

Contagem de Fibras em Imagens de Tomografia Ultrassônica em Matriz Cimentária: uma comparação entre o Laplacian of Gaussian, Difference of Gaussian e Determinant of Hessian

Jonathan Ribeiro da Silva
Programa de Pós-graduação
em Computação Aplicada
(PPCOMP)
Instituto Federal do Espírito
Santo, Campus Serra
jota.ribeiro.silva@gmail.com

Jefferson O. Andrade
Programa de Pós-graduação
em Computação Aplicada
(PPCOMP)
Instituto Federal do Espírito
Santo, Campus Serra
jefferson.andrade@ifes.edu.br

Douglas I. R. B. Braga
Sistemas de Informação
Instituto Federal do Espírito
Santo, Campus Serra
douglasbravimbraga@hotmail.com

Rodolfo G. M. de
Andrade
Programa de Pós-Graduação
em Engenharia Metalúrgica e
de Materiais (Propemm)
Instituto Federal do Espírito
Santo, - Campus Vitória
rodolfo.andrade@ifes.edu.br

Cristina K. Dominicini
Programa de Pós-graduação
em Computação Aplicada
(PPCOMP)
Instituto Federal do Espírito
Santo, Campus Serra
cristina.dominicini@ifes.edu.br

Karin S. Komati
Programa de Pós-graduação
em Computação Aplicada
(PPCOMP)
Instituto Federal do Espírito
Santo, Campus Serra
kkomati@ifes.edu.br

RESUMO

Este trabalho propõe um sistema para a detecção automática das fibras em imagens tomográficas de uma matriz cimentícia, usando método de processamento de imagem, a detecção de *blobs* usando o LoG (Laplacian of Gaussian), DoG (Difference of Gaussian) e DoH (Determinant of Hessian). Dentre os três métodos, o melhor método foi o DoG e o pior resultado foi o DoH. Verificamos que uma vantagem do DoG é que detecta *blobs* de diferentes tamanhos, característica de seu espaço escala que avalia a imagem sob diferentes máscaras e em diferentes contrastes. Os resultados foram influenciados pelo formato dos *blobs*, os métodos não detectaram os que tinham formatos ovalados e não muito compridos.

Palavras-chave

Processamento de imagens; Detecção de *blobs*.

ABSTRACT

This study proposes a system for the automatic detection of fibers in tomographic images of a cementitious matrix, using image processing methods, specifically blob detection utilizing LoG (Laplacian of Gaussian), DoG (Difference of

Gaussian), and DoH (Determinant of Hessian). Among the three methods, the DoG method proved to be the best, while the DoH method yielded the poorest results. We observed that an advantage of the DoG method is its ability to detect blobs of varying sizes, a feature of its scale space that evaluates the image under different masks and contrasts. The outcomes were influenced by the shape of the blobs; the methods failed to detect those with oval and elongated shapes.

CCS Concepts

•Computing methodologies → Image processing; •Applied computing → *Computers in other domains*;

Keywords

Image processing; Blob detection.

1. INTRODUÇÃO

O concreto, composto basicamente de cimento, areia, brita e água, é um material usado extensivamente na construção civil. Tem-se na resistência à compressão sua principal propriedade mecânica, enquanto a resistência à tração possui valor reduzido o suficiente para ser negligenciado no dimensionamento estrutural [17]. Este material é o segundo mais consumido no mundo, inferior apenas à água [19]. E, considerando que seus elementos constituintes são provenientes de matérias-primas não renováveis, a manutenção da vida útil de elementos de concreto é um fator determinante para um desenvolvimento sustentável da construção civil.

Os elementos estruturais de concreto como pontes, passarelas, edificações residenciais ou comerciais, quando em serviço (submetidas a um fluxo de veículos, pessoas, e outros carregamentos), apresentam formação de fissuras, um

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

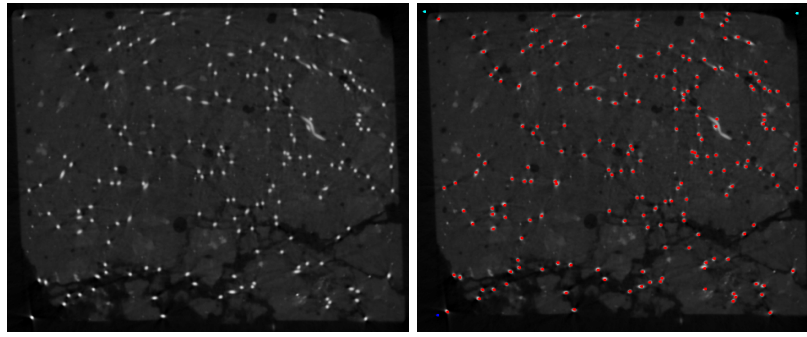


Figure 1: Imagem tomográfica obtida após o processo de tomografia computadorizada no concreto e sua marcação manual dos pontos feitos manualmente [5].

processo inerente ao material. Desse modo, deve-se dimensionar e limitar corretamente a abertura de tais fissuras [2], a fim de que permaneçam dentro de limites estipulados por normas técnicas, de forma a evitar a redução de sua vida útil [8].

Além das tradicionais barras de aço inseridas dentro do concreto, uma solução complementar para controlar e diminuir a formação de fissuras é o reforço com fibras [6]. Embora haja diversos tipos de fibras adotadas na construção civil, como polipropileno, polímeros e as de aço, a última é a mais utilizada, pois são as que proporcionam maior resistência à tração ao concreto, além de uma melhor controle da fissuração [4]. Para melhor aproveitar seu uso, as fibras de aço, que possuem corpo predominantemente reto e cilíndrico, são distribuídas e dispostas de forma aleatória quando misturadas ao concreto. A contribuição mecânica das fibras é quantificada pela quantidade, distribuição e orientação espacial, além de seu alinhamento aos planos de tensão.

Dado que a contribuição mecânica máxima ao concreto está relacionada à quantidade de fibras de aço distribuídas no concreto, é de extrema importância conhecer suas posições no espaço ao longo do seguimento da estrutura que está sendo analisada. Para isso, destaca-se a ferramenta de tomografia computadorizada 3D, que tem sido usada para detecção e consequente determinação da orientação das fibras de aço dentro do concreto endurecido [5]. A partir de cada corte tomográfico, uma imagem é formada, e então convertido em um mapa capaz de fornecer localização e orientação de cada fibra individual dentro do concreto.

Na Figura 1 à esquerda, tem-se diversos pontos cinzas, que representam as fibras de aço inseridas no concreto; seus formatos variam entre circular e elipsoidal, dada a sua distribuição espacial. Já na Figura 1 à direita, observa-se a mesma imagem após passar pelo processo manual utilizando o plugin da ImageJ¹ chamado Cell Counter [7]. Deve-se destacar que o processo de identificação e contagem manual das fibras levam em consideração a possibilidade de outros materiais, como grãos de areia, serem confundidos com as fibras de aço durante o levantamento.

Neste trabalho, o foco é a localização e a contagem das fibras, mas não a questão da análise da distribuição e orientação. Este trabalho propõe um sistema para a detecção automática das fibras em imagens tomográficas de uma matriz cimentícia, usando método de processamento de imagem, a detecção de *blobs*. Dentre as várias técnicas de detecção

de *blobs*, decidiu-se fazer a comparação entre os operadores LoG (Laplacian of Gaussian), DoG (Difference of Gaussian) e DoH (Determinant of Hessian). Antes da aplicação dos operadores, é realizado um estudo do valor do limiar na binarização da imagem, que é um pré-processamento aplicada à imagem de entrada. A base de dados de imagens foi fornecida pelo primeiro autor do artigo [5]. A análise é feita de duas formas, quantitativamente e qualitativamente.

Este texto segue com a Seção 2, que descreve trabalhos correlatos, a Seção 3 detalha a base de dados, a seleção de características e os métodos usados, a Seção 4 apresenta os resultados e a discussão dos mesmos e a Seção 5 traz a conclusão do trabalho.

2. REFERENCIAL TEÓRICO E TRABALHOS CORRELATOS

Um *blob* é uma região em uma imagem digital que difere em propriedades, como brilho ou cor, em comparação com as regiões vizinhas. Informalmente, um *blob* é uma região de uma imagem na qual algumas propriedades são constantes ou aproximadamente constantes, todos os pontos em um *blob* podem ser considerados, em algum sentido, semelhantes entre si. *Blobs* aparecem de maneiras diferentes dependendo de sua escala e ocorrem em diferentes formas.

Há uma variedade de definições de *blobs*, o trabalho de [11] apresenta cinco versões diferentes. Neste trabalho, seguimos a definição de Lindeberg [14], que define um *blob* como sendo uma região associada a pelo menos um extremo local, seja um máximo ou um mínimo, tendo uma aparência de uma bolha clara ou escura, respectivamente.

Existem duas classes principais de detectores de *blob*, segundo [21]: (i) métodos diferenciais, que são baseados em derivadas da função em relação à posição, e (ii) métodos baseados em extremos locais, que se baseiam em encontrar os máximos e mínimos locais da função. Os métodos diferenciais, em geral, são usados para detecção de bordas e cantos enquanto os métodos baseados em extremos locais para as regiões de *blobs*.

Há diferentes abordagens dos métodos baseados em extremos locais, Grauman [10] divide em três tipos: por localização do ponto-chave (Keypoint Localization), por detecção de região invariável de escala (Scale Invariant Region Detection) e por detecção de região covariante afim (Affine Covariant Region Detection). Neste trabalho, decidimos pela abordagem por detecção de região invariável de escala, cujos métodos mais usados são: LoG (Laplacian of Gaussian),

¹<https://imagej.net/ij/index.html>

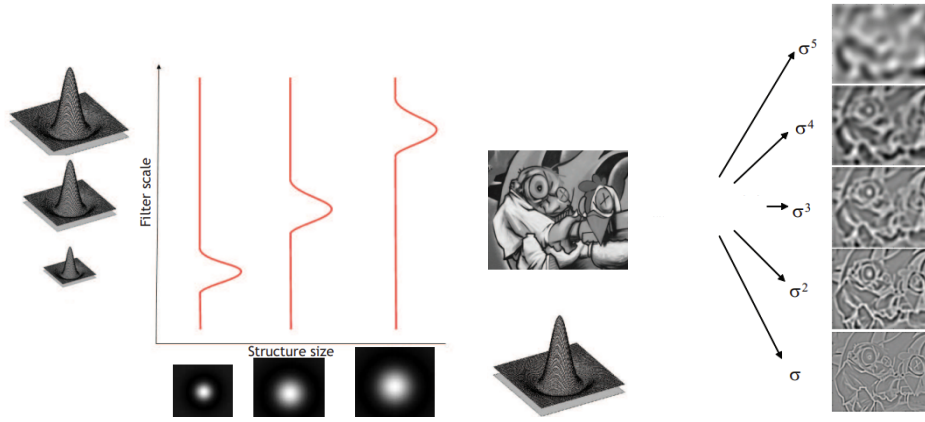


Figure 2: À esquerda: gráfico do filtro chapéu mexicano em 3 escalas, e sua imagem de *kernel* associado. À direita: aplicação de LoG com cinco diferentes valores de escala em uma imagem.

DoG (Difference of Gaussian) e DoH (Determinant of Hessian). A vantagem da abordagem por detecção de região invariável de escala é que consegue detectar estruturas, de forma confiável, mesmo sob mudanças de escala.

2.1 DoG

DoG é uma técnica que envolve a subtração de uma imagem suavizada (filtrada por uma Gaussiana) de outra imagem suavizada (filtrada por uma Gaussiana com um σ diferente), a fim de obter regiões de destaque ou bordas na imagem original.

O filtro gaussiano é muito utilizado no processamento de imagens e sinais para reduzir ruídos. Esse filtro é baseado na distribuição gaussiana, também conhecida como distribuição normal, e é amplamente utilizado para atenuar as frequências altas em um sinal ou imagem, preservando as frequências baixas [9]. Basicamente é aplicada uma média ponderada dos valores de pixel em uma vizinhança em torno de cada pixel na imagem, isto é o kernel. Os pixels mais próximos do centro do kernel têm um peso maior na média, seguindo a forma da curva gaussiana. Através disso a imagem é suavizada, reduzindo os ruídos e as diferenças bruscas entre os pixels mais próximos. Um filtro gaussiano é definido por [9]:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

onde $G(x, y, \sigma)$ representa a função gaussiana bidimensional para as coordenadas (x, y) com desvio padrão σ . É o desvio padrão, que determina o quão amplo é a suavização ou o “borramento”.

Dada uma imagem $I(x, y)$, aplicar a função gaussiana é equivalente a convolucionar essa imagem com o *kernel* gaussiano:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2)$$

onde $*$ denota a operação de convolução e $L(x, y, \sigma)$ é a imagem suavizada.

Agora, para obter o DoG, subtraímos duas imagens suavizadas (uma com desvio padrão σ e outra com desvio padrão $k\sigma$), sendo k um fator de escala maior que 1:

$$\text{DoG}(x, y, \sigma, k) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3)$$

Logo, o DoG é obtido subtraindo-se duas imagens borradas, uma com um desvio padrão maior e outra com um desvio padrão menor de uma distribuição Gaussiana. O resultado é uma imagem em que as características de escala fina são suprimidas, enquanto as características de escala mais grossa são realçadas.

2.2 LoG

Lindeberg [15] propôs o uso do operador de Laplace ou operador Laplaciano sobre uma gaussiana, que resulta no LoG (comumente chamado de chapéu mexicano). Para se criar um espaço de escala do LoG deve-se aplicar o operador Laplaciano em gaussianas de diferentes desvio-padrão na imagem de entrada (Figura 2). O Laplaciano de uma imagem I é a soma das segundas derivadas espaciais. Para uma imagem 2D:

$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (4)$$

Primeiro, suavizamos a imagem usando um filtro gaussiano e, em seguida, aplicamos o Laplaciano. A operação combinada pode ser representada como:

$$\text{LoG}(x, y, \sigma) = (\nabla^2(G(x, y, \sigma)) * I(x, y)) \quad (5)$$

O LoG age como um detector de bordas, já que a borda ou transição na imagem resultará em valores elevados após a operação LoG. Este método é considerado melhor do que o DoG em termos de localização precisa das bordas, mas é computacionalmente mais caro, pois requer cálculo de derivadas de segunda ordem.

2.3 DoH

De acordo com Perdoch et al. [20], o Detector of Hessian (DoH) é usado para uma representação eficiente da geometria local em aplicações de recuperação de objetos em grande escala. Trata-se de um algoritmo utilizado para detecção de bordas e se baseia nas propriedades da matriz Hessian, que descreve as variações da segunda ordem nas intensidades de pixel da imagem. O DoH procura pontos onde as variações

são máximas em todas as direções e onde as intensidades aumentam ou diminuem.

O Hessian é uma matriz que contém as segundas derivadas da imagem. Para uma imagem 2D, a matriz Hessian H em um ponto (x, y) é dada por:

$$H = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (6)$$

Aqui, I é a intensidade da imagem no ponto (x, y) . O determinante do Hessian é:

$$\det(H) = \left(\frac{\partial^2 I}{\partial x^2}\right)\left(\frac{\partial^2 I}{\partial y^2}\right) - \left(\frac{\partial^2 I}{\partial x \partial y}\right)^2 \quad (7)$$

O determinante do Hessian reflete a curvatura da imagem em diferentes direções. Valores altos (tanto positivos quanto negativos) do determinante indicam regiões de grande variação ou pontos-chave na imagem. Para o DoH, assim como o LoG, aplicamos primeiro um filtro gaussiano na imagem para suavização e depois calculamos o determinante do Hessian da imagem suavizada:

$$\text{DoH}(x, y, \sigma) = \det(H(G(x, y, \sigma)) * I(x, y)) \quad (8)$$

Os pontos máximos e mínimos locais do DoH são considerados pontos-chave ou características da imagem. Em comparação com o LoG, o DoH é computacionalmente mais rápido, mas pode ser menos preciso em algumas situações.

2.4 Trabalhos Correlatos

Kong et al em seu artigo [12] propõe um filtro generalizado Laplaciano de Gaussiano (gLoG) para detectar estruturas de *blob* elípticas gerais em imagens. O filtro gLoG pode não apenas localizar com precisão os centros de *blob*, mas também estimar as escalas, formas e orientações dos *blobs* detectados. O detector gLoG proposto é aplicado a imagens biomédicas e naturais, como imagens gerais de cenas de estradas. Para as aplicações biomédicas em imagens microscópicas patológicas e fluorescentes, o detector gLoG detectou com precisão os centros e estimar os tamanhos e orientações dos núcleos celulares. Para a aplicação em imagens de estradas, o detector proposto produziu estimativas promissoras de orientações de textura.

No trabalho de Xu et al [23], apresenta-se uma técnica automática baseada no filtro gLoG para detecção de núcleos em imagens histológicas digitalizadas. A técnica proposta primeiro gera um banco de *kernels* gLoG com diferentes escalas e orientações e, em seguida, realiza a convolução entre *kernels* gLoG direcionais e a imagem candidata para obter um conjunto de mapas de resposta. Os máximos locais dos mapas de resposta são detectados e agrupados em diferentes grupos pelo algoritmo *mean-shift* com base em sua proximidade geométrica. O ponto que tem a resposta máxima em cada grupo é selecionado como a semente do núcleo. Resultados experimentais em dois conjuntos de dados mostram que a técnica proposta fornece um desempenho superior na detecção de núcleos em comparação com as técnicas existentes.

No trabalho de Maroni [16] é proposto um protótipo de aplicativo para detecção automática de acne, por meio do processamento de fotos à distância tiradas por dispositivos móveis. O *pipeline* do aplicativo é composto por detecção

de partes do corpo, segmentação de pele, mapeamento de calor, extração de acne e detecção de *blobs*. Esta última tarefa foi feita pelo operador LoG. Os resultados contém a quantidade, a posição e o raio das acnes detectadas.

Assirati [1] apresentou uma proposta de compor o filtro DoG utilizando os kernels q-Gaussianos, mostrando que essa composição é uma excelente alternativa ao LoG e DoG com kernels Gaussianos clássicos, pois comparado ao filtro LoG, o método proposto possui menor custo computacional e comparado ao filtro DoG com kernels usando a distribuição normal de probabilidades, o ganho é em detalhes de detecção de bordas. Os kernels q-Gaussianos foram usados por Soares e Murta [22] para redução de ruído em imagens de ressonância magnética, obtiveram resultados mostrando que os filtros Q-Gaussianos melhoram o ganho efetivo de filtragem, quando comparados à filtragem Gaussiana clássica.

Mikolajczyk et al [18] apresenta uma análise abrangente do desempenho de vários detectores de regiões afins. Esses detectores incluem o Detector de Hessian (DoH). O DoH é avaliado em relação a outros métodos para determinar sua eficácia na detecção de pontos de interesse em imagens com variações de escala, rotação e iluminação.

O artigo de Lakemond, Fookes e Sridharan [13] apresenta uma abordagem para a adaptação afim de características de imagens locais usando a matriz Hessiana, incluindo o Detector de Hessian (DoH), para identificar e adaptar pontos de interesse afins em imagens. O trabalho demonstra a eficácia da técnica em aplicações de vigilância baseadas em vídeo e sinal, fornecendo uma contribuição valiosa para a área de processamento de imagens e visão computacional.

3. MATERIAIS E MÉTODOS

3.1 Base de dados

Para os experimentos, foram analisadas imagens tomográficas disponibilizadas por um dos autores do artigo [5], geradas a partir de elementos de concreto confeccionados em laboratório [4]. As imagens são de tamanhos diferentes, com variação de intensidade, de iluminação, e há diferença de contraste entre o fundo e os pontos que representam as fibras em algumas imagens.

Cada uma das amostras fornecidas contam com uma marcação manual feita pelo autor, com um arquivo associado de extensão .csv informando a coordenada de cada um desses pontos. Os 4 cantos da imagem são marcados manualmente, mas não são fibras. A partir daqui, as fibras ou regiões circulares em um cinza mais claro serão denominadas como *pontos de fibras*.

3.2 Método de Detecção e Contagem de fibras

3.2.1 Métodos LoG, DoG e DoH

Para realizar a detecção e contagem de fibras em uma imagem tomográfica da matriz cimentícia foram utilizados os métodos LoG, DoG e DoH, utilizando as funções **blob_log**, **blob_dog** e **blob_doh** da biblioteca scikit-image. Um *blob* é um objeto de interesse dentro de uma imagem. As funções tem os seguintes parâmetros em comum:

- **imagem**: a imagem de entrada em que se quer detectar os *blobs*. A imagem deve estar em tons de cinza de entrada, a função espera que os *blobs* sejam claros no fundo escuro.

- **overlap**: do tipo `float`, é opcional e o valor padrão é 0,5. Deve ser um valor entre 0 e 1. Se a área de dois *blobs* se sobrepuser por uma fração maior que o limite, o *blob* menor será eliminado.
- **min_sigma**: tipo escalar ou sequência de escalares, é opcional e o valor padrão é 1. É o desvio padrão mínimo para o *kernel* gaussiano. Valores baixos detectam *blobs* menores.
- **max_sigma**: tipo scalar ou sequência de escalares, é opcional e o valor padrão é 50. O desvio padrão máximo para o *kernel* gaussiano. Valores alto detectam *blobs* maiores.
- **threshold**: do tipo `float`, é opcional e o valor padrão é 0,2. É o limite inferior absoluto para máximos do espaço de escala. Máximos locais menores que o limite são ignorados. A redução deste valor faz com que a função detecte *blobs* com intensidades mais baixas. Se **threshold_rel** também for especificado, qualquer limite maior será usado. Se for **None**, **threshold_rel** será usado.
- **threshold_rel**: do tipo `float` ou **None**, é opcional. Intensidade mínima de picos, calculada como $\max(og_space) * threshold_rel$, onde **og_space** se refere à pilha de imagens LoG, DoG ou DoH computadas internamente. Deve ter um valor entre 0 e 1. Se não for informado, o valor de **threshold** será usado.

A função **blob_log**² tem os seguintes parâmetros específicos:

- **num_sigma**: do tipo inteiro, é opcional e o valor padrão é 10. É o número de valores intermediários de desvios padrão a serem considerados entre **min_sigma** e **max_sigma**.
- **log_scale**: do tipo `bool`, é opcional. Se definido, os valores intermediários dos desvios padrão são interpolados usando uma escala logarítmica na base 10. Caso contrário, é usada a interpolação linear.
- **exclude_border**: pode ser uma **tupla de int**, **int** ou **False** e é opcional. Se for uma tupla de inteiros, o comprimento da tupla deve corresponder à dimensionalidade da matriz de entrada. Cada elemento da tupla excluirá picos de **exclude_border-pixels** da borda da imagem ao longo dessa dimensão. Se for diferente de zero, **exclude_border** exclui picos de **exclude_border-pixels** da borda da imagem. Se for zero ou falso, os picos serão identificados independentemente da distância da borda.

A função **blob_dog**³ tem os seguintes parâmetros:

- **sigma_ratio**: do tipo `float`, é opcional. É a razão entre o desvio padrão dos *kernels* Gaussianos usados para calcular a Diferença de Gaussianas.
- **exclude_border**: tal qual descrição do parâmetro de mesmo nome do LoG.

² scikit-image.org/docs/dev/api/skimimage.feature.html#skimage.feature.blob_log

³ scikit-image.org/docs/dev/api/skimimage.feature.html#skimage.feature.blob_dog

A função **blob_doh**⁴ tem os seguintes parâmetros:

- **num_ratio**: do tipo `float`, é opcional e o valor padrão é 10. É o número de valores intermediários de desvios padrão a serem considerados entre **min_sigma** e **max_sigma**.
- **log_scale**: do tipo `bool`, é opcional. Se definido, os valores intermediários dos desvios padrão são interpolados usando uma escala logarítmica na base 10. Caso contrário, é usada a interpolação linear.

O retorno dessas funções é semelhante, sendo um valor **n** com a quantidade de *blobs* encontrados; uma matriz 2D, em que cada linha contém uma tupla (r, c, sigma), em que (r, c) representam as coordenadas do *blob* detectado e sigma é o desvio padrão do *kernel* gaussiano que detectou o *blob*, no caso de DoH é o desvio padrão do *kernel* gaussiano da Matriz Hessiana.

3.2.2 O fluxograma do sistema

O fluxograma presente na Figura 3 demonstra a sequência de passos do algoritmo desenvolvido. Logo ao iniciar a execução existe um *loop* para verificar se o valor de *bin_coef* é menor que o valor máximo, 0,9, caso positivo é iniciado o processo. A leitura das amostras é o processo em que o nome de cada amostra é recuperado juntamente com as marcações manuais.

Em seguida para cada amostra é feita a conversão da imagem de RGB para escala de cinza, a binarização utilizando o *bin_coef*, a operação de fechamento morfológico. O fechamento tende a suavizar áreas de contorno, mas também funde pequenas quebras e abismos longos e finos, remove pequenos buracos e preenche lacunas [3]. Com este pré-processamento é possível passar uma imagem com os objetos conectados sendo diferenciados pela escala de cinza. Identificar objetos conectados permite isolar as fibras na imagem, aumentando a precisão do processo de detecção.

Segue-se com a identificação dos *blobs* utilizando as funções **blobs_log**, **blobs_dog** e **blobs_doh**. Para cada operador as coordenadas dos *blobs* são armazenadas em um dicionário, para que depois possa ser fácil o acesso apenas passando a chave do operador ('LoG', 'DoG' ou 'DoH'). O valor de *max_sigma* é fixado em 10 para ser usado nas três funções (*blob_log*, *blob_dog*, *blob_doh*), apenas a função *blob_log* utiliza o parâmetro *num_sigma* que também foi fixado em 10.

Após a identificação dos *blobs*, são feitas as comparações com as marcações manuais, isso para cada algoritmo, calculando e identificando a posição de cada fibra que foi identificada corretamente, a quantidade de pontos de fibras que não foram identificadas e a quantidade de pontos de fibras que foram identificadas mas não estão presentes na marcação manual.

O próximo passo é a geração das imagens de resultado utilizando as coordenadas identificadas e fazendo a identificação dos pontos de fibras que foram identificados manualmente, os que foram identificados pelo algoritmo e os que deixaram de ser identificados. Essas imagens são armazenadas assim como os resultados numéricos para posterior avaliação.

O passo seguinte é adicionar ao valor de *bin_coef* 0,1 como o próximo valor a ser testado, então o ciclo se reinicia, caso tenha atingido o valor máximo, é encerrada a execução.

⁴ scikit-image.org/docs/dev/api/skimimage.feature.html#skimage.feature.blob_doh

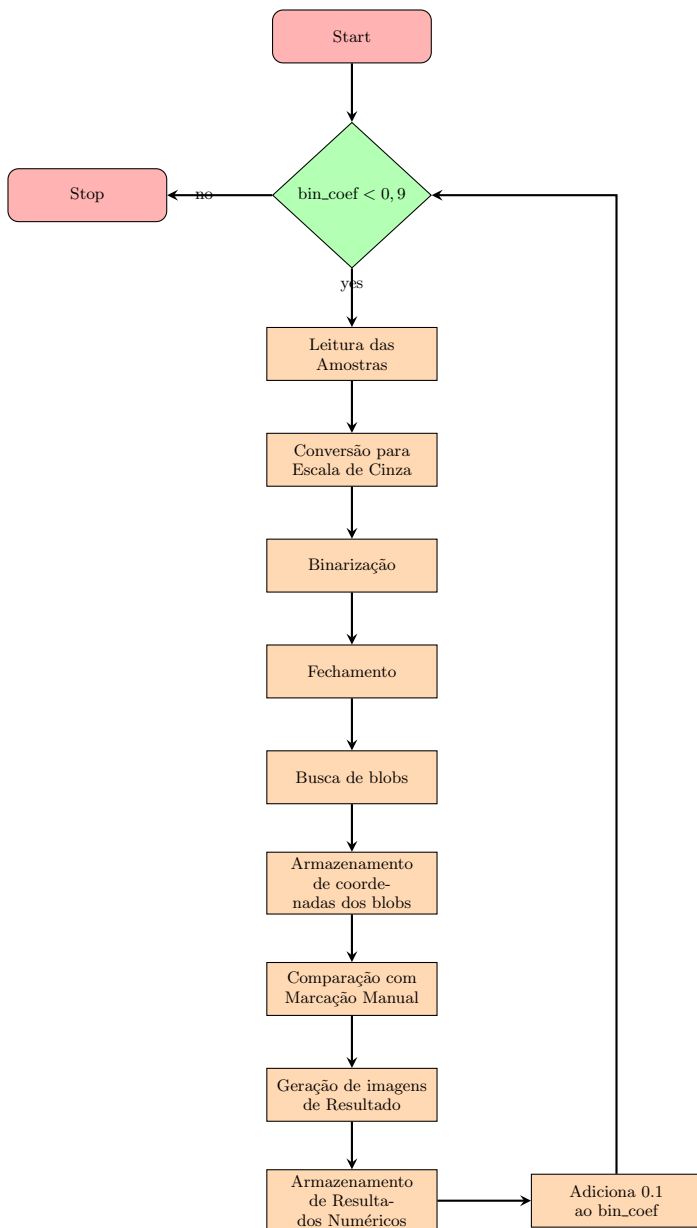


Figure 3: Fluxograma do sistema

3.3 Métricas de Avaliação

Na classificação binária, precisão é a fração de instâncias recuperadas que são relevantes, enquanto revocação é a fração de instâncias relevantes que são recuperadas. Suponha que um motor de pesquisa retorna 30 páginas, mas dessas apenas 20 são relevantes enquanto deixa de retornar 40 outras páginas relevantes, a precisão é de 20/30, enquanto a revocabilidade é 20/60.

Precisão, neste caso, é “o quanto os resultados da pesquisa são úteis”, enquanto revocabilidade é “o quão completos os resultados estão”. Em termos simples, alta precisão significa que o algoritmo retornou substancialmente mais resultados relevantes que irrelevantes, enquanto alta revocabilidade significa que o algoritmo retornou a maioria dos resultados relevantes. Neste trabalho, o resultado do sistema são os ele-

mentos que estão no círculo e os elementos relevantes são os pontos marcados manualmente. Assim:

- Os verdadeiros positivos (VP) são os *blobs* detectados corretamente pelo sistema. Entende-se que os *blobs* detectados corretamente são os que possuem associação com um ponto detectado manualmente, com uma distância de até 5 pixels.
- Os *blobs* detectados pelo sistema, mas que não eram pontos de fibra conforme a marcação manual do especialista são os falsos positivos (FP).
- Os *blobs* que o sistema não detectou, mas que eram para ter sido detectados, pois foram marcados pelo especialista, são os falsos negativos (FN).
- A revocação é a taxa de VP pela quantidade de elementos marcados pelo especialista.
- A precisão é a taxa de VP pela quantidade de elementos encontrados pelo sistema.

4. EXPERIMENTOS, RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados os resultados do processamento das oito amostras utilizadas da base de dados (CP06, CP07, CP09, CP11, CP22, CP23, CP28 e CP29), que representam imagens tomográficas da seção transversal de corpos de prova (CPs) prismáticos de tamanho variável.

Para fazer os testes comparativos entre os métodos LoG, DoG e DoH, foram feitas diversas interações no *loop* explicado no fluxograma da Figura 3. Considerou-se o valor inicial de *bin_coef* como 0,2 e a cada iteração este valor foi acrescido de 0,1 até o limite de 0,9. Desta forma, o parâmetro *bin_coef* variou de 0,2 até 0,8, enquanto o valor de *threshold* ficou fixo em 0,01.

A Tabela 1 apresenta os resultados em relação a quantidade de pontos de fibras encontrados corretamente, denotado como VP, e os parâmetros que foram usados para cada imagem, destacando em negrito os melhores resultados. É possível verificar que os melhores resultados para a taxa de VP foram obtidos utilizando o método DoG, exceto pelas amostras CP09 e CP22 que tiveram resultados de VP um pouco maiores para o método LoG, mas que podem ser consideradas como um empate. Para as amostras CP07 e CP11, houve empate entre LoG e DoG para o mesmo parâmetro de *bin_coef*. Os melhores resultados de DoH ficaram acima de 80%, embora não tenha sido o suficiente para se destacar em relação aos resultados dos outros dois operadores.

A Tabela 2 apresenta as métricas dos melhores resultados encontrados para cada amostra. Cada coluna tem a identificação da amostra e nas linhas são apresentadas: a quantidade de pontos marcada pelo especialista (Total Manual), a quantidade de pontos de fibra detectadas pelo sistema (Total), a quantidade de verdadeiros positivos (Corretos), a quantidade de falsos negativos (Não identificados pelo sistema), a quantidade de falsos positivos (Identificados, mas não deveriam), a precisão e a revocação.

Destacam-se em negrito os resultados de precisão da amostra CP22, observando que dentre as amostras avaliadas foi a que teve o menor valor nesta métrica, por conta da quantidade de falsos positivos apresentada nos três operadores. Apesar disso, o resultado de revocação demonstra que, mesmo os algoritmos não sendo precisos para essa amostra, a

Table 1: Melhores Parâmetros por Imagem e Resultado dos Métodos para a Métrica VP

Amostra	LoG		DoG		DoH	
	bin_coef	VP	bin_coef	VP	bin_coef	VP
CP06	0,4	93,56	0,4	97,03	0,4	81,20
CP07	0,8	98,02	0,8	98,02	0,8	95,05
CP09	0,6	97,97	0,5	97,46	0,6	93,90
CP11	0,3	97,92	0,3	97,92	0,3	91,66
CP22	0,2	93,36	0,2	92,89	0,2	82,46
CP23	0,7	95,24	0,7	96,43	0,7	83,63
CP28	0,3	94,83	0,3	95,26	0,3	89,22
CP29	0,2	95,24	0,2	96,83	0,2	88,09

Table 2: Métricas dos resultados da detecção de pontos de fibra

	CP06	CP07	CP09	CP11	CP22	CP23	CP28	CP29
Total manual (relevantes)	202	101	197	96	211	336	232	252
LoG								
Total	234	121	234	116	330	383	288	303
Corretos (VP)	189	99	193	94	197	320	220	240
Não identificados pelo sistema (FN)	13	2	4	2	14	16	12	12
Identificados, mas não deveriam (FP)	45	22	41	22	133	63	68	63
Precisão	80,77%	81,82%	82,48%	81,03%	59,70%	83,55%	76,39%	79,21%
Revocação	93,56%	98,02%	97,97%	97,92%	93,36%	95,24%	94,83%	95,24%
DoG								
Total	251	116	214	130	365	407	298	309
Corretos (VP)	196	99	192	94	196	324	221	244
Não identificados pelo sistema (FN)	6	2	5	2	15	12	11	8
Identificados, mas não deveriam (FP)	55	17	22	36	169	83	77	65
Precisão	78,09%	85,34%	89,72%	72,31%	53,70%	79,61%	74,16%	78,96%
Revocação	97,03%	98,02%	97,46%	97,92%	92,89%	96,43%	95,26%	96,83%
DoH								
Total	226	110	197	113	290	333	272	277
Corretos (VP)	164	96	185	88	174	281	207	222
Não identificados pelo sistema (FN)	38	5	12	8	37	55	25	30
Identificados, mas não deveriam (FP)	62	14	12	25	116	52	65	55
Precisão	72,57%	87,27%	93,91%	77,88%	60,00%	84,38%	76,10%	80,14%
Revocação	81,19%	95,05%	93,91%	91,67%	82,46%	83,63%	89,22%	88,10%

maioria dos pontos de fibras marcados manualmente foi corretamente detectada.

Além da análise das métricas, são evidenciadas as imagens geradas que apresentaram os melhores resultados para cada operador e cada amostra. Na Figura 4, são apresentados os resultados qualitativos para o operador LoG. Como comentado, a quantidade de falsos positivos na amostra CP22 é grande e pode ser visualizada pela presença alta de pontos azuis na Figura 4 e).

Na Figura 5, conseguimos constatar a alta revocabilidade do operador DoG por conta dos pontos amarelos destacados, indicando que houveram vários acertos de pontos de fibras relevantes.

Enquanto na Figura 6, podemos destacar a amostra CP09 presente no item c), que teve sua precisão e revocação com o mesmo percentual, evidenciando os acertos em amarelo e poucos pontos de falso positivo (azul) e falso negativo (vermelho).

A Figura 7 apresenta um exemplo para a amostra CP23 que ajuda a entender a representação visual das marcações resultantes após execução do algoritmo para os três méto-

dos. Os pontos identificados pelo algoritmo são divididos nas cores amarela (VP, pontos que de fato são fibras de aço e o algoritmo identificou), vermelha (FN, pontos que de fato são fibras de aço, mas o algoritmo não identificou), e azul (FP, pontos marcados como sendo fibras de aço, mas na verdade são outros materiais que compõem a matriz cimentícia). O resultado do DoH apresenta bem mais pontos em vermelho (FN) na região central.

As Figuras 8 e 9 mostram os gráficos de Percentual de Acertos pela variação do *bin_coef*. O Percentual de Acertos é a razão entre a quantidade de pontos de fibras identificados corretamente (VP) e a quantidade de pontos de fibras relevantes. As legendas mostram a identificação dos operadores: LoG em azul, DoG em laranja e DoH em verde.

Analisando os gráficos conseguimos observar que as curvas dos operadores tem um comportamento semelhante se tratando da variação do coeficiente de binarização em cada imagem, porém divergem de uma imagem para outra. O gráfico da amostra CP07 apresenta curvas semelhantes ao gráfico da amostra CP23, e uma hipótese é a semelhança de cor do fundo entre essas imagens. Essas amostras possuem

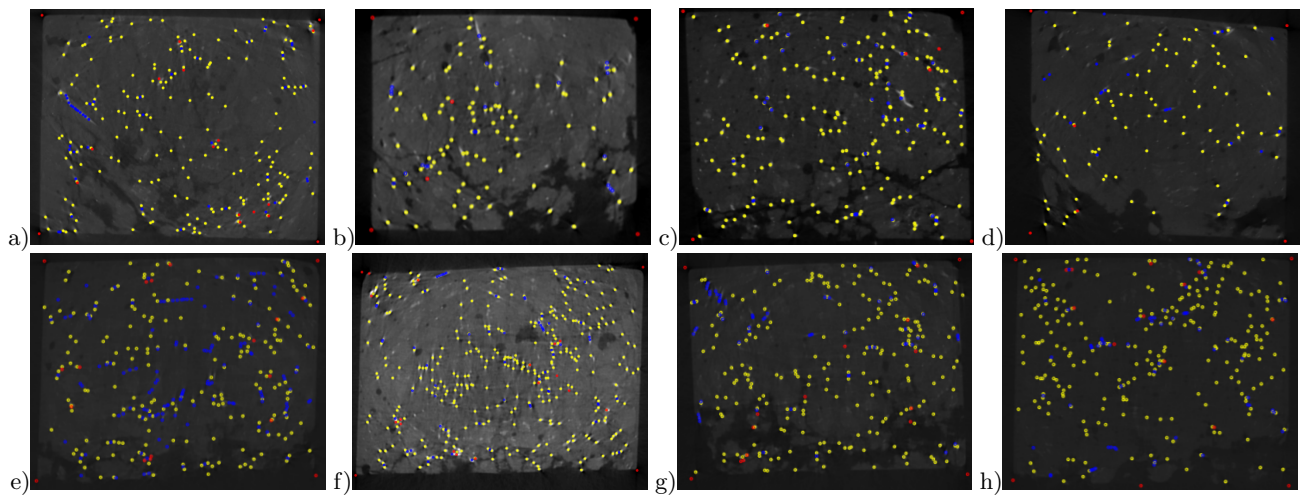


Figure 4: Imagens geradas com o melhor resultado do LoG. a) CP06, b) CP07, c) CP09, d) CP11, e) CP22, f) CP23, g) CP28, h) CP29

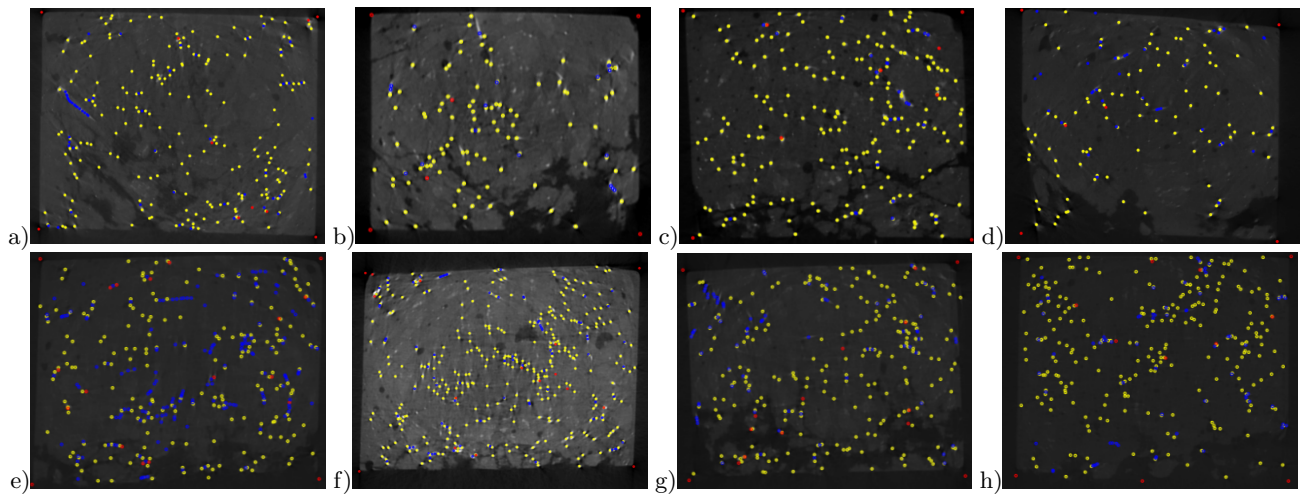


Figure 5: Imagens geradas com o melhor resultado do DoG. a) CP06, b) CP07, c) CP09, d) CP11, e) CP22, f) CP23, g) CP28, h) CP29

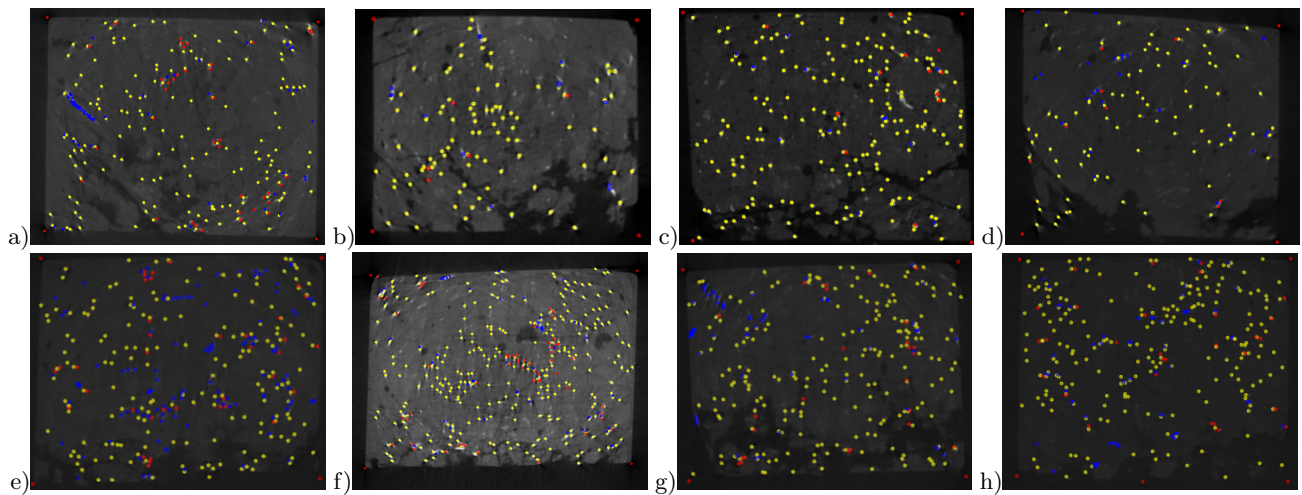


Figure 6: Imagens geradas com o melhor resultado do DoH. a) CP06, b) CP07, c) CP09, d) CP11, e) CP22, f) CP23, g) CP28, h) CP29

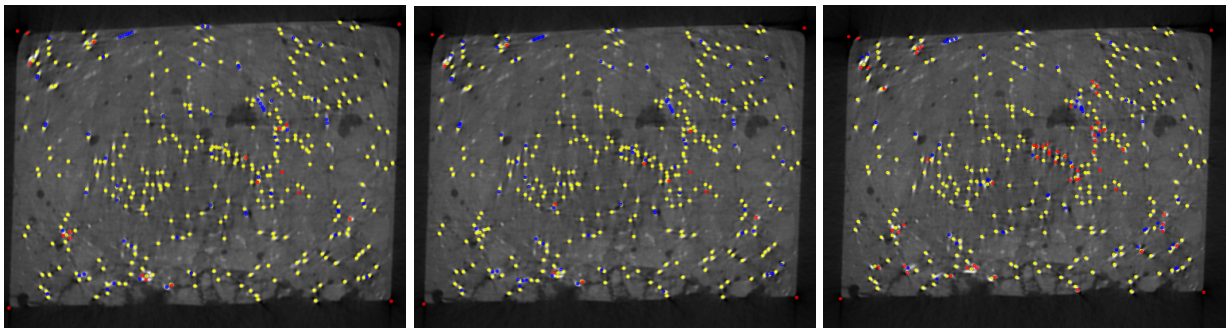


Figure 7: Amostra CP23 resultados LoG, DoG e DoH respectivamente

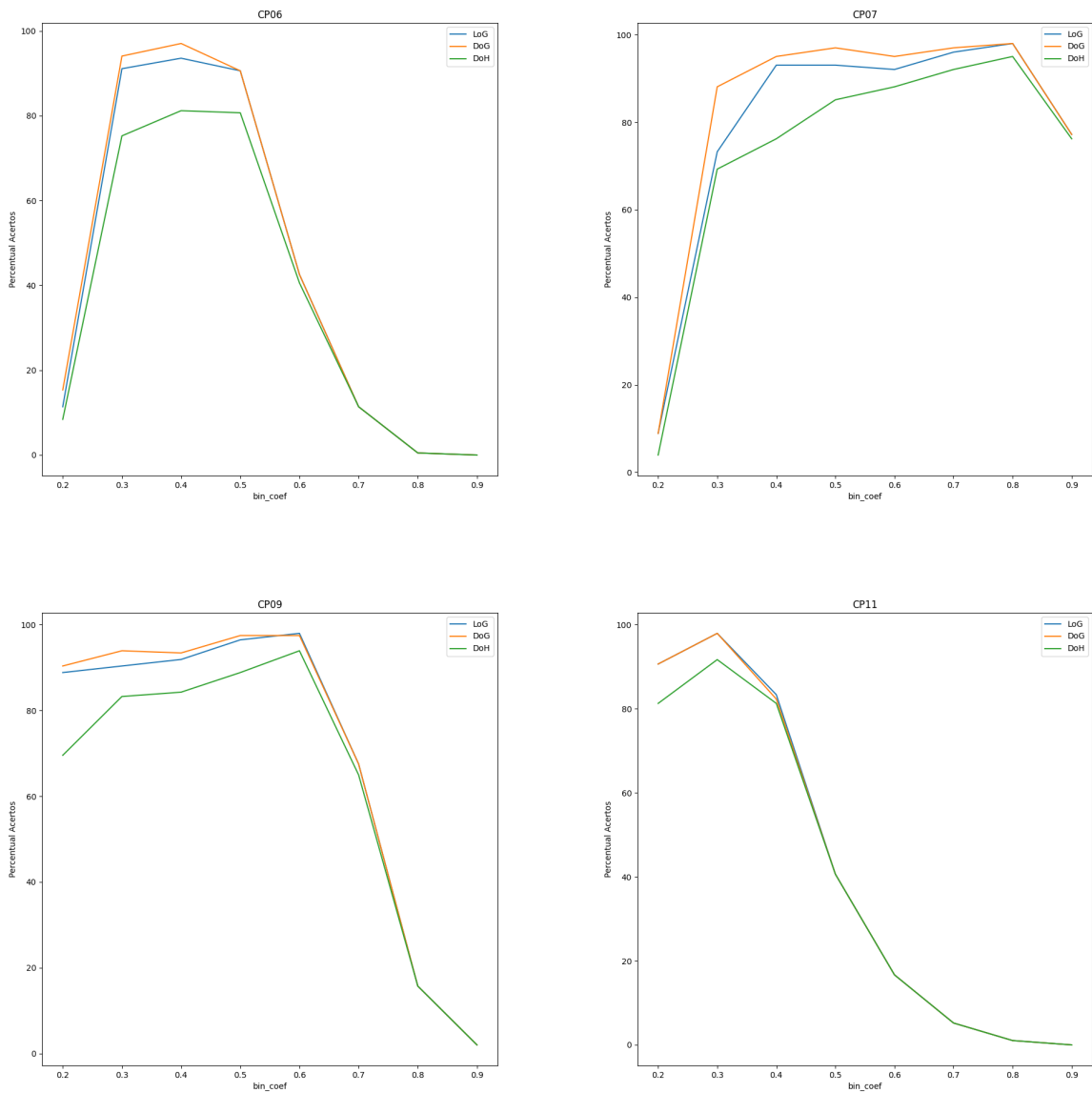


Figure 8: Gráficos Percentual de Acertos (CP06, CP07, CP09, CP11)

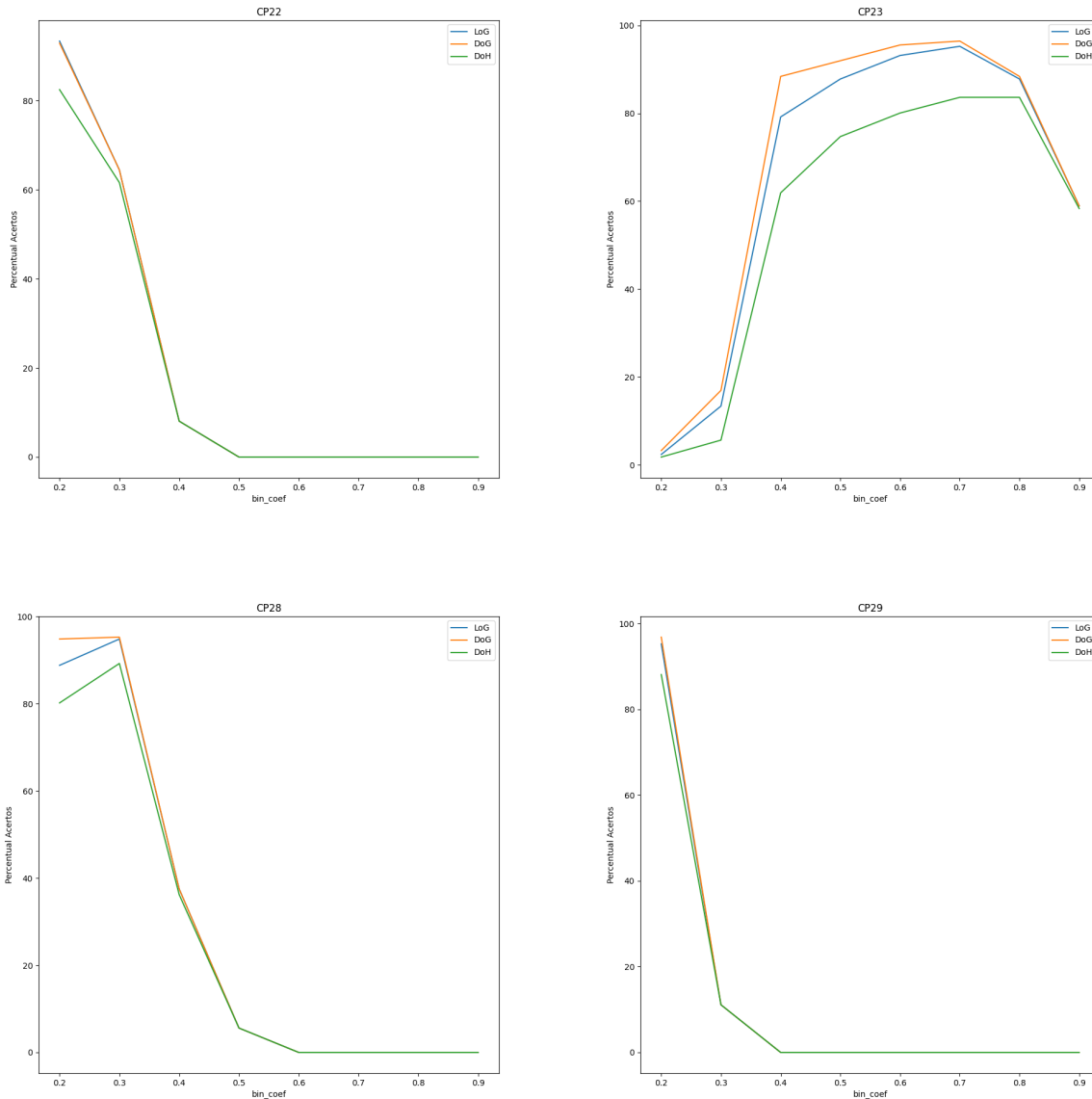


Figure 9: Gráficos Percentual de Acertos (CP22, CP23, CP28, CP29)

curvas que crescem com o aumento do *bin_coef* e decrescem ao chegar perto do valor 0,8, enquanto as imagens mais escuras possuem curvas que decrescem bem antes desse valor. Um exemplo é a amostra CP29, que tem os resultados piores após 0,3.

5. CONCLUSÃO

O objetivo deste trabalho foi o de realizar a comparação das técnicas de processamento de imagens usando os operadores LoG, DoG e DoH aplicadas na detecção, localização e contagem das fibras de reforço de uma imagem tomográfica de uma matriz cimentícia. Para isto as imagens de resultado foram geradas com marcações sobrepostas às imagens originais para que fosse mais fácil identificar VP, FP e FN.

Através da análise foi constatado que para as amostras

avaliadas o DoG se destaca em termos de VP perante os demais métodos, LOG e DoH. Além disso, a variação do coeficiente de binarização tem alto impacto no resultado, pois pode realçar os pontos de fibras presentes na imagem a depender de quão escura a imagem seja.

Para trabalhos futuros, deseja-se gerar um modelo 3D através de uma sequência de imagens obtidas de uma mesma matriz cimentícia e a possibilidade de exportação para o formato de uma ferramenta de CAD. Dessa forma, permitindo que o usuário possa ver as barras de reforço por completo dentro do material e calcular a resistência da matriz cimentícia. Também pretende-se utilizar redes neurais para aumentar a precisão da identificação das fibras.

Agradecimentos

A professora Karin Komati agradece ao CNPq pela Bolsa de Produtividade DT-2 (308432/2020-7) e pelo projeto de número 407742/2022-0, também agradece à FAPES pelo Auxílio Taxa de Pesquisa (nº 293/2021) e pelo projeto número 1023/2022 P:2022-8TZV6.

6. REFERENCES

- [1] L. Assirati, N. Rosa, L. Berton, A. Lopes, and O. Bruno. Performing edge detection by difference of gaussians using q-gaussian kernels. *Journal of Physics Conference Series*, 490, 11 2013.
- [2] G. L. Balázs, P. Bisch, A. Borosnyói, O. Burdet, C. Burns, F. Ceroni, V. Cervenka, M. A. Chiorino, P. Debernardi, L. Eckfeldt, M. El-Badry, E. Fehling, S. J. Foster, A. Ghali, V. Gribniak, M. Guiglia, G. Kaklauskas, R. J. Lark, P. Lenkei, M. Lorrain, A. Marí, J. Ozbolt, M. Pecce, A. Pérez Caldentey, M. Taliano, D. Tkalcic, J. M. Torrenti, L. Torres, F. Toutlemonde, T. Ueda, J. L. Vitek, and L. Vráblik. Design for sls according to fib model code 2010. *Structural Concrete*, 14(2):99–123, 2013.
- [3] S. Bhutada, N. Yashwanth, P. Dheeraj, and K. Shekar. Opening and closing in morphological image processing. *World Journal of Advanced Research and Reviews*, 14(3):687–695, 2022.
- [4] R. G. M. de Andrade. Estrutura pré-fabricada de concreto de alto desempenho reforçado com fibras de aço para pontilhões rodoviários, 2020.
- [5] R. G. M. de Andrade, M. S. Pfeil, R. C. Battista, R. D. Toledo Filho, O. M. O. de Araújo, and R. T. Lopes. Comparison between methods to determine the fibre orientation factor of an hpfr concrete bridge box girder. *Construction and Building Materials*, 269:121291, 2021.
- [6] A. D. de Figueiredo. Capítulo 37 – concreto com fibras. In G. C. Isaia, editor, *Concreto, Ensino, Pesquisa e Realizações*, chapter 37, pages 1194–1225. Ibracon, São Paulo, 2005.
- [7] K. De Vos. OpenStack: The Open Source Cloud Operating System, 2010.
- [8] Fédération Internationale du Béton. *fib Bulletin 55: Model Code for Concrete Structures 2010*. Ernst & Sohn, Berlin, Germany, 2010.
- [9] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (4th Edition)*. Prentice Hall, Upper Saddle River, NJ, 2007.
- [10] K. Grauman and B. Leibe. Chapter 3 – local features: detection and description. In *Visual Object Recognition*, pages 11–25. Springer, 2011.
- [11] A. Kaspers. Blob detection. Master’s thesis, Image Sciences Institute, UMC Utrecht, 2011.
- [12] H. Kong, H. C. Akakin, and S. E. Sarma. A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE transactions on cybernetics*, 43(6):1719–1733, 2013.
- [13] R. Lakemond, C. Fookes, and S. Sridharan. Affine adaptation of local image features using the hessian matrix. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 496–501, 2009.
- [14] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.
- [15] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.
- [16] G. Maroni, M. Ermidoro, F. Previdi, and G. Bigini. Automated detection, extraction and counting of acne lesions for automatic evaluation and tracking of acne severity. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–6. IEEE, 2017.
- [17] P. K. Mehta and P. J. M. Monteiro. *Concreto - Microestrutura, Propriedades e Materiais*. Ibracon, São Paulo, SP, 2 edition, 2014.
- [18] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.
- [19] F. L. Pedroso. Concreto: as origens e a evolução do material construtivo mais usado pelo homem. *Concreto & Construções*, 2009.
- [20] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pages 9–16, 2009.
- [21] B. F. d. N. d. Silva. Comparação de técnicas de correspondência de pontos chave para algoritmo orb, 2017.
- [22] I. J. A. Soares and L. O. M. Jr. Noise reduction using nonadditive Q-Gaussian filters in magnetic resonance images. In S. Ourselin and D. R. Haynor, editors, *Medical Imaging 2013: Image Processing*, volume 8669, page 86692J. International Society for Optics and Photonics, SPIE, 2013.
- [23] H. Xu, C. Lu, R. Berendt, N. Jha, and M. Mandal. Automatic nuclei detection based on generalized laplacian of gaussian filters. *IEEE journal of biomedical and health informatics*, 21(3):826–837, 2016.