objetos de interesse na figura (MARENGONI & STRINGHINI, 2009).

Para tratar esses ruídos, a biblioteca OpenCV possui diversos filtros que são divididos em filtros espaciais e filtros de frequência. Os filtros espaciais são matrizes que percorrem toda a imagem modificando diretamente os pixels, com a função de corrigir, realçar e suavizar regiões específicas. Essas matrizes são denominadas de máscaras ou núcleos e atuam alterando os valores da intensidade de cinza de cada pixel. O deslocamento da máscara sobre a imagem que modifica os pixels é chamado de convolução (BRADSKI & KAEHLER, 2008; MARENGONI & STRINGHINI, 2009; BARELLI, 2018).

Os filtros de frequência utilizam transformadas de Fourier para transformar a imagem do domínio espacial para o domínio da frequência. Estes também podem ser classificados quanto ao tipo de frequência em passa-baixas e passa-altas. Filtros passa-baixa bloqueiam altas frequências, suavizando a imagem atenuando regiões de bordas e contornos. Filtros passa-altas, portanto, inibem baixas frequências realçando bordas e contornos. A descrição do funcionamento e modelagem matemática dos filtros são descritas nos trabalhos de Bradski & Kaehler (2008), Marengoni & Stringhini (2009) e Barelli (2018).

Em nosso estudo, utilizou-se um filtro chamado “filtro bilateral” que possui um funcionamento semelhante a filtros passa-baixas, entretanto, modificado para preservar

detalhes de contorno. Na análise dimensional de um fio treilado, esse é o filtro mais indicado para suavizar o plano de fundo e a área do objeto de estudo preservando seus contornos.

3.3 Segmentação

A segmentação é uma das etapas mais importantes da visão computacional e consiste em separar os objetos de maior interesse de uma imagem. Os objetos a serem estudados são chamados de primeiro plano, e os pixels que não fazem parte dessa região são denominados como segundo plano (BARELLI, 2018).

Para extrair informações de um objeto é muito importante segmentá-lo. Para isso, existem alguns processos como segmentação por binarização, cor, bordas ou movimento. O processo mais utilizado e aplicado em nosso estudo é a binarização (*thresholding*) da imagem, em que um dos planos é preenchido com pixels brancos e o outro com pixels pretos. Adotando este processo, diminuimos o esforço computacional necessário para o tratamento da imagem (RUDEK et al., 2001).

Após aplicação da binarização, podem ocorrer falhas nas regiões que compõem os objetos de estudo, e para corrigir isso são aplicadas operações morfológicas. Essas operações modificam o formato ou estrutura dos objetos representados na imagem através de um elemento estruturante, exemplificado como uma imagem binária menor que a imagem original, descrita por uma matriz em que atribuímos suas dimensões (i,j) (BRADSKI & KAEHLER, 2008; BARELLI, 2018). Esse elemento estruturante percorre toda a imagem binarizada, corrigindo pixel a pixel as falhas de acordo com o tipo de operação morfológica e possui três padrões matriciais mais comuns, sendo eles o padrão retangular, elíptico e de cruz. Para escolha do padrão citado deve-se considerar a geometria do objeto de estudo (BRADSKI & KAEHLER, 2008; BARELLI, 2018).

Operações morfológicas são classificadas de acordo com suas características de funcionamento. Essas classificações podem ser divididas em dois grupos: operações de erosão e dilatação, e operações de abertura e fechamento.

Dentro do primeiro grupo, a primeira operação é caracterizada pela corrosão das arestas do objeto de estudo, que resulta no encolhimento do objeto. A segunda operação realiza o oposto, preenchendo a área de interesse do objeto aumentando suas dimensões.

Para o segundo grupo, a operação de abertura é caracterizada por aplicar as operações do primeiro grupo, uma seguida da outra, resultando em maior eficiência para tratar ruídos gerados durante a binarização. Por fim, a operação de fechamento é usada para preencher a

imagem, corrigindo falhas dentro dos objetos de interesse também causados pelo processo de binarização (BRADSKI & KAEHLER, 2008; BARELLI, 2018).

Neste estudo, utilizaram-se as operações morfológicas de abertura e fechamento com o elemento estruturante elíptico, para corrigir as falhas e ruídos gerados na binarização da imagem dos fios trefilados, que possuem uma geometria cilíndrica.

3.4 Extração de características

A extração de características é um processo essencial para categorizar objetos de interesse segmentados. Pode-se classificá-los quanto as características de aspecto, dimensionais, inerciais e topológicas.

As características de aspecto extraem informações sobre a cor e textura do objeto, e esses dados podem ser importantes para a distinção de diferentes materiais. As características dimensionais definem informações sobre as dimensões do objeto que podem ser utilizadas na obtenção da área, perímetro e diâmetro, portanto, este foi o processo utilizado no estudo. Sobre as características inerciais, as informações definem os momentos, o centro geométrico e as formas geométricas, possibilitando o reconhecimento de um objeto mesmo que ele tenha sofrido rotação, translação ou alterações de escala. A última classificação, topológica, define informações que não variam quando um objeto é rotacionado ou sofre distorções na largura e altura, exemplificado como as informações dos vértices de uma figura geométrica (BARELLI, 2018).

3.5 Reconhecimento de padrões

Ao longo do desenvolvimento da computação, desenvolveram-se ferramentas matemáticas e computacionais que permitem que objetos capturados em uma cena possam ser classificados a partir de suas características específicas. Essas técnicas buscam simular a visão humana possibilitando o reconhecimento de objetos como placas, peças, caracteres, faces humanas, células tumorais, entre outros (RUDEK et al., 2001). Segue-se então, um fluxo em que o objeto de interesse da imagem é separado, as características são extraídas e a classificação é aplicada considerando semelhanças particulares. Um bom exemplo de reconhecimento de padrões é a classificação de figuras geométricas em grupos distintos, sendo o grupo A responsável por abranger quadriláteros, o grupo B triângulos e o grupo C para classificar as demais figuras geométricas. Essa classificação pode ocorrer através da identificação e contagem de vértices da figura.

Embora a etapa de reconhecimento de padrões faça parte do processo de visão computacional, a ferramenta desenvolvida neste trabalho não faz uso de algoritmos computacionais classificadores.

4. Resultados e Discussão

Utilizando a linguagem Python 3 e as bibliotecas descritas da seção 2.5 à seção 2.7, implementou-se um sistema capaz de processar imagens e extrair informações referentes às dimensões dos objetos de estudo.

Para validar o funcionamento do sistema, utilizaram-se objetos do nosso cotidiano com dimensões conhecidas e analisou-se o resultado obtido para cada um deles. Os corpos de provas utilizados foram moedas brasileiras cujos valores de diâmetro são conhecidos e apresentados na Tabela 1.

Tabela 1. Valores e diâmetros nominais das moedas utilizadas como corpos de prova.

Valor (R\$)	D _n (mm)
0,05	22
0,10	20
0,25	25
0,50	23
1,00	27

Legenda: D_n = diâmetro nominal. Fonte: Autor.

Inicialmente, capturou-se uma imagem contendo uma unidade de cada moeda indicada na Tabela 1. Desta forma, foi possível processar e obter valores de diâmetro utilizando uma delas como parâmetro dimensional. A Figura 4 mostra a imagem original e os resultados obtidos através do processamento. As dimensões encontradas estão descritas na Tabela 2 conjuntamente com os respectivos erros percentuais. Os erros percentuais foram calculados através da seguinte fórmula: $Erro = \frac{|D_n - D_c|}{D_n} \times 100$, sendo D_n o diâmetro nominal e D_c o diâmetro de comparação, e tais erros podem estar relacionados com os ruídos descritos na seção 3.2.

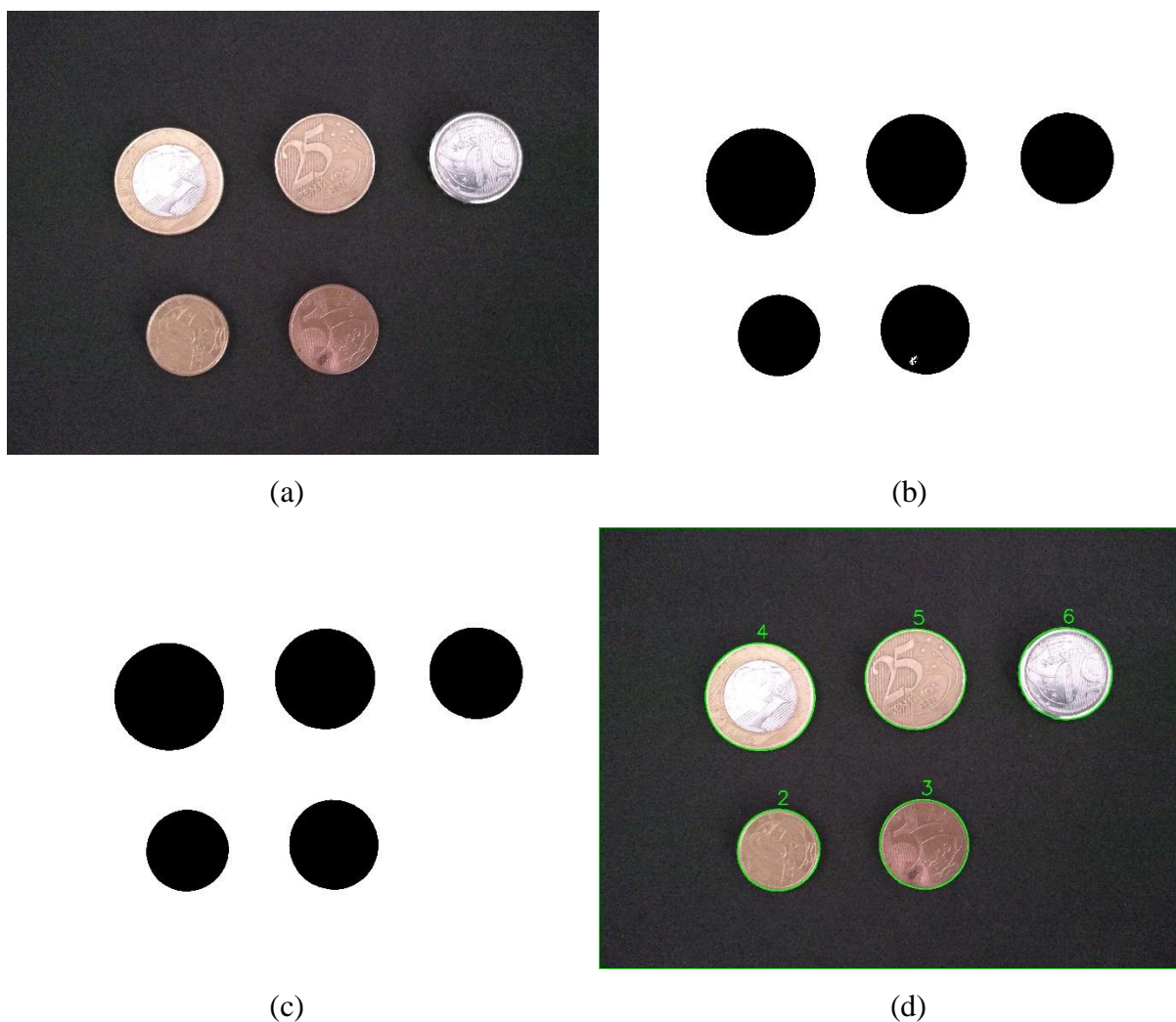


Figura 4. Imagem das moedas (corpos de prova). (a) original. (b) binarizada. (c) morfológica. (d) processada.

Fonte: Autor.

Tabela 2. Diâmetros das moedas obtidos pelo sistema de visão computacional.

Valor (R\$)	D_{vc} (mm)	Erro (%)
0,05	21,607	1,79
0,10	20,000*	-
0,25	24,296	2,82
0,50	22,596	1,76
1,00	26,553	1,66

Legenda: * Parâmetro dimensional de referência; D_{vc} = diâmetro obtido através do sistema de visão computacional. Fonte: Autor.

Pôde-se analisar na Figura 4 os resultados obtidos através do processamento de imagem das moedas (a). Nota-se na Figura 4 (b), que a binarização direta da imagem original pode resultar em pequenas regiões de falha nos objetos de estudo, assim como observado na moeda de R\$ 0,05. A Figura 4 (c) mostra que é possível corrigir essas falhas por meio de operações morfológicas e deste modo segmentar corretamente e uniformemente os objetos, assim como apresentados e numerados na Figura 4 (d).

Para obter as dimensões de um objeto de estudo é necessário, inicialmente, considerar um outro objeto com dimensões já conhecidas, pois partindo da visão computacional obtêm-se somente medidas em pixels. Desta forma, um corpo de prova é necessário para que se possa associar e relacionar dimensões em pixels com medidas reais. Para o caso de teste em que se utilizaram moedas, o corpo de prova adotado como parâmetro dimensional de referência foi a menor moeda, cujo diâmetro nominal é de 20 mm. Assim, o erro percentual apresentado na Tabela 2 referente a esta moeda não possui significado físico.

As medidas obtidas foram próximas aos valores nominais, apresentando um erro máximo de 2,82%. Deste modo, os resultados demonstraram um grau de precisão da ordem de centésimos de mm, porém tal precisão pode ser afetada por alguns fatores, tais como: dimensões do parâmetro de referência, resolução da imagem, proximidade da câmera com a relação ao objeto de estudo, iluminação e plano de fundo do ambiente, etc.

Como exemplo, podemos comparar a Figura 5, com iluminação inadequada, com a Figura 4. Nota-se claramente que alguns dos objetos de estudo (moedas de R\$ 0,10 e R\$ 0,05), receberam pouca iluminação e consequentemente foram detectadas de maneira errônea. Neste caso, mesmo com a utilização das ferramentas de suavização, binarização e morfologia, não foi possível segmentar corretamente todos os objetos, contrapondo os resultados apresentados na Figura 4.



Figura 5. Imagem das moedas com iluminação inadequada. (a) original. (b) processada. Fonte: Autor.

Após validar o funcionamento do sistema, aplicaram-se as ferramentas desenvolvidas para dimensionar fios trefilados. Para isso, utilizaram-se fios de cobre de diferentes bitolas, sendo elas 2.5, 4.0, 6.0 e 10.0 mm². A Figura 6 mostra a aplicação da visão computacional para obtenção do diâmetro do fio com bitola de 10 mm² desencapado, e a Figura 7, o resultando considerando o mesmo fio encapado.

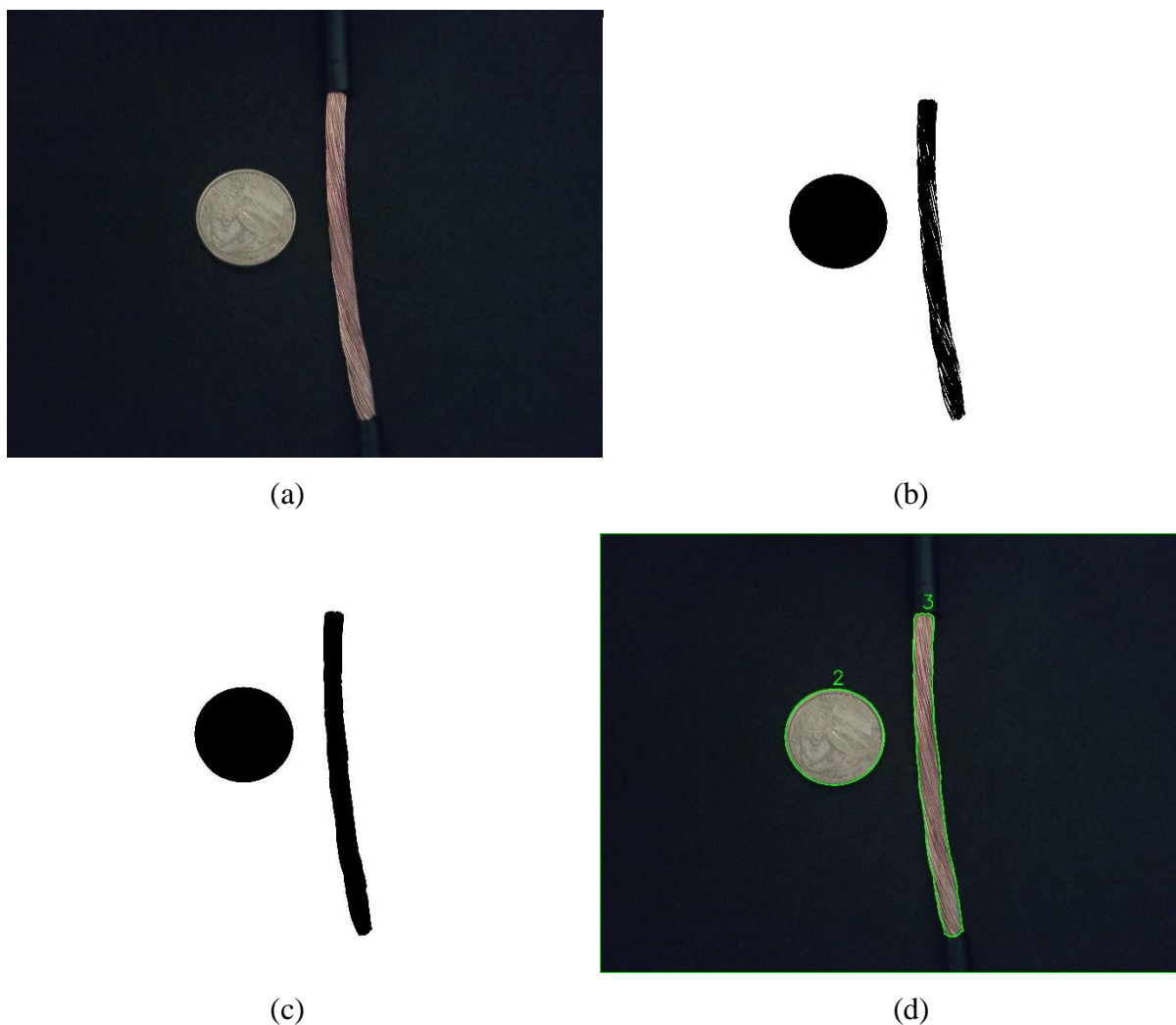


Figura 6. Imagem do fio desencapado de bitola nominal de 10 mm². (a) original. (b) binarizada. (c) morfológica. (d) processada.

Fonte: Autor.

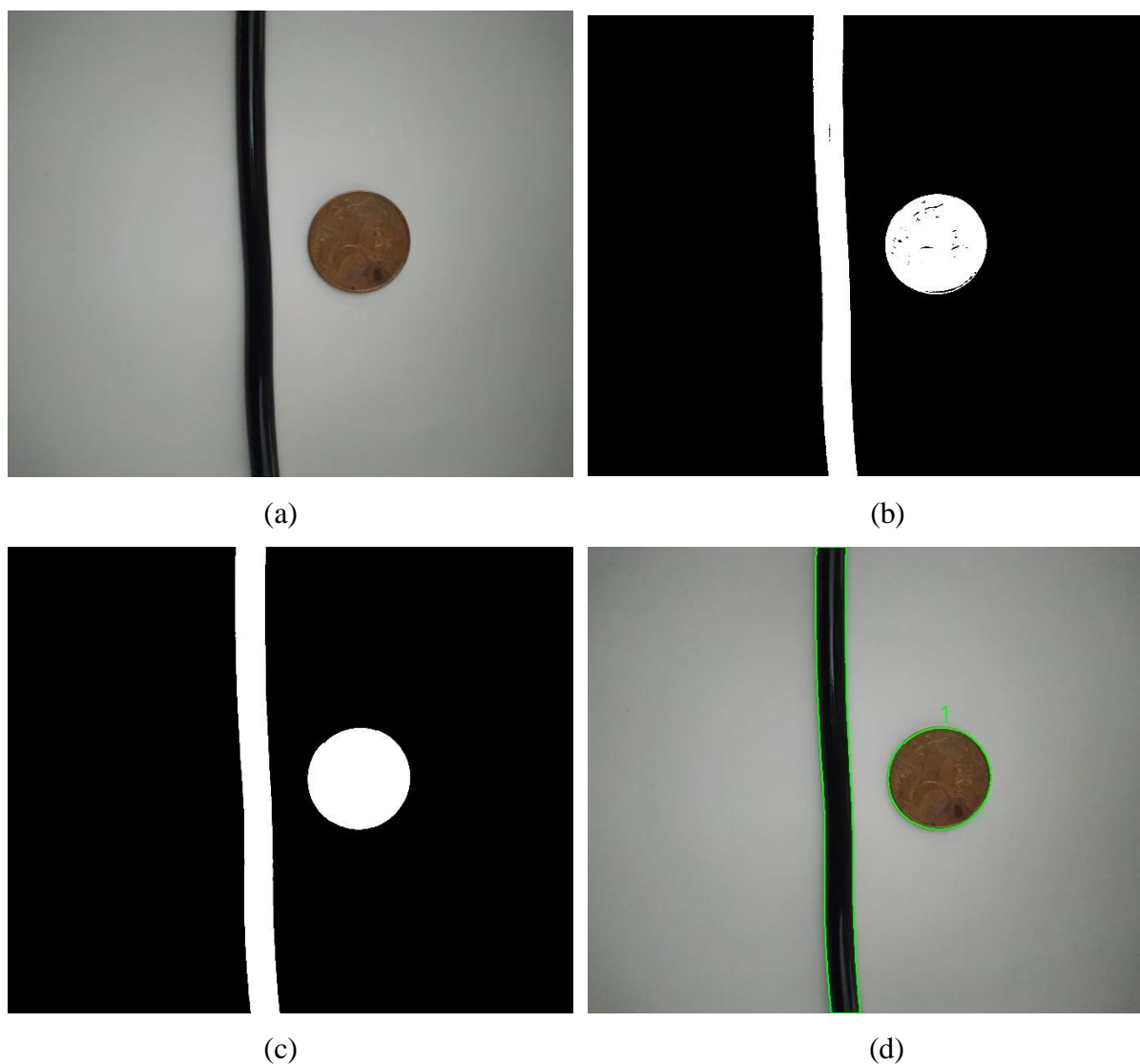


Figura 7. Imagem do fio encapado de bitola nominal de 10 mm². (a) original. (b) binarizada. (c) morfológica. (d) processada.

Fonte: Autor.

Utilizando a mesma metodologia aplicada no caso das moedas, foi possível obter o diâmetro de cada um dos fios. Para as Figuras 6 e 7, as medidas foram mensuradas utilizando como parâmetro as moedas de R\$ 0,10 e R\$ 0,05 respectivamente, e então comparadas com os valores obtidos através de um paquímetro de precisão 0,05 mm. Foram realizadas quatro medições utilizando o paquímetro para cada fio, e considerou-se a média como valor de comparação com o diâmetro nominal. A Tabela 3 mostra os resultados considerando os fios desencapados.

Tabela 3. Valores nominais e mensurados dos fios desencapados.

Bitola (mm ²)	D _n (mm)	D _p (mm)	Erro _p (%)	D _{vc} (mm)	Erro _{vc} (%)	Diferença _{pvc} (%)
2,5	1,784	1,900	6,5	2,133	19,6	12,3
4,0	2,257	2,650	17,4	2,888	27,9	9,0
6,0	2,764	3,475	25,7	3,301	19,5	5,0
10,0	3,568	3,875	8,6	4,066	13,9	4,9

Legenda: D_n = diâmetro nominal; D_p = diâmetro obtido através do paquímetro; D_{vc} = diâmetro obtido através do sistema de visão computacional; Erro_p = erro percentual do paquímetro em relação ao diâmetro nominal; Erro_{vc} = erro percentual do sistema de visão computacional em relação ao diâmetro nominal; Diferença_{pvc} = diferença entre valores medidos pelo paquímetro com o sistema de visão computacional. Fonte: Autor.

Analisando a Tabela 3, é possível notar erros percentuais relativamente altos comparados aos valores nominais, tanto para as medidas obtidas através do paquímetro quanto as medidas geradas pelo sistema de visão computacional. Essas diferenças podem ser causadas pelo fato das amostras serem formadas por uma malha de pequenos fios treilados de cobre. Desta forma, o diâmetro nominal é dado considerando o valor médio dos fios tracionados sem quaisquer deformações. Além disso, parte do erro pode ser recorrente do grau de imprecisão já observado na etapa em que se utilizaram somente moedas.

Com relação aos resultados obtidos considerando os fios encapados, pôde-se somente comparar os diâmetros gerados pelo sistema com a média dos valores obtidos com o paquímetro. Esses resultados podem ser analisados na Tabela 4.

Tabela 4. Valores nominais e mensurados dos fios encapados.

Bitola (mm ²)	D _p (mm)	D _{vc} (mm)	Erro (%)
2,5	3,4	3,613	6,3
4,0	4,0	4,427	10,7
6,0	4,5	4,937	9,7
10,0	5,9	6,029	2,18

Legenda: D_n = diâmetro nominal; D_p = diâmetro obtido através do paquímetro; D_{vc} = diâmetro obtido através do sistema de visão computacional. Fonte: Autor.

Os resultados apresentados na Tabela 4, mostram uma diferença consideravelmente menor entre os valores mensurados pelo paquímetro e o sistema. Essa proximidade entre as medidas pode estar relacionada com o fato da amostra dos fios encapados terem uma geometria mais uniforme.

5. Considerações Finais

Partindo das ferramentas computacionais apresentadas, foi possível desenvolver um sistema baseado em visão computacional capaz de obter valores de diâmetro de fios trefilados, utilizando um corpo conhecido como parâmetro dimensional. Para validar o sistema utilizou-se objetos comuns do nosso cotidiano (moedas brasileiras) e atingiu-se um resultado eficiente na análise dimensional. Após a validação, o sistema foi capaz de obter medidas relativamente próximas aos valores convencionais mensurados através de um paquímetro.

Embora tenha sido possível obter as medidas aproximadas de diâmetro dos diferentes fios, notou-se uma diferença com relação aos valores nominais para os fios desencapados. Foi perceptível que tal diferença não foi causada diretamente apenas pelo processamento realizado no sistema, pois o mesmo ocorreu com a medição convencional. Essa maior diferença pode estar relacionada com a geometria e não uniformidade do objeto de estudo.

A partir das ferramentas implementadas, observou-se a existência de adversidades que podem agir diretamente na captura e processamento de imagens, afetando os resultados finais. Assim, é possível aprimorar as ferramentas de visão computacional agindo de forma a compensar tais fatores, dentre eles a iluminação inadequada da cena em que o corpo de estudo se situa, qualidade dos sensores de captura e influência do ambiente. Como possível melhoria, também poderia ser utilizado um parâmetro dimensional de geometria uniforme e tamanho reduzido, aumentando assim a precisão das medidas que estão relacionadas com dimensões em pixels.

O estudo apresentado demonstrou que o sistema de visão computacional obteve resultados significativos em relação a medição convencional. Embora seja necessária controlar algumas variáveis responsáveis por causar interferências que atuam diretamente na qualidade do processamento de imagens, o sistema pode ser considerado uma alternativa para aplicação na indústria com o objetivo de reduzir custos de produção aumentando a rapidez no controle de qualidade de produtos.

As ferramentas de visão computacional apresentadas não se limitam apenas a análises dimensionais, e podem ser utilizadas em diversas áreas da engenharia mecânica. Assim, a

exploração de tais ferramentas pode ser de grande valia tanto para os setores industriais como em possíveis avanços em trabalhos acadêmicos que envolvam processamento de imagens e vídeos.

6. Referências

- ALTAN, T.; OH, S.; GEGEL, H. L. **Conformação de metais: fundamentos e aplicações**. São Carlos: EESC-USP, 1999.
- BALLARD, D. H.; BROWN, C. M. **Computer Vision**. Nova Jersey: Prentice Hall, 1982.
- BARELLI, F. C. **Introdução à Visão Computacional**. São Paulo: Casa do Código, 2018.
- BORGES, L. E. **PYTHON para desenvolvedores**. 1 ed. São Paulo: Novatec Editora Ltda, 2014.
- BRADSKI G.; KAEHLER A. **Learning OpenCV**. 1 ed. Sebastopol: O'Reilly Media, Inc., 2008.
- BUTTON, S. T. **Trefilação: programa de educação continuada**. São Paulo: ABM, 2001.
- DIETER, G. E. **Metalurgia Mecânica**. 2 ed. Rio de Janeiro: Guanabara Dois, 1981.
- FELICIANO, F. F.; SOUZA, I. L.; LETA, F. R. VISÃO COMPUTACIONAL APLICADA À METROLOGIA DIMENSIONAL AUTOMATIZADA: CONSIDERAÇÕES SOBRE SUA EXATIDÃO. **ENGEVISTA**, Niterói, v. 7, n. 2, p. 38-50, dez. 2005.
- HELMAN, H.; CETLIN, P. R. **Fundamentos da Conformação Mecânica dos Metais**. 1 ed. Rio de Janeiro: Guanabara Dois, 1983.
- LUNDH, F.; CLARK, A. & Contributors. **Pillow**. c2015. Disponível em: <<https://pillow.readthedocs.io/en/3.0.0/index.html>>. Acesso em: 29 ago. 2019.
- LUNDH, F.; ELLIS, M. **Python Imaging Library Overview**. PIL 1.1.3. 2002.
- MARENGONI, M.; STRINGHINI, D. **Tutorial: Introdução à Visão Computacional usando OpenCV**. Revista de Informática Teórica e Aplicada, Porto Alegre, v. 16, n. 1, 2009.
- MENEZES, N. N. C. **Introdução à Programação com Python: Algoritmos e lógica de programação para iniciantes**. 1ª edição, São Paulo: Novatec Editora Ltda, 2010.
- MORDVINTSEV, A.; RAHMAN K, A. **Welcome to OpenCV-Python Tutorials's documentation!** c2013. Disponível em: <<https://opencv-python-tutroals.readthedocs.io/en/latest/index.html>>. Acesso em: 29 ago. 2019.
- NUMPY DEVELOPERS. **NumPy Documentation**. c2019. Disponível em: <<https://numpy.org/doc/>>. Acesso em: 29 ago. 2019.

NUNES, R. M. **Estudo de distorção de barras cilíndricas de aço ABNT 1045 em uma rota de fabricação envolvendo trefilação combinada e têmpera por indução**. Tese (Doutorado em Engenharia) - Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais - PPGE3M, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. Acesso em: 30 mai. 2019.

PEIXOTO, F.M.; REBOUÇAS, E.S.; XAVIER, F.G.L. REBOUÇAS FILHO, P.P. **Revista Matéria**, v. 20, n. 1, pp. 262– 272, 2015.

RUDEK, M.; COELHO, L. S.; CANCEGLIERI JR., O. Visão Computacional Aplicada a Sistemas Produtivos: Fundamentos e Estudo de Caso. **XXI Encontro Nacional de Engenharia de Produção - 2001**, Salvador: 2001.

SOARES, C. A. T. **Análise das Tensões Residuais no Processo de Trefilação Considerando os Efeitos de Anisotropia**. 2012. Dissertação (Mestrado em Engenharia) - Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais - PPGE3M, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. Acesso em: 30 mai. 2019.

SOUZA, T. F. **Simulações Computacionais para Análise e Minimização das Tensões Residuais no Processo de Trefilação**. 2011. Dissertação (Mestrado em Engenharia) - Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais - PPGE3M, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011. Acesso em: 30 mai. 2019.