



UNIVERSIDADE DE ÉVORA

Escola de Ciências e Tecnologias

Departamento de Informática

Sistema de informação avançado de caracterização de blocos de mármore por tratamento de imagens

Tiago Manuel Fragoso Conceição

Orientador: Mouhaydine Tlemcani

Orientadora: Irene Pimenta Rodrigues

Mestrado em Engenharia Informática

Ciência e Computação

Dissertação

Évora, 2019



UNIVERSIDADE DE ÉVORA

Sistema de informação avançado de caracterização de blocos de mármore por tratamento de imagens

Tiago Manuel Fragoso Conceição

Orientador: Mouhaydine Tlemcani

Orientadora: Irene Pimenta Rodrigues



Constituição de Júri:

Lígia Maria Rodrigues da Silva Ferreira - Presidente

Carlos Correia Ramos - Arguente

Irene Pimenta Rodrigues - Orientadora

Resumo

Sistema de informação avançado de caracterização de blocos de mármore por tratamento de imagens

Faturas e diaclases designa-se qualquer rutura de um maciço rochoso na qual não se verifica deslocamento apreciável de um bordo em relação ao outro. As fissuras macroscópicas distinguem-se das diaclases por não se apresentarem dispostas com regularidade, sendo em regra o seu desenvolvimento pequeno e a forma irregular. É frequente estas microfissuras só serem observáveis quando se fratura o material rochoso. Em oposição às diaclases, designa-se por falha, uma fratura em que houve deslocamento ao longo da superfície de rutura, no decurso da história geológica.

A solução para os problemas de mecânica das rochas implica o conhecimento:

1. Da deformabilidade dos maciços rochosos, isto é, das relações entre forças e deformações;
2. Da resistência dos maciços, ou seja, das condições que determinam a sua rutura;
3. Do estado de tensão inicial, a que se encontra submetido o maciço;
4. Dos estados de tensão que se desenvolvem nos maciços em virtude das solicitações aplicadas, entre as quais divididas à percolação.

Não existindo ainda soluções completas que substituam por completamente o perito, pretende-se criar um sistema que manipule as imagens enviadas para uma tomada de decisão, de forma a simplificar o trabalho do perito.

Abstract

Advanced Image Marble Block Characterization Information System

Fractures and diaclases are any rupture of a rock mass in which there is no appreciable displacement from one edge to the other. Macroscopic clefts are distinguished from diaclases because they are not regularly arranged, and as a rule their small development and irregular shape. Often these micro cracks are only observable when fracturing the rocky material. In opposition to diaclases, it is called a failure, a fracture in which there was displacement along the rupture surface during the course of geological history.

The solution to the problems of rock mechanics implies the knowledge:

1. The deformability of rock masses, that is, the relationships between forces and deformations;
2. The strength of the massifs, the conditions that determine their breakdown;
3. From the initial state of detention to which the massif is subjected;
4. The states of detention that develop in the massifs due to the requests applied, including divided into percolation.

Although there are no complete solutions that completely replace the expert, it is intended to create a system that manipulates the images sent for decision making, in order to simplify the expert's work.

Índice

1. Introdução.....	8
1.1. Estrutura da Tese	9
2.1. Detecção de descontinuidades.....	10
2. Estado da Arte	10
2.2. Canny Edge.....	12
2.2.1. Processo da deteção	13
2.2.2. Parâmetros	13
2.2.3. Áreas de utilização.....	14
2.2.4. Exemplos Práticos	15
2.3. Filtro Sobel	17
2.3.1. Implementação do Pseudocódigo	18
2.3.2. Exemplos práticos:	19
2.4. Filtro Laplace.....	21
2.4.1. Exemplos práticos:	22
2.5. Contexto na Vida Real.....	24
2.5.1. Reconhecimento automático de matrículas	24
2.5.2. Condução autónoma	24
2.5.3. Análise de imagens médicas.....	25
3.1. Desenvolvimento	27
3. Apresentação do Software	27
3.2. Fluxograma e Pseudocódigo da Aplicação.....	27
3.3. Aplicação Gráfica	30
3.3. Histograma.....	35
4. Avaliação.....	36
5. Conclusão e trabalho futuro.....	39
6. Bibliografia.....	39

Lista de Figuras

Figura 1 - Original	15
Figura 2 - Processada com <i>Canny Edge</i>	15
Figura 3 - Original	19
Figura 4 - Processada com Sobel.....	19
Figura 5 - Original	22
Figura 6 - Processada com Laplace	22
Figura 7 - Original	24
Figura 8 - Detecção de contornos com <i>Sobel</i>	24
Figura 9 - Original	25
Figura 10 - <i>Canny Edge</i>	25
Figura 11 - Original Ressonância Magnética	26
Figura 12 - Processada com Canny	26
Figura 13 - Fluxograma da Aplicação	27
Figura 14 - Interface Gráfica da Aplicação	30
Figura 15 - Aplicação - Painel 1 - Imagem Original	30
Figura 16 - Aplicação - Painel 2 - Imagem Suavizada com o filtro gaussiano	31
Figura 17 - Aplicação - Painel 3 - Canny	31
Figura 18 - Aplicação - Painel 4 - Sobel	32
Figura 19 - Aplicação - Painel 5 - Laplace.....	33
Figura 20 - Aplicação - Painel 6 - Sobel Heavy	34
Figura 21 - Avaliação - Painel 1.....	37
Figura 22 - Avaliação - Painel 2.....	38

1. Introdução

“A Rocha ou material rochoso é o material entre as superfícies de compartimentação de um maciço rochoso, isto é, o material dos blocos, completamente separados ou não, que essas superfícies determinam. O material rochoso não é contínuo apresentando em regra, além de fraturas macroscópicas e falhas, fraturas microscópicas e poros. Designamos as fraturas da rocha por fissuras e genericamente, por interstícios as fissuras e os poros.” (Rocha 2013)

Fraturas e diaclases designa-se qualquer rutura de um maciço rochoso na qual não se verifica deslocamento apreciável de um bordo em relação ou outro. As fissuras macroscópicas distinguem-se das diaclases por não se apresentarem dispostas com regularidade, sendo em regra o seu desenvolvimento pequeno e a forma irregular. É frequente estas microfissuras só serem observáveis quando se fratura o material rochoso. Em oposição às diaclases, designa-se por falha, uma fratura em que houve deslocamento ao longo da superfície de rotura, no decurso da história geológica.

As diaclases podem ocorrer perfeitamente fechadas ou abertas. No caso de serem abertas podem encontrar-se preenchidas por ar, água, minerais diversos como quartzo, calcite e dolomite, ou materiais granulares, mais ou menos argilosos. A grandeza da abertura e a natureza do enchimento influencia muito a deformabilidade e a resistência ao corte dos maciços, assim como as condições de circulação da água.

Nas diaclases sem enchimento, interessa considerar a rugosidade das paredes das diaclases, pois neste caso a resistência ao corte é muito influenciada pelo parâmetro em consideração.

As microfissuras têm em regra uma influência menos marcada do que as diaclases na deformabilidade e na resistência dos maciços rochosos. As microfissuras e os poros ocorrem mesmo nas rochas mais compactas, influenciando acentuadamente as suas propriedades, em particular as propriedades mecânicas e a permeabilidade.

As falhas das rochas podem ter grande influência na deformabilidade e na resistência dos maciços rochosos. As falhas podem originar o aparecimento de famílias de diaclases ou de superfícies de clivagem com atitudes associadas às falhas. Em regra, as falhas originam a perturbação e a destruição do maciço na sua vizinhança.

1. Introdução

A solução para os problemas de mecânica das rochas implica o conhecimento:

1. Da deformabilidade dos maciços rochosos, isto é, das relações entre forças e deformações;
2. Da resistência dos maciços, ou seja, das condições que determinam a sua rutura;
3. Do estado de tensão inicial, a que se encontra submetido o maciço;
4. Dos estados de tensão que se desenvolvem nos maciços em virtude das solicitações aplicadas, entre as quais divididas à percolação.

Para conhecer estes problemas pode-se recorrer a modelos reológicos que simulam os diversos comportamentos das rochas quando solicitados por tensões. Também por ensaios tais como o ensaio de carga na superfície, o ensaio em fenda aberta no maciço, ensaio com dilatómetro, ensaio de fluência, ensaio triaxial, ensaio de corte...

Não existindo ainda soluções completas que substituam por completo o perito, pretende-se criar um sistema capaz de o fazer autonomamente. Analisando e avaliando as amostras enviadas para uma tomada de decisão.

1.1. Estrutura da Tese

- **Capítulo 2: Estado da arte e descrição dos algoritmos** - Descrição de deteção de descontinuidades e algoritmos usados para a deteção de contornos, exemplos de uso prático dos algoritmos e a sua aplicação/utilidade no contexto da vida real;
- **Capítulo 3:** Apresentação do trabalho, pseudocódigo e demonstração do programa - Descrição do trabalho prático e demonstração do software.
- **Capítulo 4:** Avaliação e comparação de resultados - Avaliação de entradas diferentes comparação de resultados e análise de performance.
- **Capítulo 5:** Conclusão, trabalhos futuros e melhoramentos

2. Estado da Arte

2.1. Detecção de descontinuidades

A detecção de contornos inclui uma variedade de métodos matemáticos que visam identificar pontos numa imagem digital na qual o brilho da imagem muda acentuadamente ou, mais formalmente, tem descontinuidades, como descrito por (Lindeberg 1998). Os pontos nos quais o brilho da imagem muda drasticamente são tipicamente organizados num conjunto de segmentos de linhas curvas chamados de arestas. O mesmo problema de encontrar descontinuidades em sinais unidimensionais é conhecido como detecção de passos e o problema de encontrar descontinuidades de sinais ao longo do tempo é conhecido como detecção de alterações. A detecção de contornos é uma ferramenta fundamental no processamento de imagens, visão de máquina e visão computacional, particularmente nas áreas de detecção de recursos e extração de recursos.

O objetivo de detetar mudanças acentuadas no brilho da imagem é capturar eventos importantes e mudanças nas propriedades do mundo. Pode ser demonstrado que, sob suposições bastante gerais para um modelo de formação de imagem, as descontinuidades no brilho da imagem provavelmente corresponderão a:

1. Descontinuidades em profundidade;
2. Descontinuidades na orientação da superfície;
3. Mudanças nas propriedades do material;
4. Variações na iluminação da cena.

No caso ideal, o resultado da aplicação de um detetor de contornos a uma imagem pode levar a um conjunto de curvas ligadas que indicam os limites dos objetos, os limites das marcas de superfície e as curvas que correspondem às descontinuidades na orientação da superfície. Assim, aplicar um algoritmo de detecção de contornos numa imagem pode reduzir significativamente a quantidade de dados a serem processados, e, portanto, filtrar informações que possam ser consideradas menos relevantes, preservando as importantes propriedades estruturais de uma imagem. Se a etapa de detecção de contornos for bem-sucedida, a tarefa subsequente de interpretar o conteúdo da informação na imagem original pode, portanto, ser substancialmente simplificada. No entanto, nem sempre é possível obter essas margens ideais a partir de imagens da vida real de complexidade moderada.

Contornos extraídos de imagens não-triviais são frequentemente dificultados pela fragmentação, o que significa que as curvas de contornos não estão ligadas, segmentos de contornos ausentes e contornos falsos não correspondem a fenómenos interessantes na imagem - complicando assim a tarefa subsequente de interpretar os dados da imagem.

2. Estado da Arte

A detecção de contornos é uma das etapas fundamentais no processamento de imagens, análise de imagens, reconhecimento de padrões de imagem e técnicas de visão computacional.

Os contornos extraídos de uma imagem bidimensional de uma cena tridimensional podem ser classificados como dependentes ou independentes dependendo do ponto de vista. Um contorno independente do ponto de vista geralmente reflete as propriedades inerentes dos objetos tridimensionais, como as marcas de superfície e a forma da superfície. Uma aresta dependente do ponto de vista pode mudar à medida que o ponto de vista muda e, normalmente, reflete a geometria da cena, como objetos eclodindo-se um ao outro.

Uma aresta típica pode, por exemplo, ser a fronteira entre um bloco de cor vermelha e um bloco de amarelo. Em contraste, uma linha (como pode ser extraída por um detetor de relevo) pode ser um pequeno número de pixéis de uma cor diferente em um fundo de outra maneira inalterável. Para uma linha, normalmente pode haver uma borda em cada lado da linha.

Embora certas literaturas considerem a detecção de contornos de degraus ideais, os contornos obtidos em imagens naturais geralmente não são de todo contornos de degrau ideais. Em vez disso, eles são normalmente afetados por um ou vários dos seguintes efeitos:

1. Desfoco causado por uma função finita de profundidade de campo e de ponto finito.
2. Desfoco penumbral causado por sombras criadas por fontes de luz de raio diferente de zero.
3. Sombras em um objeto suave

Em algumas abordagens foram usados contornos de degrau gaussiano suavizado (uma função de erro) como a extensão mais simples do modelo de contornos de degrau ideal para modelar os efeitos de desfoco de contorno em aplicações práticas.

Existem muitos métodos para detecção de contornos, mas a maioria deles pode ser agrupado em duas categorias, baseada em pesquisa e baseada em cruzamento zero. Os métodos baseados em pesquisa detetam contornos calculando primeiro uma medida da força do contorno, geralmente uma expressão derivada de primeira ordem, como a magnitude do gradiente, e pesquisando os máximos direcionais locais da magnitude do gradiente usando uma estimativa calculada da orientação local do contorno, geralmente a direção do gradiente. Os métodos baseados em cruzamento de zero pesquisam cruzamentos de zero numa expressão derivada de segunda ordem calculada a partir da

2. Estado da Arte

imagem a fim de encontrar contornos, geralmente o operador *Laplaciano* ou os cruzamentos de zero de uma expressão diferencial não linear. Como etapa de pré-processamento para a detecção de contornos, quase sempre é aplicado um estágio de suavização, geralmente suavização gaussiana (redução de ruído).

Os métodos de detecção de contornos que foram publicados diferem principalmente nos tipos de filtros de suavização aplicados e na maneira como as medidas da resistência de contornos são calculadas. Como muitos métodos de detecção de contornos dependem do cálculo de gradientes da imagem, eles também diferem nos tipos de filtros usados para calcular as estimativas de gradiente nas direções x e y .

2.2. Canny Edge

O *Canny Edge* é um algoritmo de detecção de contornos introduzido por Canny J. em (Canny 1986) que extrai informações estruturais úteis de diferentes objetos obtidos com o objetivo de reduzir drasticamente a quantidade de dados a serem processados.

Este detetor de contornos, *Canny*, é amplamente utilizado em visão computacional para localizar mudanças de intensidade acentuadas e encontrar limites de objetos numa imagem. O detetor *Canny* classifica um pixel como um limite se a magnitude do gradiente do pixel atual for maior que a dos pixels em ambos os lados na direção da mudança de intensidade máxima. Os critérios gerais para a detecção de um contorno consistem em:

1. Baixa taxa de erro, a detecção deve apresentar com precisão o máximo de contornos possíveis;
2. O ponto do contorno detetado pelo operador deve localizar com precisão o centro do contorno.
3. O contorno só deve ser identificado uma vez, quando possível, e o ruído da imagem não deve criar contornos falsos.

Para satisfazer esses requisitos, *Canny* usou o cálculo das variações - uma técnica que encontra a função que otimiza um dado funcional. A função ideal no detetor de *Canny* é descrita pela soma de quatro termos exponenciais, mas pode ser aproximada pela primeira derivada de um gaussiano.

Entre os métodos de detecção de contornos desenvolvidos até agora, o algoritmo de detecção de *Canny* é um dos métodos estritamente mais bem definido que fornece uma detecção boa e confiável, devido à sua otimização para atender aos três critérios de detecção de contornos e à simplicidade do processo de implementação, este tornou-se um dos algoritmos mais populares para detecção de contornos devido a esses aspetos.

2. Estado da Arte

2.2.1. Processo da detecção

1. **Redução de ruído:** Como a detecção de contornos é suscetível ao ruído, é necessário remover o ruído na imagem usando um filtro gaussiano 5x5.
2. **Procurar a intensidade do gradiente da imagem:** Após a redução do ruído a imagem é filtrada com o *Sobel Kernel* (Irwin 2014) em ambas as direções (horizontal e vertical) para obter a primeira derivada na direção horizontal (G_x) e na direção vertical (G_y). A partir dessas duas imagens podemos encontrar o gradiente e a direção do contorno para cada pixel da seguinte maneira:

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2}$$
$$Angle(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

A direção do gradiente é sempre perpendicular aos contornos.

3. **Supressão:** Após obter a magnitude e a direção do gradiente, é feita uma pesquisa completa na imagem para remover os pixels indesejados que não constituam um contorno.
4. **Limite de histerese:** Esta fase decide quais os contornos válidos. Para isso, é necessário especificar dois valores, *minVal* e *maxVal*. Todos os contornos com gradiente de intensidade superior a *maxVal* serão aceites e os abaixo de *minVal* não serão considerados um contorno válido. Os que se encontram entre os dois valores são contornos classificados ou não-contornos com base na sua ligação. Se estiverem ligados a *pixels* "seguros", são considerados parte dos contornos. Caso contrário, serão descartados.

2.2.2. Parâmetros

O algoritmo *Canny* contém vários parâmetros aplicáveis, que podem afetar a duração de processamento e a eficiência do algoritmo.

1. **O tamanho do filtro gaussiano:** O filtro de suavização usado no primeiro estágio afeta diretamente os resultados do algoritmo de *Canny*. Filtros menos agressivos causam mais foco e permitem assim a detecção de linhas pequenas e mais nítidas. Um filtro agressivo pelo contrário, causa menos nitidez, distorcendo o valor de um determinado pixel numa área maior da imagem. Áreas de desfoco maiores são

2. Estado da Arte

mais úteis para detetar contornos maiores e mais suaves, por exemplo, o contorno de um arco-íris.

2. **Limites de histerese:** O uso de dois limites com histerese permite mais flexibilidade do que numa abordagem com um único limite. Um limite elevado pode perder informações relevantes, por outro lado, um limite demasiado reduzido irá identificar incorretamente informações irrelevantes (como ruído) sendo importantes. É impossível fornecer um limite universal que funcione bem em todos os casos e sendo ainda não há uma abordagem experimentada e testada para resolver este problema.

2.2.3. Áreas de utilização

O algoritmo *Canny Edge* é adaptável a vários envolventes. Os seus parâmetros possibilitam que seja adaptado para o reconhecimento de contornos com diferentes características, dependendo dos requisitos específicos de uma determinada implementação.

No artigo original de *Canny* em (Canny 1986), a derivação do filtro ótimo levou a um filtro de "Resposta de Impulso Finito", que pode ser demorado para calcular no domínio espacial se a quantidade de suavização necessária for relevante (o filtro terá um grande suporte espacial nesse caso). Por este motivo, é frequentemente sugerido usar a forma de resposta ao impulso infinito de Rachid Deriche do filtro de *Canny* (o detector de Canny-Deriche), que é recursivo, e que pode ser calculado em um período curto e fixo para qualquer quantidade desejada de suavização. A segunda forma é adequada para implementações em tempo real em FPGAs¹ ou DSPs², ou em PCs muito rápidos. Neste contexto, a implementação recursiva inicial do operador *Canny* não fornece uma boa aproximação da simetria rotacional e, portanto, fornece um viés para as margens horizontais e verticais.

¹ Field-programmable gate array

² Digital Signal Processor

2. Estado da Arte

2.2.4. Exemplos Práticos

Figura 1 - Original

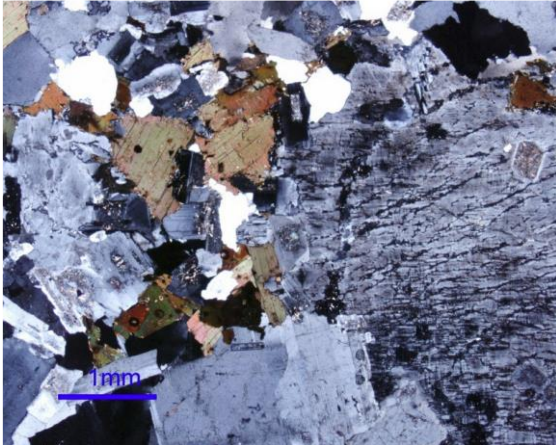
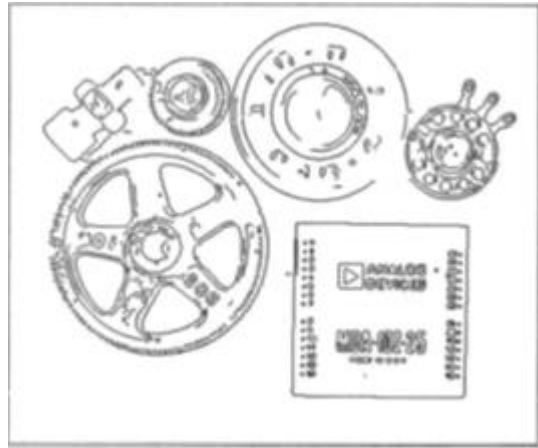
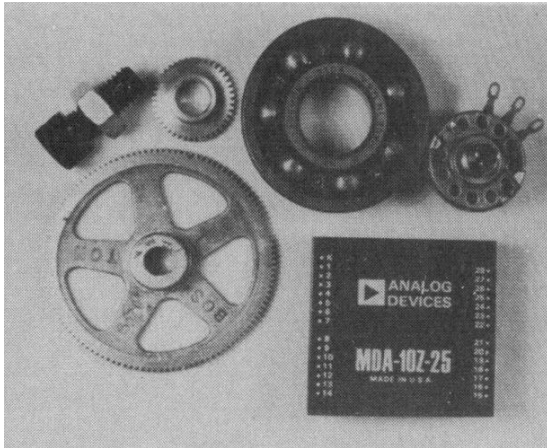
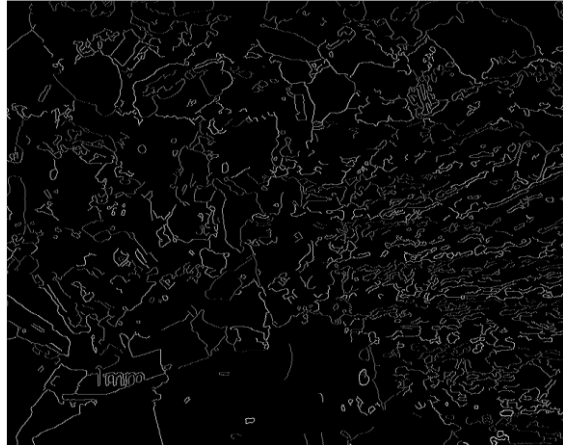
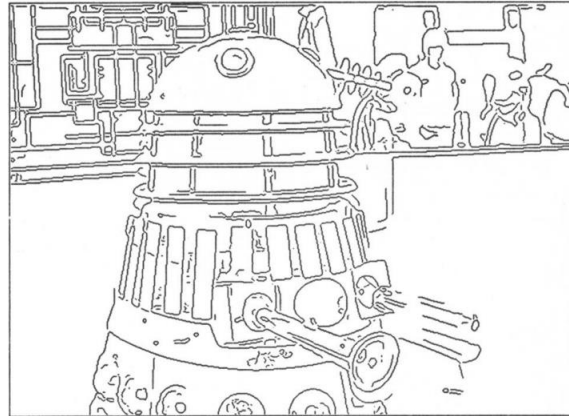
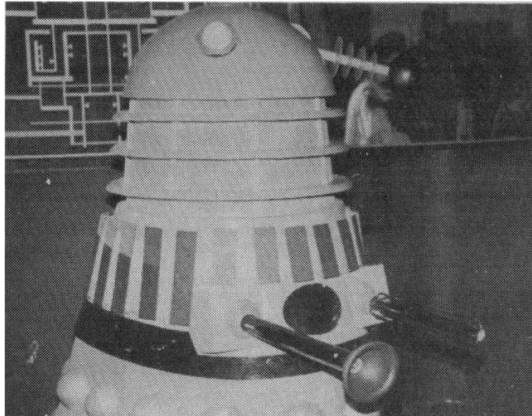


Figura 2 - Processada com *Canny Edge*



2. Estado da Arte



Nas figuras anteriores observamos a aplicação e transformação real do algoritmo *Canny Edge*, este começa por aceitar uma imagem "original", esta é sujeita a um filtro de tons de cinzento prévio para eliminar o ruído e suavizar os pixéis, este filtro transforma a imagem em escalas de cinzento para simplificar o seu processamento posterior com o atual algoritmo de *Canny Edge*. Após isso, o algoritmo faz uma pesquisa por pixéis vizinhos usando parâmetros de "threshold1 e threshold2 [Limite de histerese]", esses parâmetros são usados para determinar se um contorno é válido ou não baseado na sua diferença, no caso de ser válido e constituir um contorno, este é adicionado à lista do resultado que representará um pixel na sua localização de descoberta. O resultado final da aplicação do algoritmo pode ser observado em "Processada com *Canny Edge*", que após ter o mapa dos contornos a imagem é refeita apenas com os contornos encontrados, e tudo o resto é descartado.

2. Estado da Arte

2.3. Filtro Sobel

Sobel é um operador discreto de diferenciação (Irwin 2014), que calcula uma aproximação do gradiente de uma função de intensidade de imagem.

“O filtro *Sobel* calcula o gradiente da intensidade da imagem em cada ponto, dando a direção da maior variação de claro para escuro, e a quantidade de variação nessa direção, assim, obtém-se uma noção de como varia a luminosidade em cada ponto, de forma mais suave ou abrupta.” - (Sobel operator 2019)

“Com isto consegue-se estimar a presença de uma transição claro-escuro e de qual a orientação desta. Como as variações claro-escuro intensas correspondem a fronteiras bem definidas entre objetos, consegue-se fazer a detecção de contornos.” - (Sobel operator 2019)

“Matematicamente este operador utiliza duas matrizes 3x3 que são convoluídas com a imagem original para calcular aproximações das derivadas - uma para as variações horizontais e uma para as verticais. Sendo \mathbf{A} a matriz inicial então, \mathbf{G}_x e \mathbf{G}_y serão duas matrizes que em cada ponto contêm uma aproximação às derivadas horizontal e vertical de A .” - (Sobel operator 2019)

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \quad e \quad \mathbf{G}_y = \begin{bmatrix} 1 & +2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

Portanto a magnitude, \mathbf{G} , e a direção Θ do gradiente são dados por:

$$\mathbf{G} = \sqrt{G_x^2 + G_y^2} \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

2. Estado da Arte

2.3.1. Implementação do Pseudocódigo

```
// Retirado em 27/09/2019 de: https://en.wikipedia.org/wiki/Sobel\_operator
function sobel(A : as two dimensional image array)
    Gx=[-1 0 1; -2 0 2; -1 0 1]
    Gy=[-1 -2 -1; 0 0 0; 1 2 1]

    rows = size(A,1)
    columns = size(A,2)
    mag=zeros(A)

    for i=1:rows-2
        for j=1:columns-2
            S1=sum(sum(Gx.*A(i:i+2,j:j+2)))
            S2=sum(sum(Gy.*A(i:i+2,j:j+2)))

            mag(i+1,j+1)=sqrt(S1.^2+S2.^2)
        end for
    end for

    threshold = 70 %varies for application [0 255]
    output_image = max(mag, threshold)
    output_image(output_image == round(threshold))=0;
    return output_image

end function
```

Como a função de intensidade de uma imagem digital só é conhecida em pontos discretos, as derivadas dessa função não podem ser definidas, a menos que consideremos que exista uma função de intensidade contínua subjacente amostrada nos pontos da imagem. Com algumas suposições adicionais, a derivada da função de intensidade contínua pode ser calculada como uma função na função de intensidade amostrada, ou seja, a imagem digital. Acontece que as derivadas em qualquer ponto particular são funções dos valores de intensidade virtualmente de todos os pontos de imagem. No entanto, aproximações dessas funções derivativas podem ser definidas em graus menores ou maiores de precisão.

2. Estado da Arte

O operador *Sobel-Feldman* representa uma aproximação bastante imprecisa do gradiente de imagem, no entanto, ainda tem qualidade suficiente para ter uso prático em muitas aplicações. Mais precisamente, ele usa valores de intensidade apenas numa região 3×3 em torno de cada ponto de imagem para aproximar o gradiente de imagem correspondente e usa apenas valores inteiros para os coeficientes que medem as intensidades da imagem para produzir a aproximação do gradiente.

2.3.2. Exemplos práticos:

Figura 3 - Original

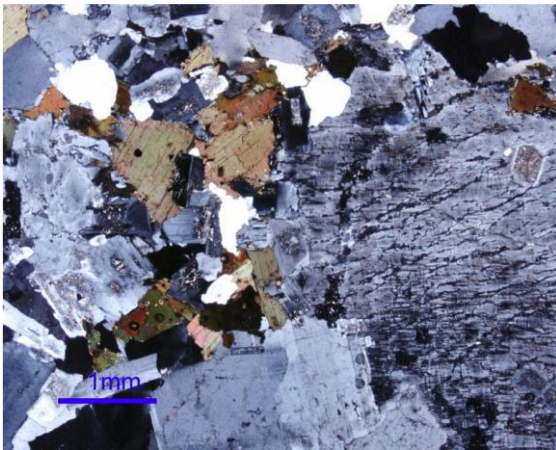
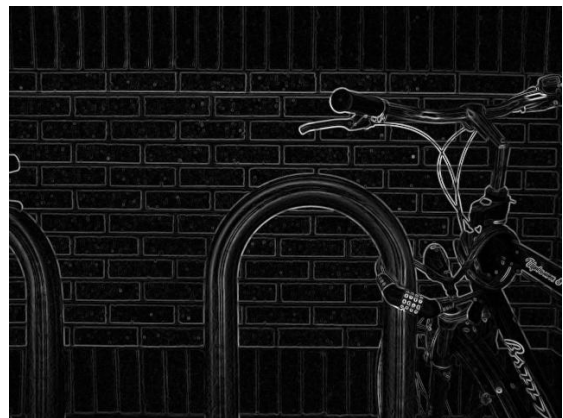
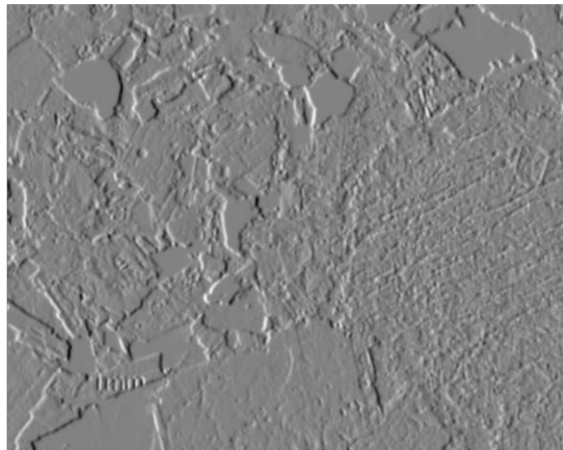
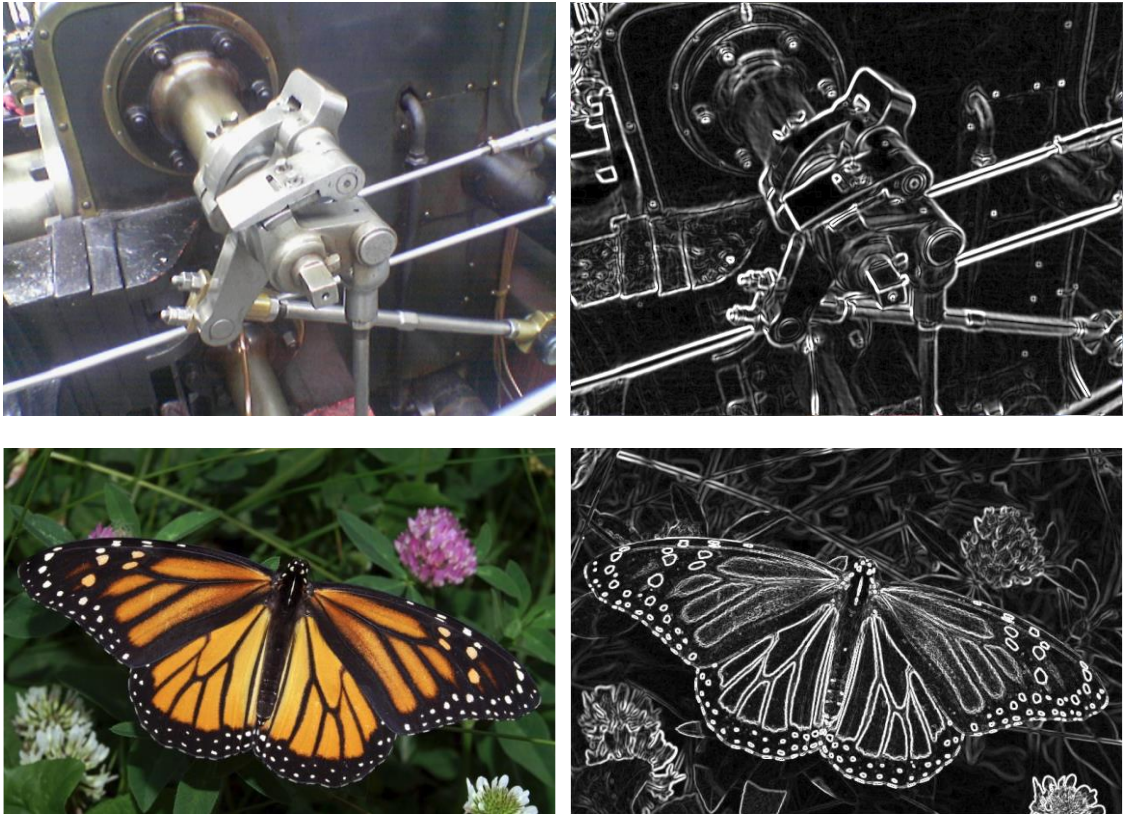


Figura 4 - Processada com Sobel



2. Estado da Arte



Nas figuras anteriores observamos a aplicação e transformação real do filtro *Sobel*, este começa por aceitar uma imagem "original", esta é sujeita a um filtro de tons de cinzento prévio para eliminar o ruído e suavizar os pixéis, este filtro transforma a imagem em escalas de cinzento para simplificar o seu processamento posterior com o atual algoritmo de *Sobel*. O resultado do operador *Sobel-Feldman* é um mapa bidimensional do gradiente em cada ponto. Pode ser processado e visualizado como se fosse uma imagem, com as áreas de alto gradiente (os contornos prováveis) visíveis como linhas brancas.

2. Estado da Arte

2.4. Filtro Laplace

O operador Discreto de *Laplace* é frequentemente usado no processamento de imagem, na detecção de contornos e aplicações de estimativa de movimento. Este é definido como a soma das expressões de coordenadas do operador segundo Laplace, e calculado como soma de diferenças sobre os vizinhos mais próximos do pixel central.

O operador é definido segundo a expressão:

$$Laplace(f) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} f$$

Laplace é um operador derivado como referido matematicamente acima, o seu uso destaca as discontinuidades na escala de cinzentos numa imagem, tentando assim diminuir a ênfase em regiões com níveis de cinza que variam lentamente. Essa operação resulta em imagens que possuem linhas de contornos acinzentados e outras discontinuidades num fundo escuro. Isso produz contornos internos e externos numa imagem.

O operador Laplaciano positivo e negativo não podem ser aplicados na mesma imagem. Apenas se pode aplicar um, se for aplicado o operador laplaciano positivo na imagem, subtraímos a imagem resultante da imagem original para obter a imagem mais nítida. Da mesma forma, se aplicarmos o operador Laplaciano negativo, temos que adicionar a imagem resultante à imagem original para obter a imagem mais nítida.

2. Estado da Arte

2.4.1. Exemplos práticos:

Figura 5 - Original

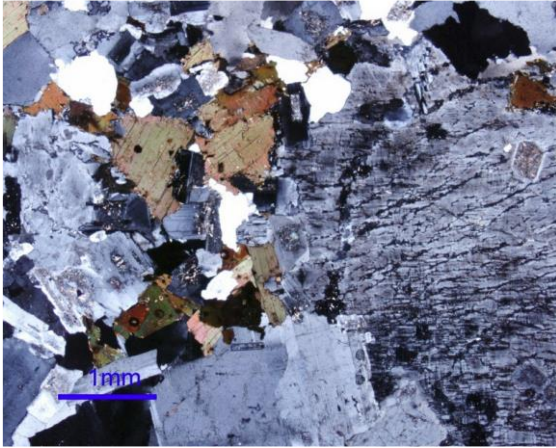
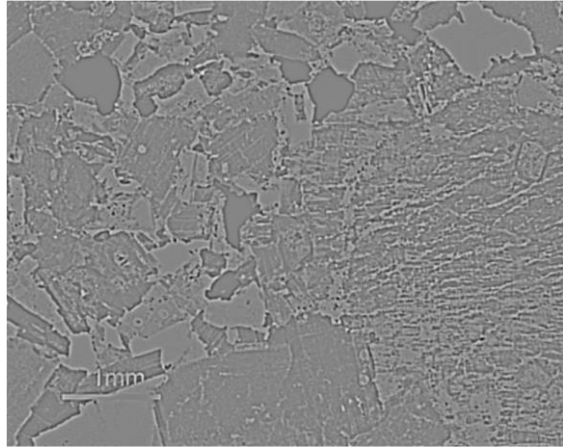


Figura 6 - Processada com Laplace



2. Estado da Arte



2. Estado da Arte

2.5. Contexto na Vida Real

2.5.1. Reconhecimento automático de matrículas

O reconhecimento automático de matrículas, é uma aplicação que utiliza reconhecimento ótico de caracteres em imagens para ler matrículas de veículos para criar dados de localização de veículos. Alguns sistemas usam técnicas de detecção de contornos para melhorar o reconhecimento entre as letras e o fundo da placa.

Figura 7 - Original



Figura 8 - Detecção de contornos com Sobel



2.5.2. Condução autónoma

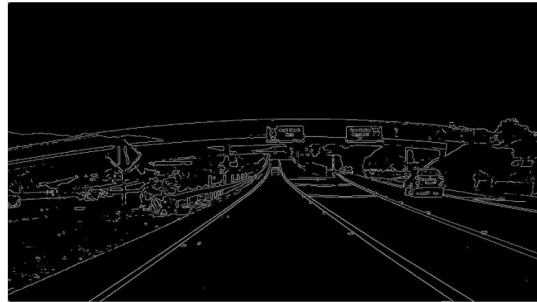
Um carro autónomo é um veículo capaz de sentir o ambiente à sua volta e mover-se com pouca ou nenhuma interação humana. Estes combinam uma variedade de sensores para perceber o mundo, como radar, visão computacional, lidar, sonar, GPS, odometria e unidades de medida inerciais. Os sistemas avançados de controlo interpretam informações sensoriais para identificar caminhos de navegação apropriados, bem como obstáculos e sinalização relevante. Por isso, a deteção de contornos tem um papel fulcral nestes sistemas, onde é necessário avaliar estradas, sinais de trânsito e outras linhas importantes para uma tomada de decisão.

2. Estado da Arte

Figura 9 - Original



Figura 10 - *Canny Edge*



2.5.3. Análise de imagens médicas

A detecção de contornos é geralmente usada em tipos de imagens médicas, como exames de ressonância magnética como exemplo (Abbass and Mousa 2017). Em uma ressonância magnética, o tecido que tem menos átomos de hidrogénio (como ossos) fica escuro, enquanto o tecido que tem muitos átomos de hidrogénio (como o tecido adiposo) fica muito mais brilhante. Assim os valores de intensidade dos pixéis em exames de ressonância magnética são calculados. A detecção de contornos de imagens médicas é um método importante no reconhecimento dos órgãos humanos e é uma etapa importante de pré-processamento na reconstrução 3D, como a reconstrução de imagens cerebrais. O objetivo principal de um detetor de contornos é produzir os contornos necessários para outros estágios de processamento de imagem, como detetar o objeto, sua forma e seu tamanho. O algoritmo *Canny* é geralmente preferível ao *Sobel*, pois mantém uma estrutura mais relevante em relação ao *Sobel*, este mostra apenas os contornos relevantes.

2. Estado da Arte

Figura 11 - Original Ressonância Magnética

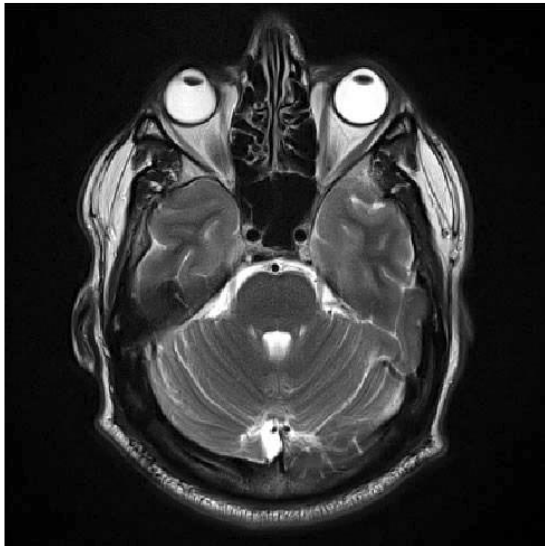


Figura 12 - Processada com Canny



Estes são alguns dos exemplos da utilização dos algoritmos de detecção de contornos na vida real. A sua utilidade e uso são quase ilimitados, estes sistemas surgem por necessidade e pela procura de solucionar problemas na inteligência artificial, no entanto, também podem ser utilizados para ajudar/facilitar o utilizador a perceber melhor uma perspetiva, eliminando ruído e detalhes descartáveis à nossa visão. Para além disso é também usado para a reconstrução de imagens e até mesmo para o melhoramento e aplicação de filtros em imagens para realçar detalhes e/ou relevos, dando ênfase aos objetos na imagem, e suavizando o fundo.

Na aplicação em estudo deste artigo, todos os algoritmos acima descritos serão usados para reproduzir uma representação de contornos a partir de uma imagem (Planos de rochas), essas representações servirão para auxiliar o utilizador a tomar uma decisão sobre a avaliação do plano (Descontinuidades e áreas). Os diferentes algoritmos representam um papel importante, um pode ser mais adequado que outro em determinada imagem (tipos de rochas diferentes), assim o utilizador terá a liberdade de analisar várias representações ao mesmo tempo, todas elas sincronizadas entre elas para que se possa fazer uma comparação de 1:1, sem recorrer a técnicas de multi-janelas e de pesquisa manual.

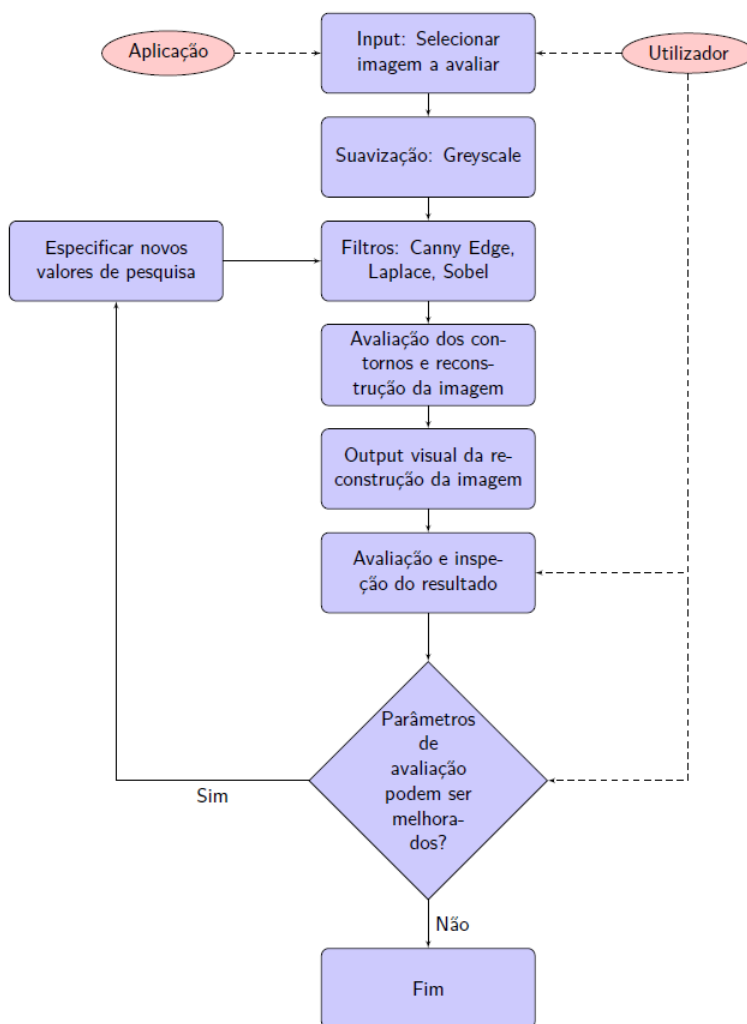
3. Apresentação do Software

3.1. Desenvolvimento

A aplicação foi desenvolvida na linguagem C# (CSharp), versão 7.0, sobre o Microsoft .NET Framework 4.6.2. O editor de código (IDE) usado para programar a aplicação foi o Visual Studio 2017. Todo o código e repositório da aplicação pode ser encontrado em: <https://github.com/sn4k3/MineralAnalysis>

3.2. Fluxograma e Pseudocódigo da Aplicação

Figura 13 - Fluxograma da Aplicação



3. Apresentação do Software

O caso de uso da aplicação (fluxograma) inicia com uma janela vazia, aguarda ações por parte do utilizador, sendo esta, seleccionar uma imagem para ser processada e filtrada ("Input: Seleccionar imagem a avaliar"). Após a escolha a aplicação faz o processamento da imagem que consiste em: 1. "Suavização: Greyscale", este filtro torna a imagem original numa imagem em tons de cinzento para suavizar e simplificar a imagem, e também para remover algum ruído. A partir dessa imagem são aplicados os algoritmos acima descritos: "Canny Edge, Laplace, Sobel" com os parâmetros específicos de cada algoritmo especificados pelo utilizador. Após cada aplicação dos algoritmos, é feita uma representação dos contornos em forma de imagem, recriada a partir dos algoritmos e da informação resultante da sua aplicação. Todas essas imagens são apresentadas na janela da aplicação em forma de grelha para uma fácil comparação entre elas. Por fim cabe ao utilizador avaliar e usar o resultado obtido ("Avaliação e inspeção do resultado"), caso o resultado seja indesejado ou incorreto, podem ser definidos parâmetros diferentes para o algoritmo de forma a melhorar a sua precisão ("Especificar novos valores de pesquisa"), até ter o melhor resultado.

Pseudocódigo da aplicação

```
image      = LoadImage(Path)
imageInfo  = GetImageInfo(image)
histogram  = GenerateHistogram(image)

imageGray  = GrayScale(image)
imageCanny= Canny(imageGray, threshold1, threshold2, L2gradient)
imageSobel = Sobel(imageGray, xOrder, yOrder, apertureSize)
imageLaplace    = Laplace(imageGray, apertureSize)
imageHeavySobel = Sobel(imageGray, 0, 1, 3).AbsDiff().Gray()

ShowImages(image, imageGray, imageCanny, imageSobel,
            imageLaplace, imageHeavySobel)
ShowImageInfo(imageInfo)
ShowImageHistogram(histogram)
```


3. Apresentação do Software

O pseudocódigo acima descrito representa a parte principal e funcional da aplicação, que consiste em:

1. Carregar a imagem para a memória para ser manipulada;
2. Obter informações genéricas sobre a imagem, tal como: Tamanho, resolução, datas, etc
3. Gerar o histograma de cores a partir da imagem;
4. Criar uma variante da imagem a tons de cinzento para suavização;
5. Aplicar o filtro Canny sobre a imagem suavizada (4.) e recriar uma representação com base na informação dos contornos obtidos;
6. Aplicar o filtro Sobel sobre a imagem suavizada (4.) e recriar uma representação com base na informação dos contornos obtidos;
7. Aplicar o filtro Laplace sobre a imagem suavizada (4.) e recriar uma representação com base na informação dos contornos obtidos;
8. Aplicar o filtro Sobel Heavy sobre a imagem suavizada (4.) e recriar uma representação com base na informação dos contornos obtidos;
9. Mostra todas as imagens obtidas dispostas em grelha numa janela;
10. Mostra as informações genéricas da imagem na janela (2.);
11. Mostra o histograma de cores da imagem original na janela (3.).

3. Apresentação do Software

3.3. Aplicação Gráfica

Figura 14 - Interface Gráfica da Aplicação

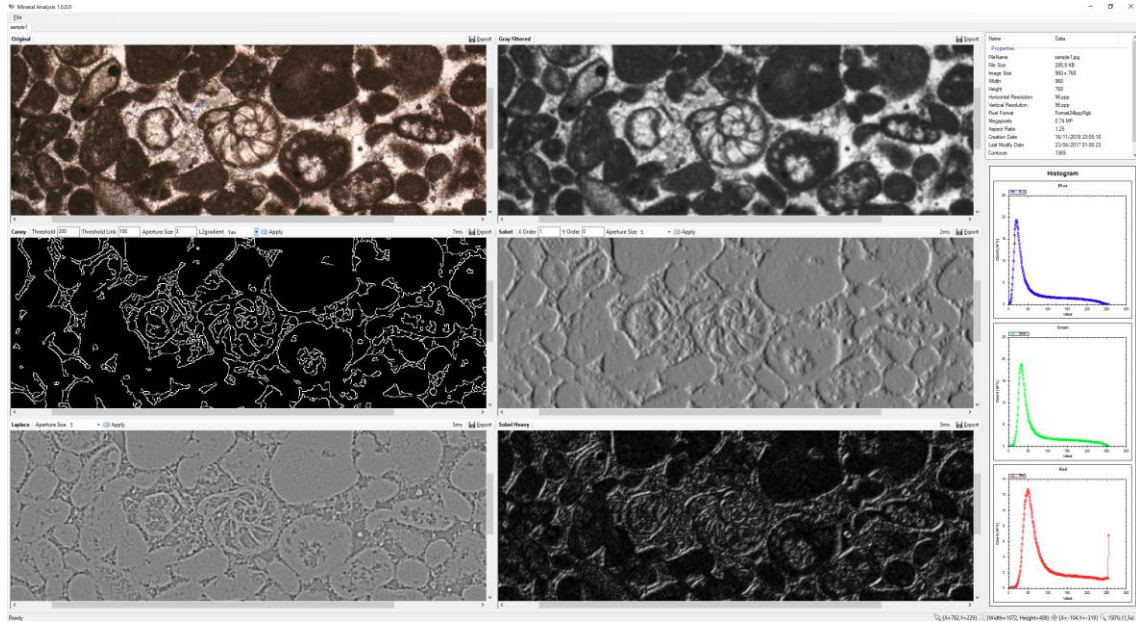
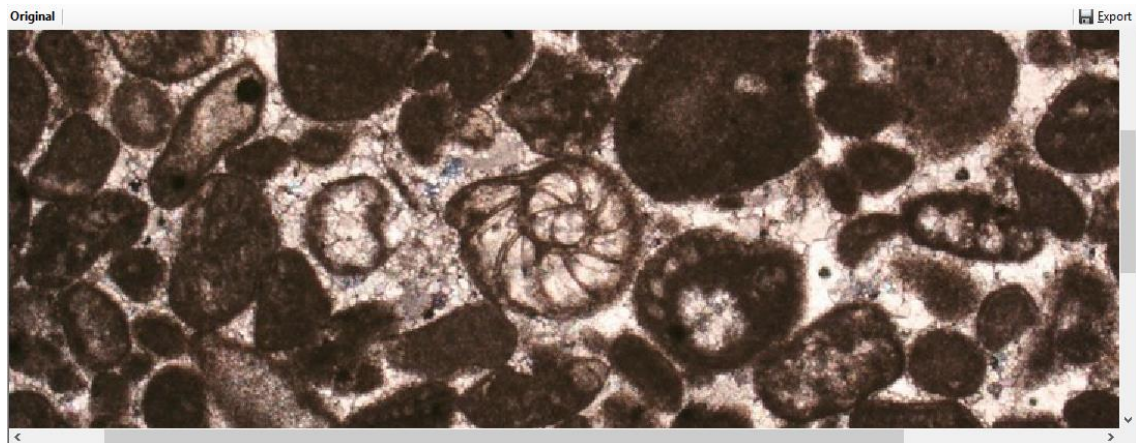


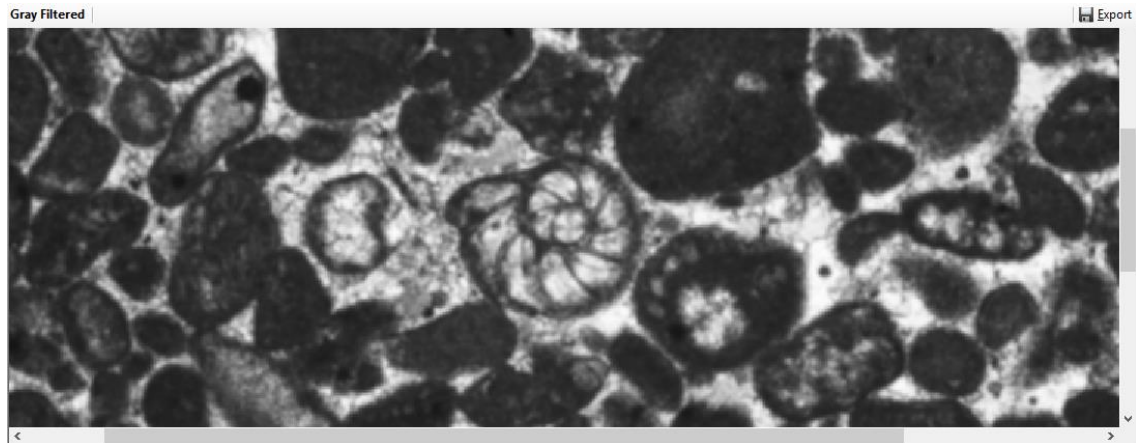
Figura 15 - Aplicação - Painel 1 - Imagem Original



Neste painel é apresentada a imagem original, sem filtros ou tratamentos.

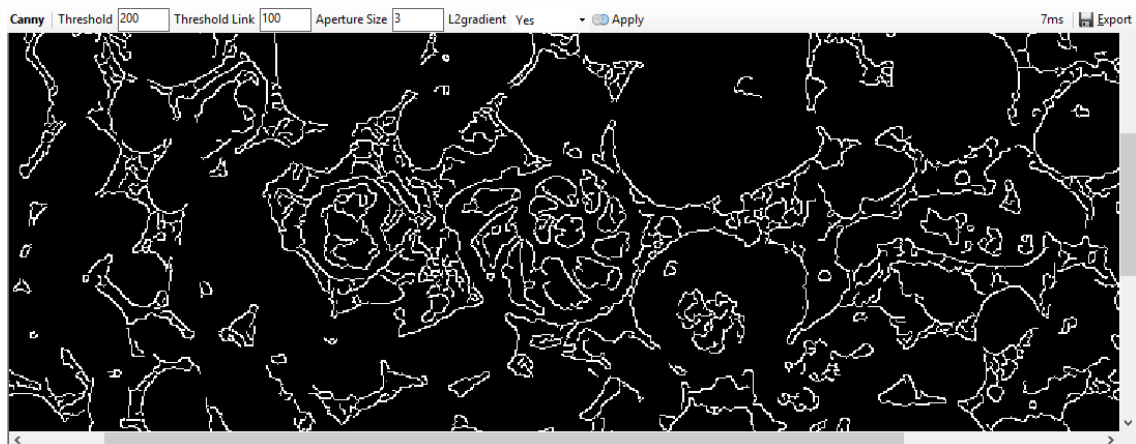
3. Apresentação do Software

Figura 16 - Aplicação - Painel 2 - Imagem Suavizada com o filtro gaussiano



A partir da imagem original, esta é duplicada e aplica-se um filtro gaussiano de *grey scale* apenas para suavizar a imagem e reduzir o ruído. A imagem filtrada é apresentada neste painel.

Figura 17 - Aplicação - Painel 3 - Canny



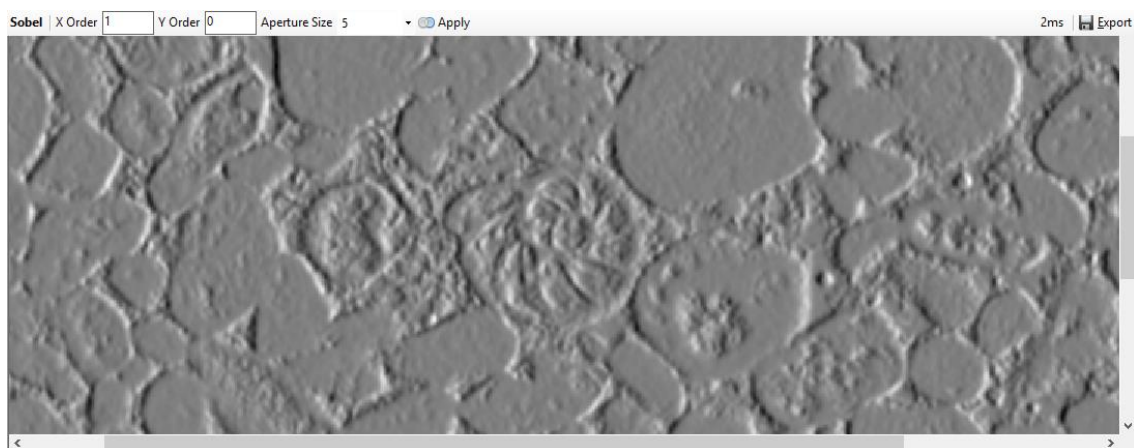
3. Apresentação do Software

A partir da imagem suavizada, é aplicado o algoritmo de Canny para detecção dos contornos, após isso é recriada uma imagem a partir da informação obtida do algoritmo. Essa imagem é depois apresentada neste painel.

É possível alterar os parâmetros do algoritmo para obter um resultado mais preciso, estando disponíveis:

1. "Threshold" que representa o valor do gradiente mínimo, caso o valor do gradiente do pixel for maior que o valor do "Threshold", este será considerado como um contorno válido. O valor recomendado por defeito é '200'.
2. "Threshold Link" faz o inverso do "Threshold", caso o valor do gradiente do pixel for menor que o valor do "Threshold Link", este será considerado como um contorno válido. O valor recomendado por defeito é '100'.
3. "Aperture Size" define a abertura do filtro mediano com uma matriz de ksize x ksize, o valor apenas pode ser 3, 5 e 7. Valores maiores geram mais contornos sendo mais sensível ao ruído. O valor recomendado é '3' por defeito.
4. "L2gradient", Valor de 'Sim' ou 'Não', usado para calcular a norma da magnitude do gradiente de uma forma mais precisa. 'Sim' pode levar a uma representação mais real dos contornos, mas também pode perder informação, dependendo da imagem em análise por isso o valor recomendado por defeito é 'Não'.

Figura 18 - Aplicação - Painel 4 - Sobel



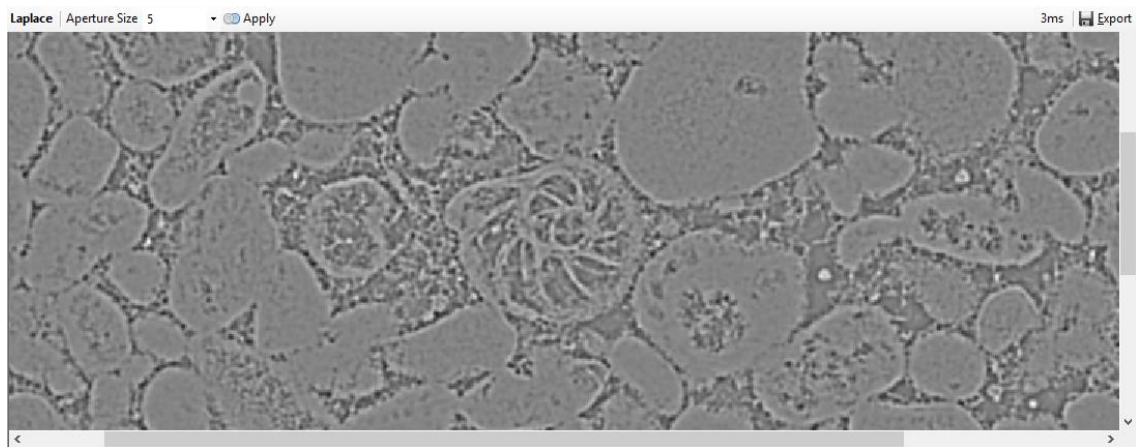
3. Apresentação do Software

A partir da imagem suavizada, é aplicado o algoritmo de Sobel, este operador combina o alisamento e diferenciação gaussiana, de modo a que o resultado seja mais ou menos resistente ao ruído.

É possível alterar os parâmetros do algoritmo para obter um resultado mais preciso, estando disponíveis:

1. "X Order" ordem da derivada X. O valor recomendado por defeito é '1'.
2. "Y Order" ordem da derivada Y. O valor recomendado por defeito é '0'.
3. "Aperture Size" define a abertura do filtro mediano com uma matriz de ksize x ksize, o valor apenas pode ser 1, 3, 5 e 7. Valores maiores geram mais contornos sendo mais sensível ao ruído. O valor recomendado é '5' por defeito.

Figura 19 - Aplicação - Painel 5 - Laplace



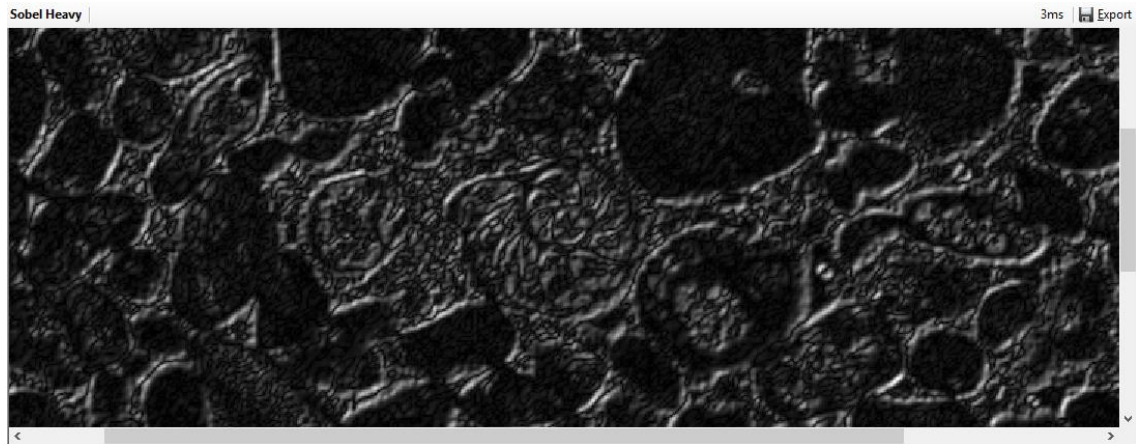
A partir da imagem suavizada, é aplicado o algoritmo de Laplace, que calcula o Laplaciano da imagem, somando os segundos derivados x e y calculados a partir do operador Sobel.

É possível alterar os parâmetros do algoritmo para obter um resultado mais preciso, estando disponíveis:

1. "Aperture Size" define a abertura do filtro mediano com uma matriz de ksize x ksize, o valor apenas pode ser 1, 3, 5 e 7. Valores maiores geram mais contornos sendo mais sensível ao ruído. O valor recomendado é '5' por defeito.

3. Apresentação do Software

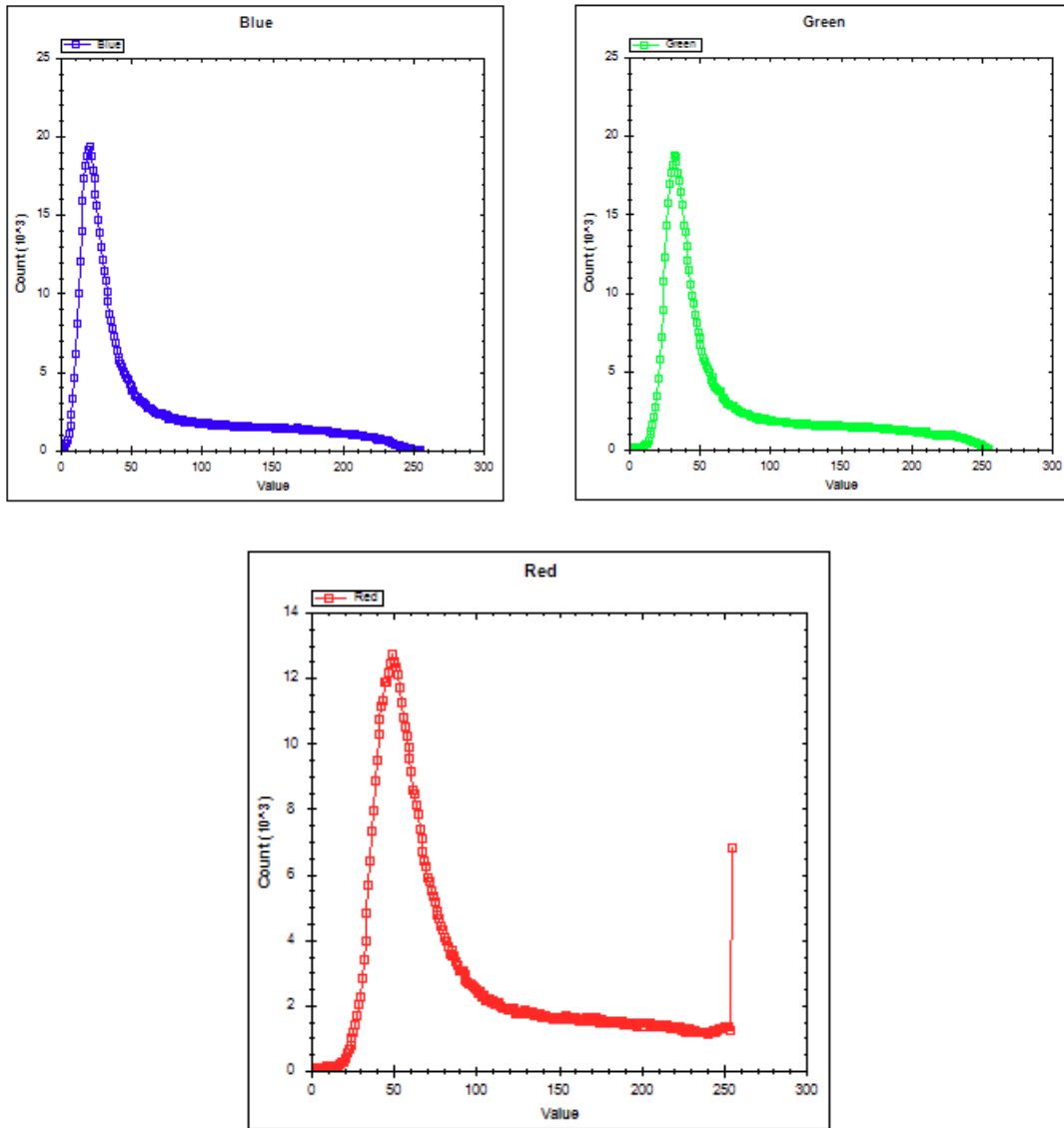
Figura 20 - Aplicação - Painel 6 - Sobel Heavy



A partir da imagem suavizada, é aplicado o algoritmo de sobel sobre sobel novamente, criando uma representação mais profunda da imagem.

3. Apresentação do Software

3.3. Histograma



Histograma de intensidade das cores sobre a imagem original, ilustrando a distribuição da intensidade. Onde o X é valor do pixel, de 0 a 255 e o Y a quantidade de pixéis que repetem esse valor. A partir do histograma podemos reconhecer qual o tom mais abundante na imagem, e com isso reconhecer zonas de cores características.

4. Avaliação

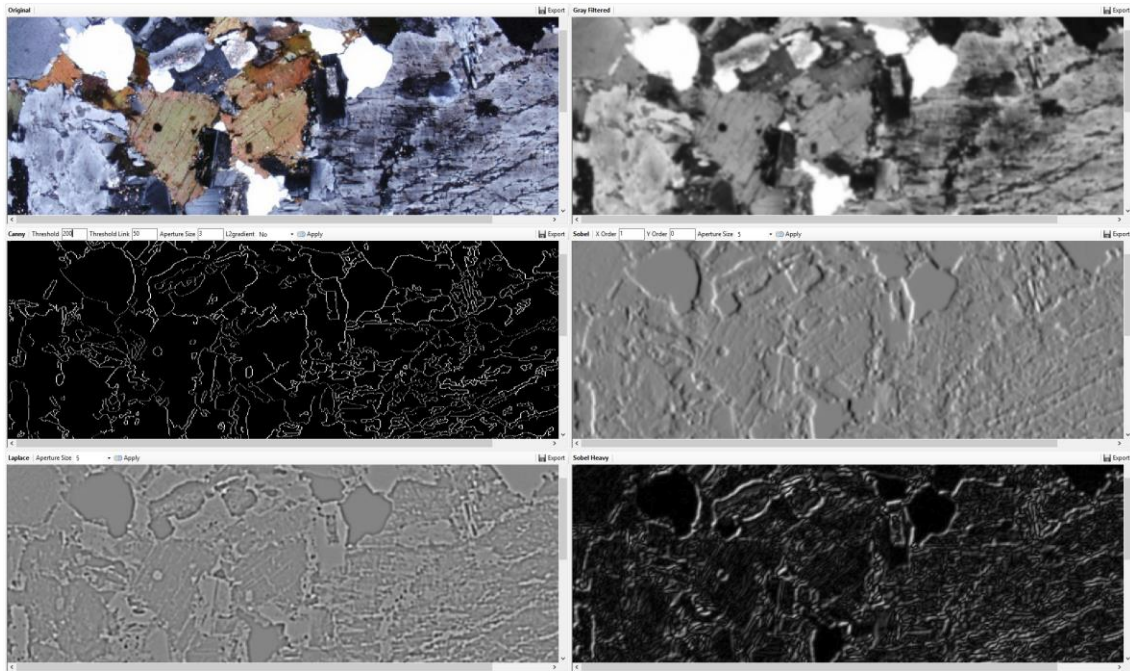
Analisar uma imagem manualmente sem suporte e traçar zonas descontínuas é uma tarefa exigente com grande margem para erro ou distrações óticas, o fator erro humano representa uma grande responsabilidade para quem as analisa, dependendo do relevo, tipo de rocha e danos da mesma esta tarefa pode ser complexa e distrativa ao ponto de criar ilusão ótica e uma saturação rápida da visão.

O output da aplicação tem como objetivo auxiliar o técnico que analisa as imagens, tornando a tarefa mais simples. A aplicação traça as linhas descontínuas e recria a imagem com essas mesmas linhas, eliminando tudo o resto. Para além das linhas é também criada outros mapas das imagens para facilitar uma decisão, dando mais informação sobre as linhas e outra perspectiva de visualização da mesma imagem, tal como zonas e profundidade que estão ausentes na representação dos contornos. Em última instância as imagens e as suas representações são expostas para o utilizador, são colocadas lado a lado numa grelha de 2 colunas por 3 linhas (Original, Tons de cinzento; Canny, Sobel; Laplace, Sobel Heavy), todas as movimentações feitas numa imagem (zoom ou mover) é sincronizado entre todas as representações para que o nível de seleção de uma esteja a par com todas as outras imagens, dessa forma é possível analisar uma determinada zona entre todas com precisão.

Dependendo do tipo de rocha, cores, ou estrutura pode haver um erro significativo gerado pela aplicação, cabe ao técnico reconhecer esse erro na visualização geral e ajustar os valores de reconhecimento (Thresholds) dos vários algoritmos. Pode ser construída uma tabela com os valores que melhor se adequa com determinado tipo de rocha de modo a facilitar o processo.

4. Avaliação

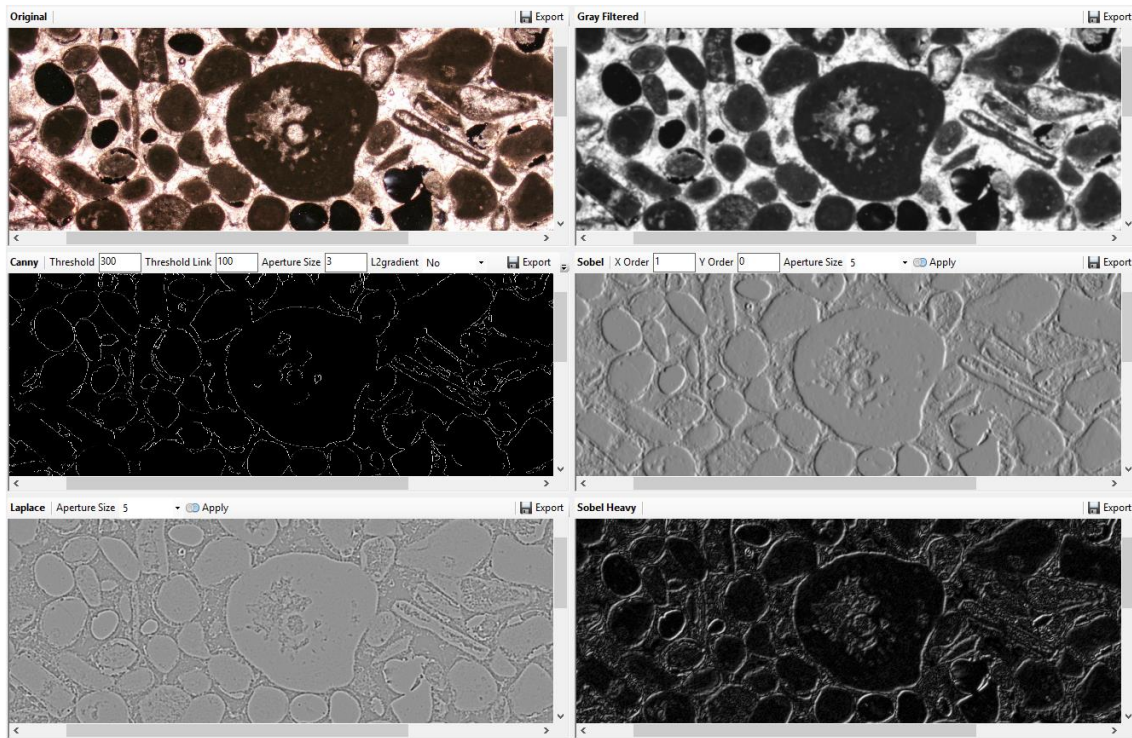
Figura 21 - Avaliação - Painel 1



Neste painel podemos observar a transformação feita a partir da imagem original. Numa análise à original ou até mesmo à imagem após a filtragem (tons de cinzento) podemos concluir que detetar fraturas ou até mesmo zonas pode ser confuso e difícil. No entanto se analisarmos o mapa gerado pelo *Canny* conseguimos facilmente detetar as zonas e áreas descontínuas. As outras representações auxiliam a decisão e a comparação entre mapas, no *Sobel* observamos uma "espécie" de relevo que ajuda na interpretação das zonas.

4. Avaliação

Figura 22 - Avaliação - Painel 2



No "Painel 2" temos uma imagem com bastante ruído de fundo, os algoritmos foram capazes de filtrar parte do ruído, no entanto no caso do *Canny* algum desse ruído foi transformado em linhas. Para corrigir essa situação o "Threshold" foi aumentado para "300" em vez dos "200", como resultado mostrado no "Painel 2", muito desse ruído foi reduzido. No entanto para estas imagens podemos observar o uso dos outros algoritmos como o *Sobel* e o *Laplace* que fizeram uma representação muito significativa de forma a ajudar a uma decisão e avaliação.

5. Conclusão e trabalho futuro

Proposta para trabalho futuro:

1. Melhoramento da aplicação de forma a detetar automaticamente zonas, minerais, tipos de rochas, fraturas e indicar possíveis soluções;
2. Implementação de um sistema de Inteligência artificial (IA) capaz de tornar o sistema autónomo e mais eficaz;
3. Implementação de um sistema via ultrassons capaz de sondar internamente um bloco e identificar possíveis falhas internas. Desta forma pretende-se saber o local exato onde se encontra danos, para que se possa atuar sobre esse mesmo local e corrigir o problema, evitando o alastramento da fissura de modo a salvaguardar toda a estrutura e composição do bloco.

6. Bibliografia

Abbass, Husein Hadi, and Zainab Radhi Mousa. "Edge detection of medical images using Markov basis." *Vol. 11, 2017, no. 37, 1825-1833*, 2017: 9.

Canny, J. "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, no. 6 (Nov 1986): 679-698.

Irwin, Sobel. "An Isotropic 3x3 Image Gradient Operator." *Presentation at Stanford A.I. Project 1968*, 02 2014: 5.

Lindeberg, Tony. "Edge Detection and Ridge Detection with Automatic Scale Selection." *International Journal of Computer Vision* 30, n° 2 (Nov 1998): 117-156.

Rocha, Manuel. *Mecânica das Rochas*. LNEC, 2013.

Sobel operator. 27 de 09 de 2019. https://en.wikipedia.org/wiki/Sobel_operator.