

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Erick Jhonson Oliveira da Silva Sá

**Método de detecção de trincas por
fadiga com base na análise de malhas
de superfície**

Projeto de Graduação

Projeto de Graduação apresentado ao Departamento de Engenharia Mecânica
da PUC-Rio

Orientador: Ivan F.M. de Menezes

Rio de Janeiro
julho de 2018

AGRADECIMENTOS

Agradeço a Deus e à minha família, sobretudo, aos meus pais, Claudio da Conceição e Ana Paula; às minhas tias, Raquel e Rejane; aos meus avós, Paulo Roberto e Maria da Glória; aos meus irmãos, Ana Carolina e Daniel; à minha namorada e grande amiga, Eliane, e aos bons professores que tive. Sem esses atores, simplesmente, todo o meu esforço seria inútil.

RESUMO

Método de detecção de trincas por fadiga com base na análise de malhas de superfície

O método foi desenvolvido utilizando-se a linguagem de programação do MATLAB. Inicialmente, implementou-se um algoritmo que recebia como entrada malhas de superfície, no formato STL. A análise foi, então, feita com base nas distâncias e ângulos de faces vizinhas. Os resultados foram satisfatórios, na medida em que permitiram a detecção de trincas em modelos tridimensionais. Infelizmente, depois de realizada uma pesquisa mais apurada, descobriu-se que a resolução dos scanners 3D atuais não era suficiente para a detecção de aberturas de trincas.

Dado o caráter essencialmente teórico do método baseado nas distâncias e ângulos de faces de modelos tridimensionais, desenvolveu-se uma segunda vertente, baseada, agora, nas cores das faces de objetos tratados com líquido penetrante. Foram utilizadas algumas imagens para a calibração dos parâmetros de busca, tendo o espaço LAB de cores sido adotado na análise, dada a sua forma consistente de identificar as cores. O algoritmo apresentou resultados satisfatórios, fornecendo ferramental adicional para tornar a detecção de trincas mais confiável.

Palavras-chave: trincas; inspeção; detecção; malhas; fotos; scanners; algoritmo; confiabilidade.

ABSTRACT

Fatigue cracks detection method based on meshes analysis

The method was developed using the programming language of MATLAB. In the beginning, was created an algorithm that received as input STL meshes. The first analysis was done based on the distances and angles between neighbor faces. The results seemed to be good, because it allowed the cracks detection on the meshes. Unfortunately, after a deeper research, it was realized that the 3d scanners resolution was not good enough, in order to detect the typical crack mouth opening displacement.

Once the first method became just a theory, it was developed a second method based on the color of the surfaces after a dye penetrant liquid application. Images were used to adjust the parameters in the algorithm. The LAB space of color was used, since it provides a more consistent way to detect colors. The results were, relatively, good and provided an additional tool to make the crack detection more reliable.

Keywords: cracks; inspection; detection; meshes; photos; scanners; algorithm; reliability.

SUMÁRIO

1	Introdução	8
1.1	Inspeção visual	8
1.2	Líquido penetrante	8
1.3	Partículas magnéticas	9
1.4	Correntes parasitas	9
1.5	Ultrassom	9
1.6	Radiografia	9
2	Objetivo	10
3	Escaneamento tridimensional	10
3.1	Malhas tridimensionais	11
4	O método de detecção	12
4.1	Algoritmo baseado em distâncias e ângulos	12
4.2	Algoritmo baseado em cores	18
4.2.1	O método inspeção por líquido penetrante	18
4.2.2	Tratamento de cores	19
4.2.3	Trabalhando-se com malhas de superfície	22
4.2.4	Testes	24
5	A interface	29
5.1	Fluxograma de funcionamento do programa	29
6	Conclusão	32
7	Perspectivas para o futuro	32
8	Referências bibliográficas	33

LISTA DE FIGURAS

Figura 1: configuração comum de malhas geradas por scanners 3D	11
Figura 2: padrão geométrico considerado para uma trinca	12
Figura 3: apótemas de um triângulo	13
Figura 4: raio de busca.....	13
Figura 5: modelo de testes	14
Figura 6: localização das trincas no modelo	15
Figura 7: modelo de testes	16
Figura 8: trincas detectadas pelo algoritmo.....	16
Figura 9: superfície tratada com líquido penetrante (Imagem: http://www.inspectro.com.br/valores/solda/ensaio-de-liquido-penetrante/)	18
Figura 10: líquido penetrante (Imagem: https://www.casadooperario.com.br) .	19
Figura 11: espaço LAB (Fonte: http://sensing.konicaminolta.com.br)	20
Figura 12: superfície com líquido penetrante	20
Figura 13: subdivisão da imagem em blocos de cores.....	21
Figura 14: histograma de cores.....	21
Figura 15: seleção dos grupos com as componentes vermelhas mais intensas	22
Figura 16: composição das malhas de teste	23
Figura 17: simulação de malhas.....	23
Figura 18: resultado do algoritmo para malhas de superfície.....	23
Figura 19: teste do algoritmo (Imagem: http://engeend.com.br/ensaios/liquido-penetrante)	24
Figura 20: teste do algoritmo (Imagem: http://ibndt.com/ver_servicos.asp?codigo=32&secao=3)	25
Figura 21: teste do algoritmo (Imagem: http://www.inspectro.com.br/valores/solda/ensaio-de-liquido-penetrante)	26
Figura 22: teste do algoritmo (Fonte: http://www.novadatatestinglabs.com/liquid-dye-penetrant-testing.html).....	26
Figura 23: interface criada.....	29

LISTA DE TABELAS

Tabela 1: inspecção visual (pontos fortes e fracos)	8
Tabela 2: líquido penetrante (pontos fortes e fracos)	8
Tabela 3: partículas magnéticas (pontos fortes e fracos)	9
Tabela 4: correntes parasitas (pontos fortes e fracos)	9
Tabela 5: ultrassom (pontos fortes e fracos)	9
Tabela 6: radiografia (pontos fortes e fracos)	9
Tabela 7: trincas detectadas pelo programa para o modelo 1	15
Tabela 8: trincas detectadas pelo programa para o modelo 2	17

1 Introdução

As trincas causadas por fadiga sempre foram objeto de preocupação no ramo da engenharia mecânica. Negligenciar tal tema é assumir o risco de falhas catastróficas em projetos, com impactos financeiros significativos.

Ao longo dos anos, foram desenvolvidos diversos métodos de inspeção, com vistas à detecção de trincas. Listam-se os principais abaixo:

- Inspeção visual
- Líquido penetrante
- Partículas magnéticas
- Correntes parasitas
- Ultrassom
- Radiografia

A descrição dos pontos fortes e fracos de cada método é feita a seguir:

1.1 Inspeção visual

Tabela 1: inspeção visual (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Barato	<ul style="list-style-type: none">▪ Requer condições de acesso ao local adequadas▪ Depende da habilidade do inspetor▪ Tamanho mínimo das trincas detectadas grande

1.2 Líquido penetrante

Tabela 2: líquido penetrante (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Barato▪ Simples▪ Confiável	<ul style="list-style-type: none">▪ Área superficial avaliada depende da quantidade de fluido existente▪ Penetração do fluido pode demorar até 120 min▪ Peça deve ser pintada▪ Depende das habilidades do inspetor

1.3 Partículas magnéticas

Tabela 3: partículas magnéticas (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Boa visibilidade de trincas	<ul style="list-style-type: none">▪ Requer utilização de limalhas de ferro sobre a superfície▪ Campo magnético pode causar interferência em instalações industriais sensíveis▪ Depende do fator humano para a determinar a concentração de limalhas▪ Preço

1.4 Correntes parasitas

Tabela 4: correntes parasitas (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Versátil▪ Automatável	<ul style="list-style-type: none">▪ Sondas precisam se adaptar às diversas geometrias existentes no mercado▪ Preço

1.5 Ultrassom

Tabela 5: ultrassom (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Detecta defeitos internos▪ Versátil	<ul style="list-style-type: none">▪ Peça precisa ser recoberta por um gel▪ Depende da habilidade do inspetor▪ Preço

1.6 Radiografia

Tabela 6: radiografia (pontos fortes e fracos)

Prós	Contras
<ul style="list-style-type: none">▪ Detecta defeitos internos	<ul style="list-style-type: none">▪ Fontes radioativas perigosas

Outros métodos merecem ser citados, mas não gozam de tanta relevância quanto os anteriores. São eles:

- Emissão acústica
- Fotoelasticidade
- Análise espectral
- Extensometria
- Termografia
- Interferometria holográfica

Apesar de a lista ser, relativamente extensa, nenhum dos métodos, entretanto, é capaz de detectar trincas menores do que 0,1 mm. O menor tamanho para o qual há alta probabilidade de detecção é da ordem de 10 mm. Esse fato pode ter impactos consideráveis na indústria, uma vez que a detecção precoce de defeitos de caráter estrutural constitui a melhor forma de evitar falhas catastróficas, o que é particularmente interessante, sobretudo, do ponto de vista financeiro.

Nesse contexto, métodos mais modernos têm sido desenvolvidos para melhorar a capacidade de inspeção em busca de defeitos em componentes mecânicos. Daremos destaque à técnica baseada no escaneamento tridimensional, usada hoje para detecção de defeitos como corrosões, indentações e demais danos mecânicos.

2 Objetivo

Este projeto consiste da implementação computacional de um algoritmo capaz de reconhecer, de forma simples e eficiente, a localização de trincas por fadiga em superfícies a partir de malhas de elementos finitos geradas por ferramentas de escaneamento tridimensional disponíveis no mercado.

3 Escaneamento tridimensional

Notoriamente, a plausibilidade do projeto é limitada pela qualidade das malhas geradas por ferramentas externas, sendo, portanto, necessário o conhecimento prévio dos métodos utilizados para a geração de tais malhas. Sendo assim,

foram comparadas especificações de alguns modelos de scanners tridimensionais de fabricantes específicos e reunidos os principais aspectos a serem considerados para a elaboração do algoritmo.

3.1 Malhas tridimensionais

As malhas geradas pelos scanners, em geral, podem ser definidas pela combinação de dois vetores: um de vértices e outro de índices. Essa combinação dá origem às faces que comporão as superfícies escaneadas.

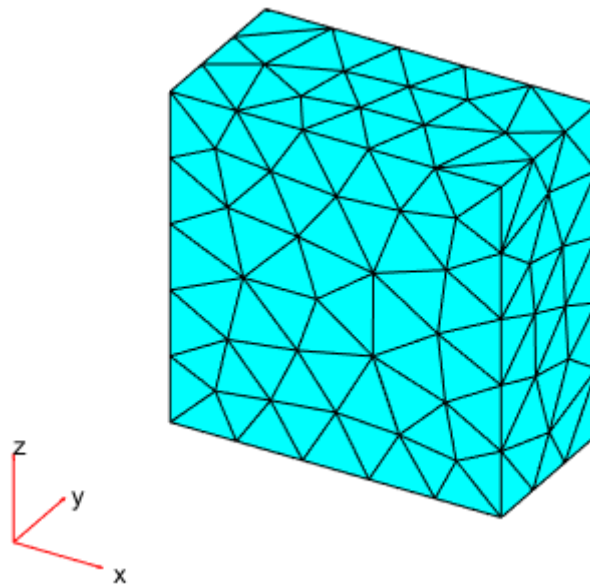


Figura 1: configuração comum de malhas geradas por scanners 3D

Na Figura 1, é ilustrado um exemplo de configuração de malha gerada por scanners 3D. Ressalta-se que, em grande parte dos casos, as malhas geradas são de elementos com três vértices.

Procurado, um dos fabricantes afirmou que seus scanners eram capazes de gerar malhas com elementos que poderiam ir de 0,1 mm a 0,050 mm, para corpos com dimensões entre 0,1 m a 4 m. Esses valores nos permitem saber a menor abertura de trinca passível de ser detectada pelo equipamento e, portanto, exibida pelo algoritmo.

4 O método de detecção

O método desenvolvido, na forma de um aplicativo utilizando-se a linguagem MATLAB, foi projetado para sinalizar regiões onde, potencialmente, há a presença de trincas por fadiga. Após carregada uma malha, o algoritmo, basicamente, percorre uma região específica (ou toda a malha) buscando por padrões geométricos considerados críticos. Foram desenvolvidos dois algoritmos para a análise, descritos a seguir.

4.1 Algoritmo baseado em distâncias e ângulos

O primeiro algoritmo, relativamente simples, foi baseado, sobretudo, na distância e ângulo entre faces.

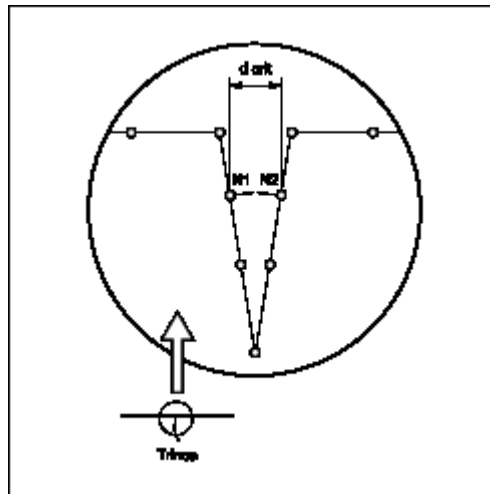


Figura 2: padrão geométrico considerado para uma trinca

Na Figura 2, as faces da malha foram representadas por seus centroides. Dessa maneira, para a classificação de uma face como pertencente ou não a uma trinca, é necessário que a distância entre essas seja suficientemente pequena e que o ângulo entre suas normais seja próximo de 180° , condição necessária para que sejam opostas.

Uma vez que a distância d_{crit} - uma medida da abertura da trinca - não é previamente definida no código, esta foi assumida como uma fração do **menor apótema** (Figura 3) de triângulo de toda a malha. Com essa condição, garante-se que não serão selecionadas faces adjacentes. Naturalmente, o usuário poderá, a seu critério, especificar a distância crítica d_{crit} .

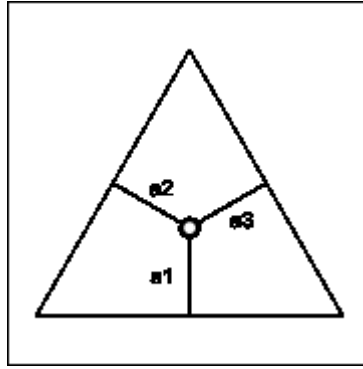


Figura 3: apótemas de um triângulo

Embora a Figura 3 ilustre um triângulo equilátero, vale mencionar que, em uma malha tridimensional, nem todos o são.

Essa análise pode ser feita em toda a malha. Nesse caso, para n faces, seriam necessárias $\frac{n(n-1)}{2}$ operações, o que poderia deixar o código extremamente lento, dependendo do grau de refino da malha. Dessa forma, para tornar o processamento mais rápido, foi definido um raio de busca, a ser usado em casos específicos, para diminuir o número de operações e, por conseguinte, do tempo de processamento, uma vez que as análises das distâncias e ângulos só serão efetuadas entre triângulos que estiverem dentro do referido raio (vide Figura 4).

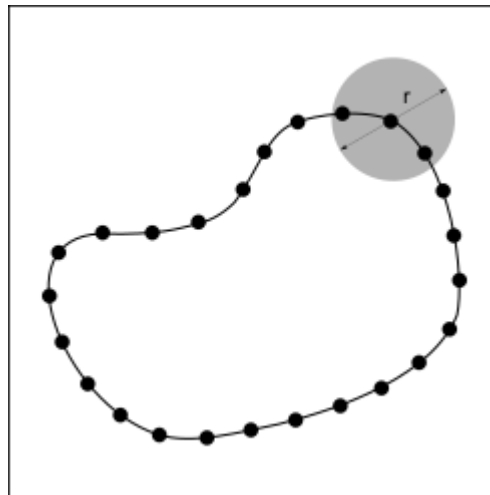


Figura 4: raio de busca

Dada a dificuldade para obtenção de modelos reais - escaneados em campo -, foi utilizado o software SolidWorks para a modelagem de peças para testes. Foram geradas algumas superfícies, com diferentes valores de d_{crit} , as quais foram analisadas com o programa desenvolvido.

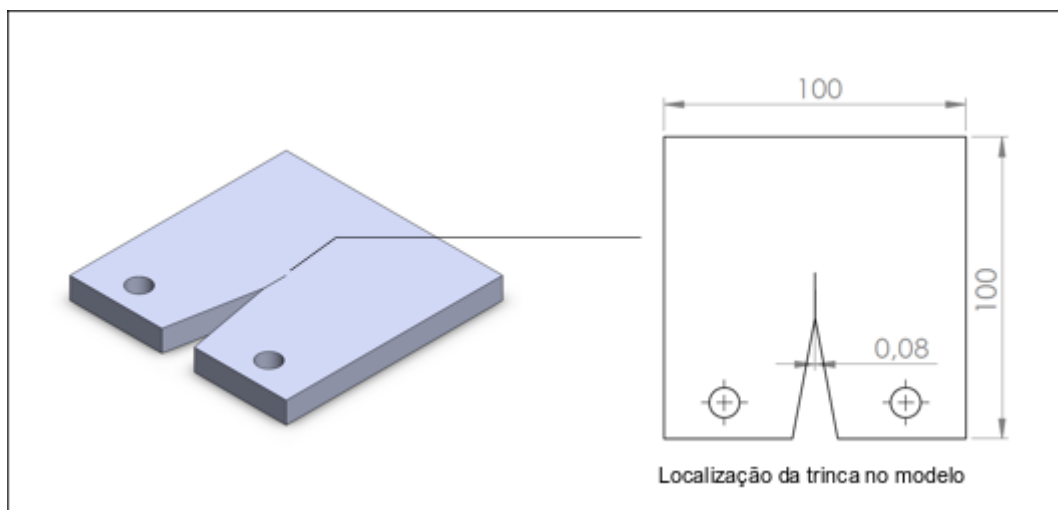


Figura 5: modelo de testes

Para o modelo da Figura 5, o algoritmo de análise nos levou aos seguintes resultados:

Sem otimização

- Total de faces do modelo: 344
- Combinações de faces testadas: 58996
- Faces em trincas: 2
- Tempo de execução: 152.485 s

Com otimização

- Total de faces do modelo: 344
- Combinações de faces testadas: 4242
- Faces em trincas: 2
- Tempo de execução: 20.911 s

No primeiro caso, o modelo foi analisado utilizando-se o algoritmo sem otimização, o que resultou em $\frac{n(n-1)}{2}$ operações, com tempo de processamento de 152,485 s. Dessa forma, foram testadas todas as combinações de faces possíveis.

No segundo caso, o algoritmo foi otimizado (vide Figura 4) para que fossem testadas somente faces próximas entre si, o que reduziu o tempo de processamento para 20,399 s (redução de 86,62%). Ressalta-se que, nos casos em que a otimização não produzir resultados, o algoritmo, automaticamente, repetirá a análise sem considerar a otimização proposta.

Para ambos os casos (com e sem otimização), a quantidade de faces em trincas detectadas foi a mesma, como mostra a tabela a seguir, gerada em *.xlsx, automaticamente, pelo programa.

Tabela 7: trincas detectadas pelo programa para o modelo 1

Face	cx	cy	cz	d	ang
Face 341	49,97	3,33	55,10	0,08	179,69
Face 344	50,03	6,67	55,10	0,08	179,69

Os resultados do primeiro teste mostram que apenas duas faces foram caracterizadas como pertencentes a trincas. A distância e o ângulo entre elas são representados pelas variáveis d e ang, respectivamente. As variáveis cx, cy e cz representam as coordenadas dos centroides destas.

A Figura 6 mostra, em verde, as faces selecionadas com o algoritmo, exatamente como esperado.

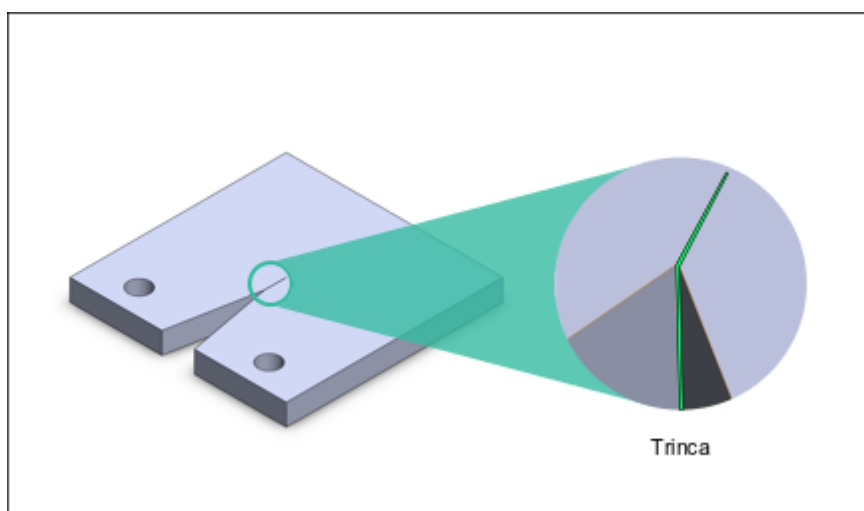


Figura 6: localização das trincas no modelo

As faces sempre surgirão aos pares, dada a forma como o algoritmo foi feito. Para o primeiro teste, por exemplo, os valores foram referentes a uma trinca formada pela combinação entre as faces 341 e 344.

O programa, também, gera um relatório *.txt com informações mais detalhadas da análise:

```
Faces em trincas

Faces 341 x 344: d = 0.080; theta = 179.694

Total de faces do modelo: 344
Combinações de faces testadas: 4242
Faces em trincas: 2
Tempo de execução: 20.911 s
```

Foi elaborado um segundo teste, com um modelo diferente.

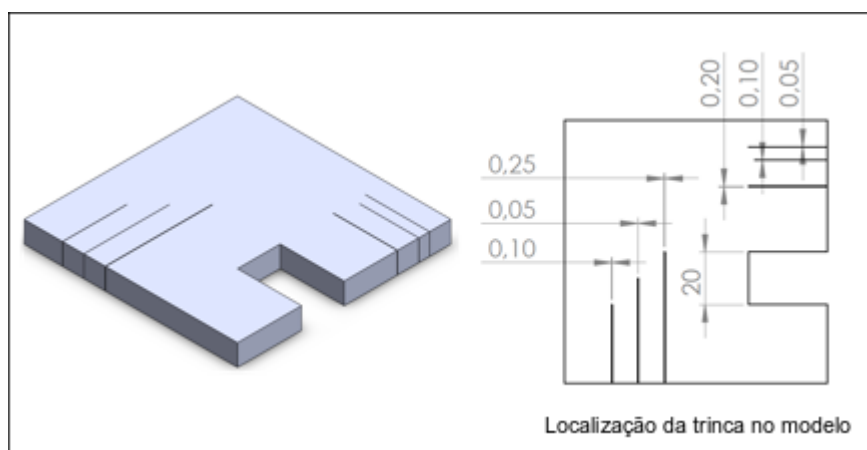


Figura 7: modelo de testes

A Figura 8 mostra as trincas detectadas pelo algoritmo.

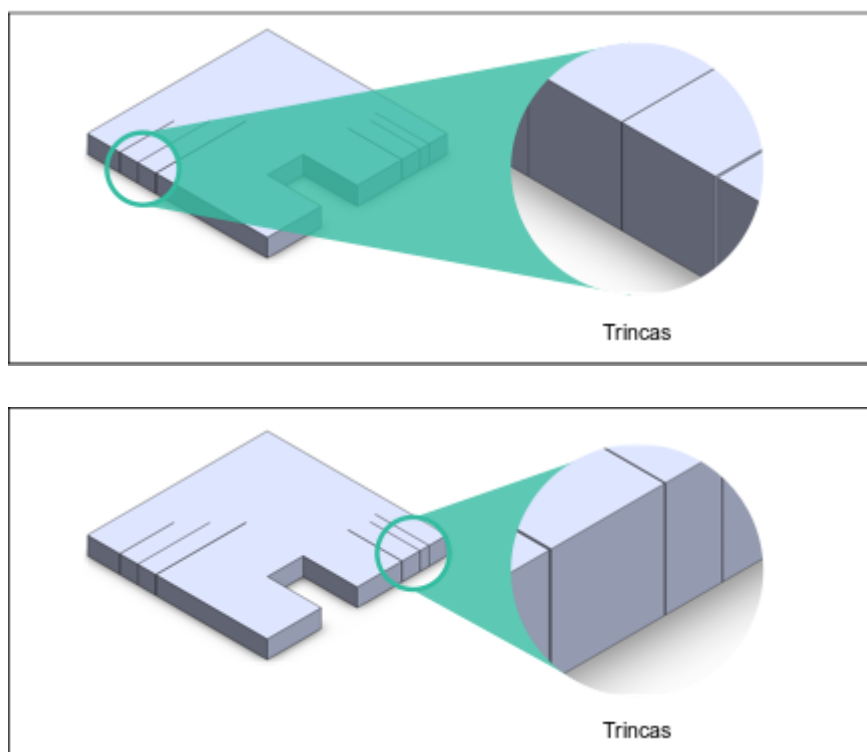


Figura 8: trincas detectadas pelo algoritmo

A tabela (reduzida) de resultados obtida foi:

Tabela 8: trincas detectadas pelo programa para o modelo 2

faces	cx	cy	cz	d	ang
Face 83	18,00	3,33	90,00	0,10	180,00
Face 87	18,10	3,33	80,00	0,10	180,00
Face 83	18,00	3,33	90,00	0,10	180,00
Face 88	18,10	6,67	90,00	0,10	180,00
Face 84	18,00	6,67	80,00	0,10	180,00
Face 87	18,10	3,33	80,00	0,10	180,00
Face 84	18,00	6,67	80,00	0,10	180,00
Face 88	18,10	6,67	90,00	0,10	180,00
Face 89	28,00	3,33	86,67	0,05	180,00
Face 93	28,05	3,33	73,33	0,05	180,00
Face 89	28,00	3,33	86,67	0,05	180,00
Face 94	28,05	6,67	86,67	0,05	180,00
...

A seguir, temos o relatório da análise gerado pelo programa:

Faces em trincas

```
Faces 83 x 87: d = 0.100; theta = 180.000
Faces 83 x 88: d = 0.100; theta = 180.000
Faces 84 x 87: d = 0.100; theta = 180.000
Faces 84 x 88: d = 0.100; theta = 180.000
Faces 89 x 93: d = 0.050; theta = 180.000
Faces 89 x 94: d = 0.050; theta = 180.000
Faces 90 x 93: d = 0.050; theta = 180.000
Faces 90 x 94: d = 0.050; theta = 180.000
Faces 95 x 99: d = 0.250; theta = 180.000
Faces 95 x 100: d = 0.250; theta = 180.000
Faces 96 x 99: d = 0.250; theta = 180.000
Faces 96 x 100: d = 0.250; theta = 180.000
Faces 101 x 105: d = 0.050; theta = 180.000
Faces 101 x 106: d = 0.050; theta = 180.000
Faces 102 x 105: d = 0.050; theta = 180.000
Faces 102 x 106: d = 0.050; theta = 180.000
Faces 107 x 111: d = 0.100; theta = 180.000
Faces 107 x 112: d = 0.100; theta = 180.000
Faces 108 x 111: d = 0.100; theta = 180.000
Faces 108 x 112: d = 0.100; theta = 180.000
Faces 113 x 117: d = 0.200; theta = 180.000
Faces 113 x 118: d = 0.200; theta = 180.000
Faces 114 x 117: d = 0.200; theta = 180.000
Faces 114 x 118: d = 0.200; theta = 180.000
```

Total de faces do modelo: 124

Combinações de faces testadas: 7626

Faces em trincas: 48

Tempo de execução: 21.792 s

Percebe-se, da observação dos resultados, que o algoritmo funcionou perfeitamente, nos dois testes, mesmo com a presença de padrões geométricos no modelo que poderiam causar erros, como o corte de 20 mm, na Figura 7 e; a abertura e os furos passantes, na Figura 5. Verifica-se, ainda, que, no segundo teste, a otimização não foi acionada, dada a quantidade de faces testadas (operações = $\frac{n(n-1)}{2}$).

Devido às dimensões típicas de abertura de trincas e, considerando-se as capacidades dos atuais modelos de scanners tridimensionais, esse método assume papel essencialmente teórico, podendo ser utilizado na prática à medida que os métodos de escaneamento forem sendo desenvolvidos.

4.2 Algoritmo baseado em cores

Além do algoritmo baseado nas distâncias e ângulos, para que o trabalho tivesse aplicações reais na indústria, desenvolveu-se um algoritmo para detecção de trincas com base na busca por padrões de cores.

Tal método recebe como entrada malhas obtidas do escaneamento de superfícies tratadas com líquido penetrante, conforme ilustrado na Figura 9. Uma vez que os equipamentos de escaneamento são capazes de detectar cores, podemos utilizar esse fato para encontrar trincas.



Figura 9: superfície tratada com líquido penetrante (Imagem: <http://www.inspectro.com.br/valores/solda/ensaio-de-liquido-penetrante/>)

4.2.1 O método inspeção por líquido penetrante

O método de detecção de trincas por líquido penetrante, regulamentado pela norma NBR 16450 da Associação Brasileira de Normas Técnicas, baseia-se na

aplicação de um líquido (Figura 10) sobre a superfície analisada. O líquido preenche as microfissuras e, depois da aplicação de um revelador, permite a visualização das trincas, sinalizadas por uma cor de tom semelhante ao vermelho/rosa.



Figura 10: líquido penetrante (Imagem: <https://www.casadooperario.com.br>)

Apesar de, relativamente barato, o método é completamente dependente do fator humano e, sob condições adversas de aplicação, poderia acarretar erros de detecção.

O algoritmo auxilia o operário na etapa de inspeção, ficando este responsável apenas pela aplicação do líquido sobre a superfície.

4.2.2 Tratamento de cores

Para o tratamento de cores, foi utilizado o espaço de cor LAB ou CIELAB, devido à sua forma consistente de identificar as cores. Inicialmente, foram utilizadas fotos para a calibração dos parâmetros empregados no algoritmo para, posteriormente, aplica-lo às malhas de superfície.

Nesse espaço, a cor é definida por três parâmetros, sendo um de luminosidade (L), um referente à componente vermelho/verde (A) e por outro referente à componente amarelo/azul (B). A Figura 10 ilustra bem a constituição de tal espaço.

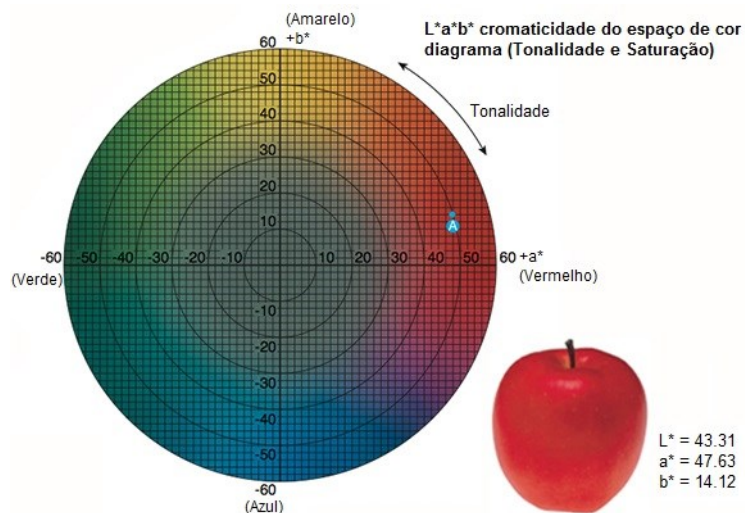


Figura 11: espaço LAB (Fonte: <http://sensing.konicaminolta.com.br>)

Da observação da Figura 11, podemos concluir:

$A > 0 \rightarrow \text{vermelho}$
 $A < 0 \rightarrow \text{verde}$
 $B > 0 \rightarrow \text{amarelo}$
 $B < 0 \rightarrow \text{azul}$

Dada a tonalidade característica do líquido penetrante, consideramos apenas cores com componente A positiva.

A Figura 12 mostra um exemplo de imagem tratada com líquido penetrante.

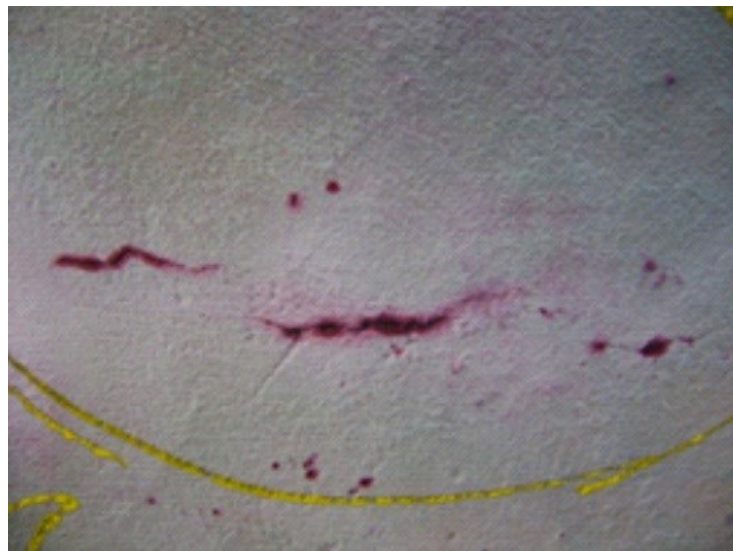


Figura 12: superfície com líquido penetrante

(Imagem: http://www.intequi.com.br/liquido_penetrante.php)

As manchas de tom vermelho/rosa indicam a presença de trincas na superfície.

O primeiro passo do algoritmo foi simplificar a imagem, agrupando-se os pixels parecidos em vários grupos, todos eles com as cores médias dos pixels que os formam.

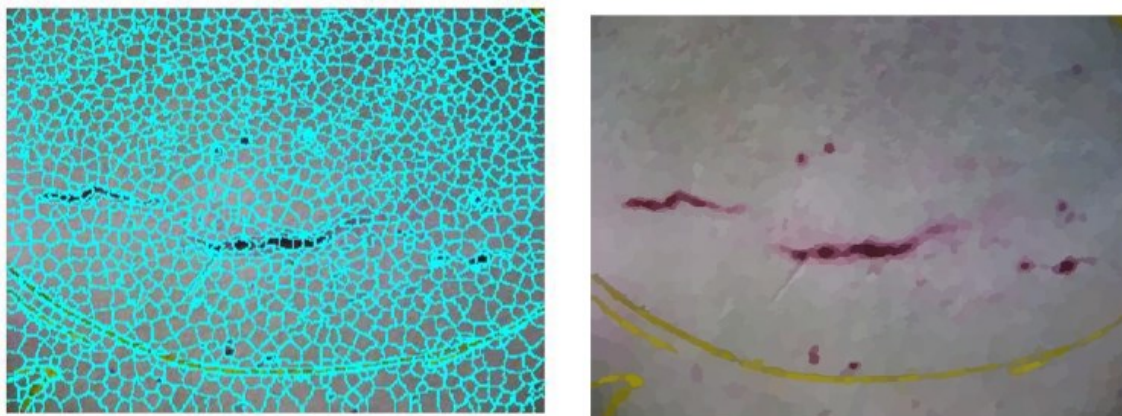


Figura 13: subdivisão da imagem em blocos de cores

A foto em questão tinha resolução de 396 x 527 pixels, tendo, portanto, 208.692 pixels a serem testados. A simplificação reduziu esse total a 1282 grupos de cores, como evidencia a Figura 13.

Para uma análise apurada, elaborou-se um histograma de cores para cada imagem (Figura 14).

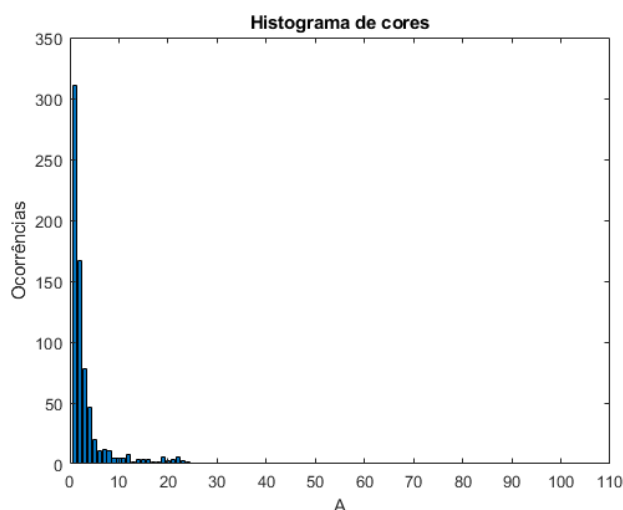


Figura 14: histograma de cores

No eixo horizontal, temos a componente vermelha indo da mais branda (0) até a mais intensa (110). No eixo vertical, podemos ver a quantidade de grupos associados a cada componente. Destacamos, então, os grupos cujas cores médias representavam as **quinze componentes vermelhas mais intensas**. Essa escolha foi fruto da observação de diversos resultados para várias rodadas de teste.



Figura 15: seleção dos grupos com as componentes vermelhas mais intensas

A Figura 15 representa o estágio final do algoritmo. Nela, as regiões críticas foram devidamente sinalizadas. A utilização de diferentes perspectivas, tornou mais confiável o código, na medida em que não se limitou a busca a apenas uma componente de cor.

4.2.3 Trabalhando-se com malhas de superfície

Uma vez calibrados os parâmetros de análise de cores, pôde-se testar o algoritmo para malhas de superfície. Tais malhas foram obtidas a partir das fotos usadas na seção anterior. Isso foi feito subdividindo-se cada pixel em dois triângulos, como mostra a Figura 16.

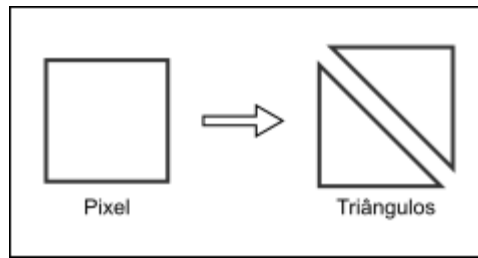


Figura 16: composição das malhas de teste

Tal aproximação foi necessária dada a inacessibilidade das malhas reais obtidas com scanners tridimensionais. O resultado desse processo é ilustrado na Figura 17.

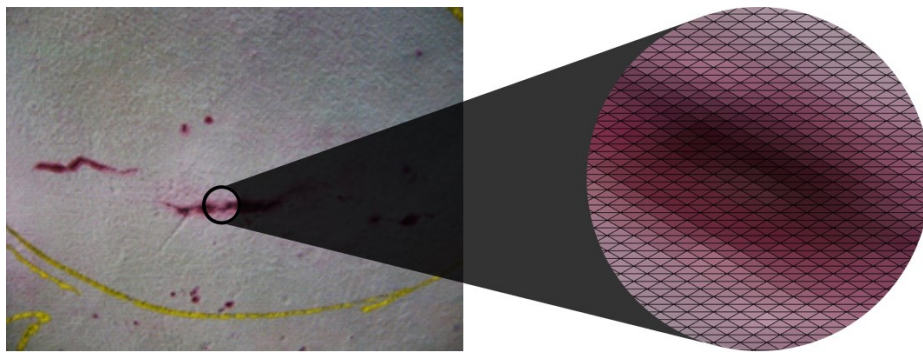


Figura 17: simulação de malhas

Com as malhas prontas, aplicamos os procedimentos de análise já descritos. Devido à complexidade da análise de objetos tridimensionais, não foi feita a subdivisão da malha em grupos de cores semelhantes. Os triângulos foram testados individualmente. Na Figura 18, podemos ver o resultado obtido com o algoritmo adaptado para malhas de superfície.

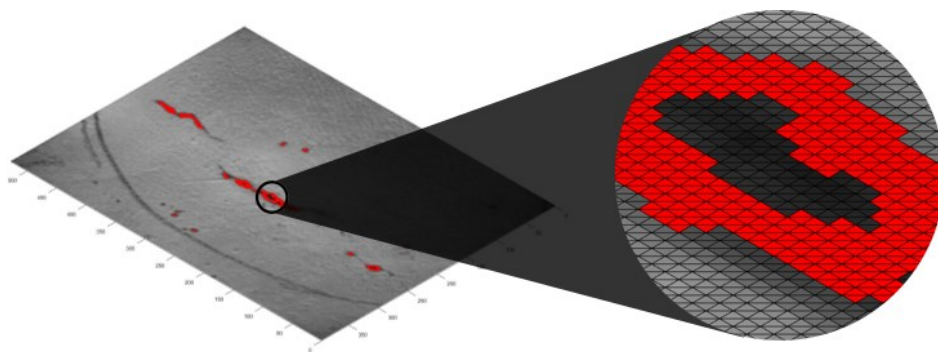


Figura 18: resultado do algoritmo para malhas de superfície

4.2.4 Testes

Para que se tivesse uma melhor noção da eficiência do algoritmo, foram feitos diversos testes, inicialmente, recebendo fotos como entrada. Em seguida, para a análise dos resultados do algoritmo de tratamento de malhas, converteram-se essas imagens em malhas, as quais foram processadas.

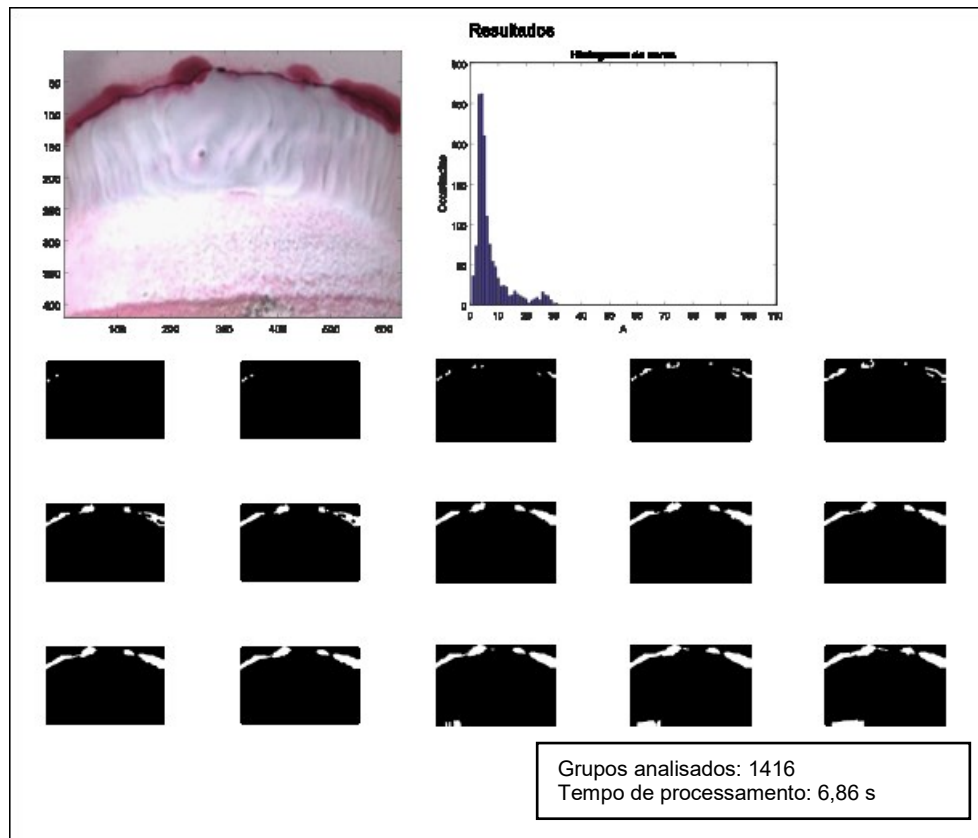


Figura 19: teste do algoritmo (Imagem: <http://engeend.com.br/ensaios/liquido-penetrante>)

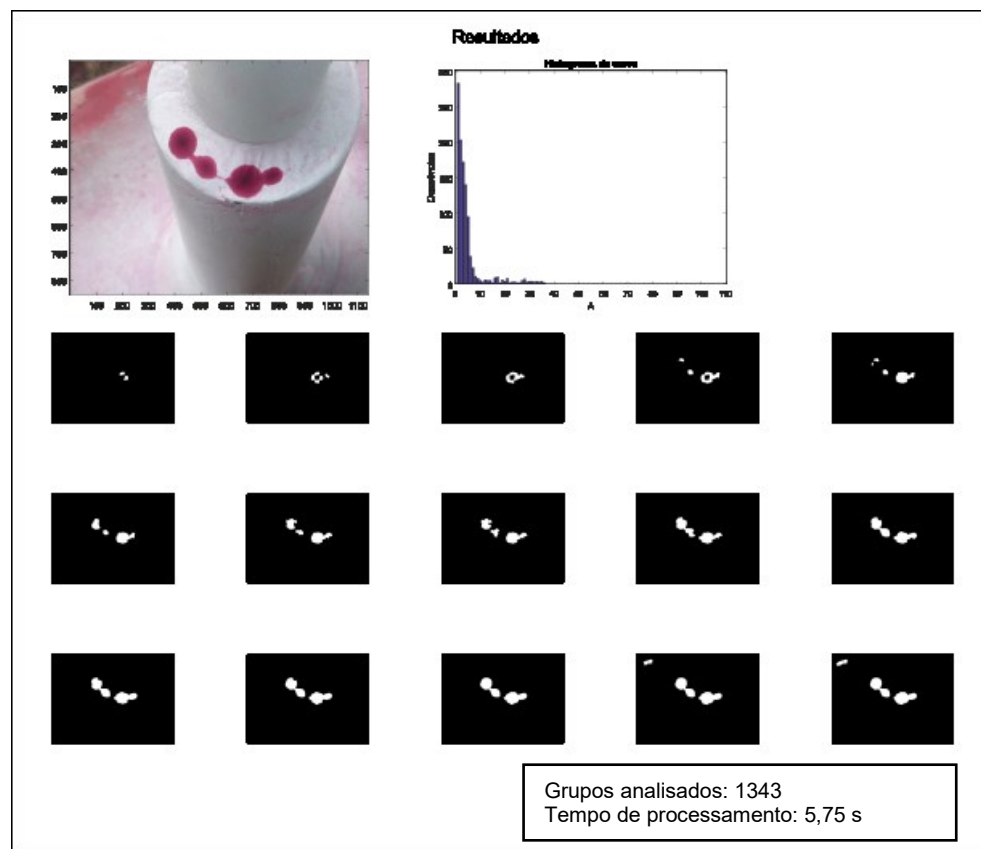


Figura 20: teste do algoritmo (Imagem: http://ibndt.com/ver_servicos.asp?codigo=32&secao=3)

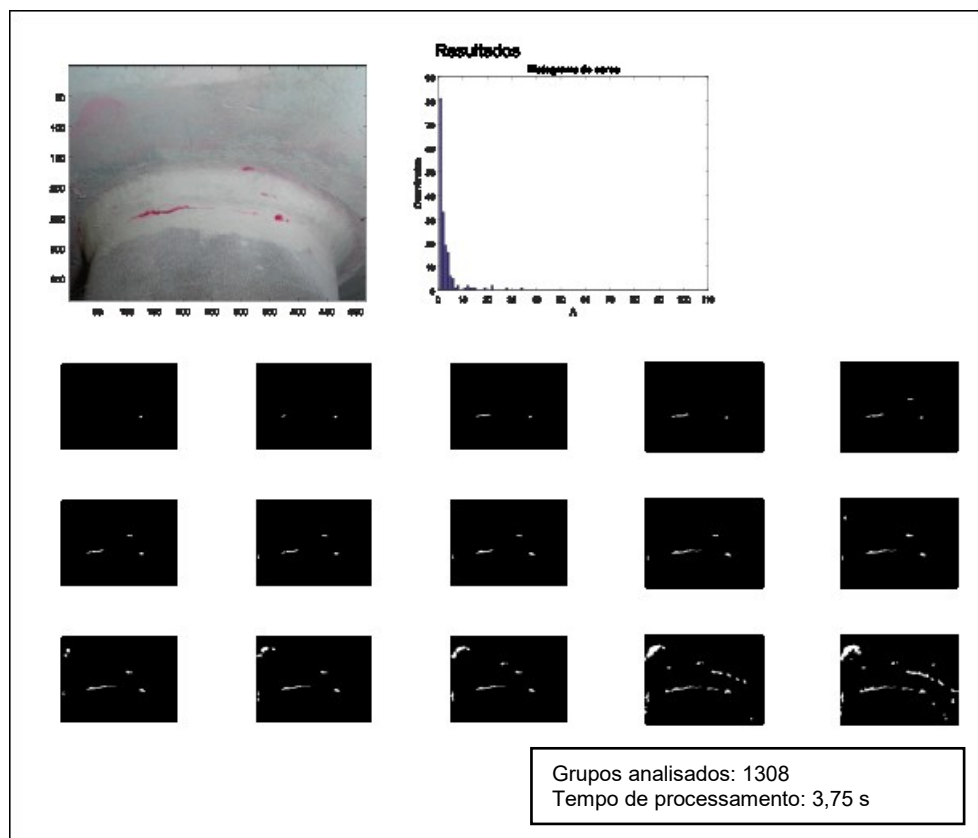


Figura 21: teste do algoritmo (Imagem: <http://www.inspectro.com.br/valores/solda/ensaio-de-liquido-penetrante>)

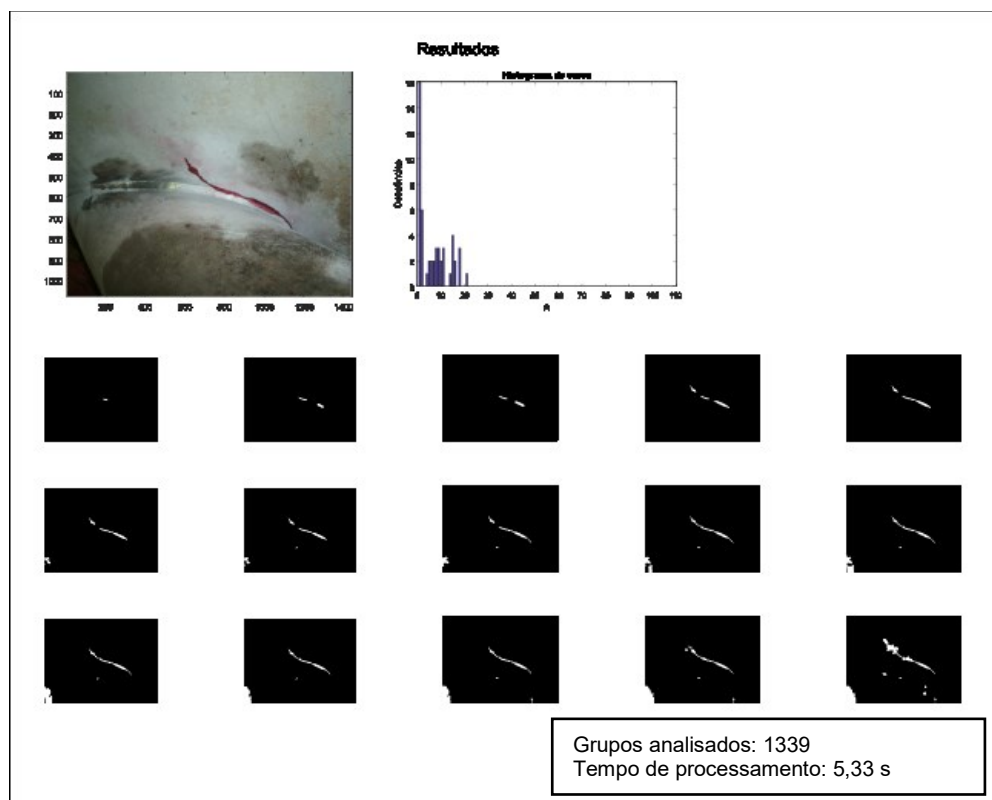
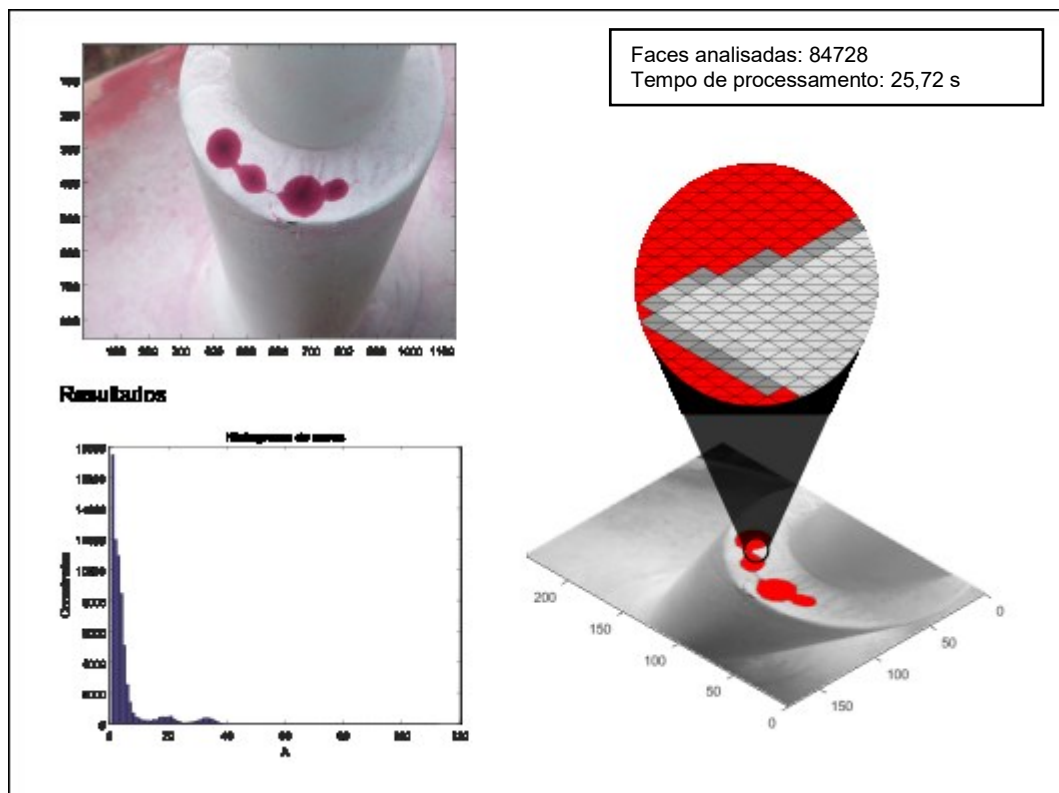
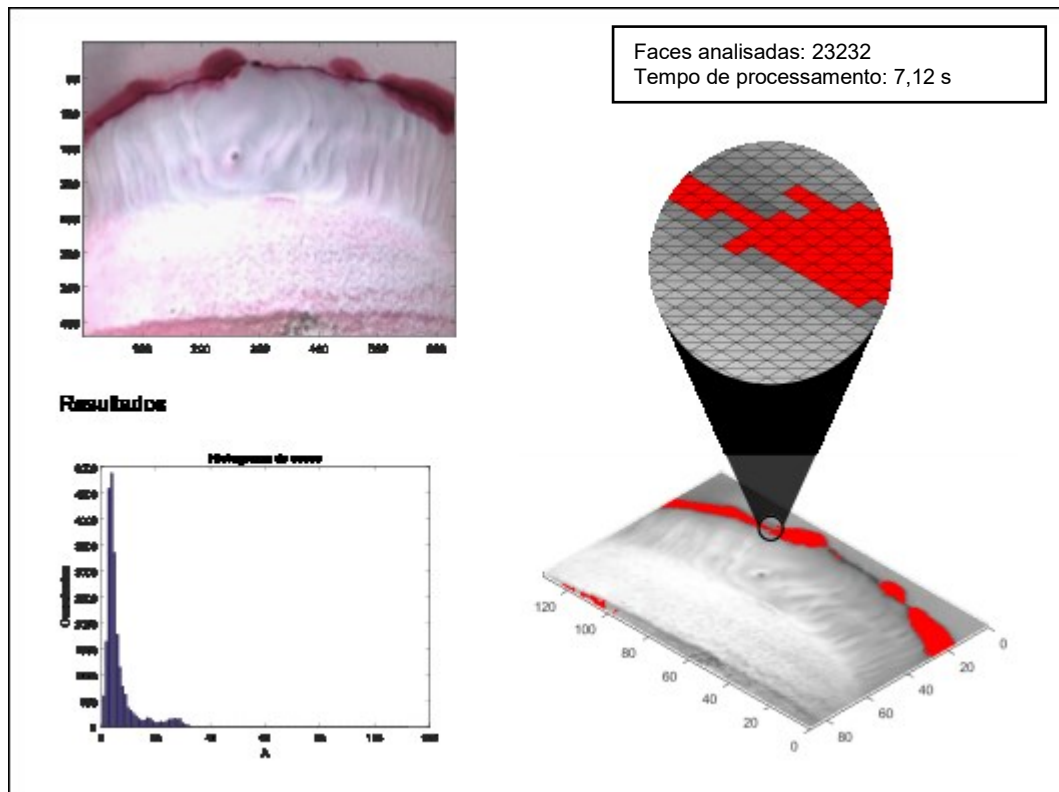
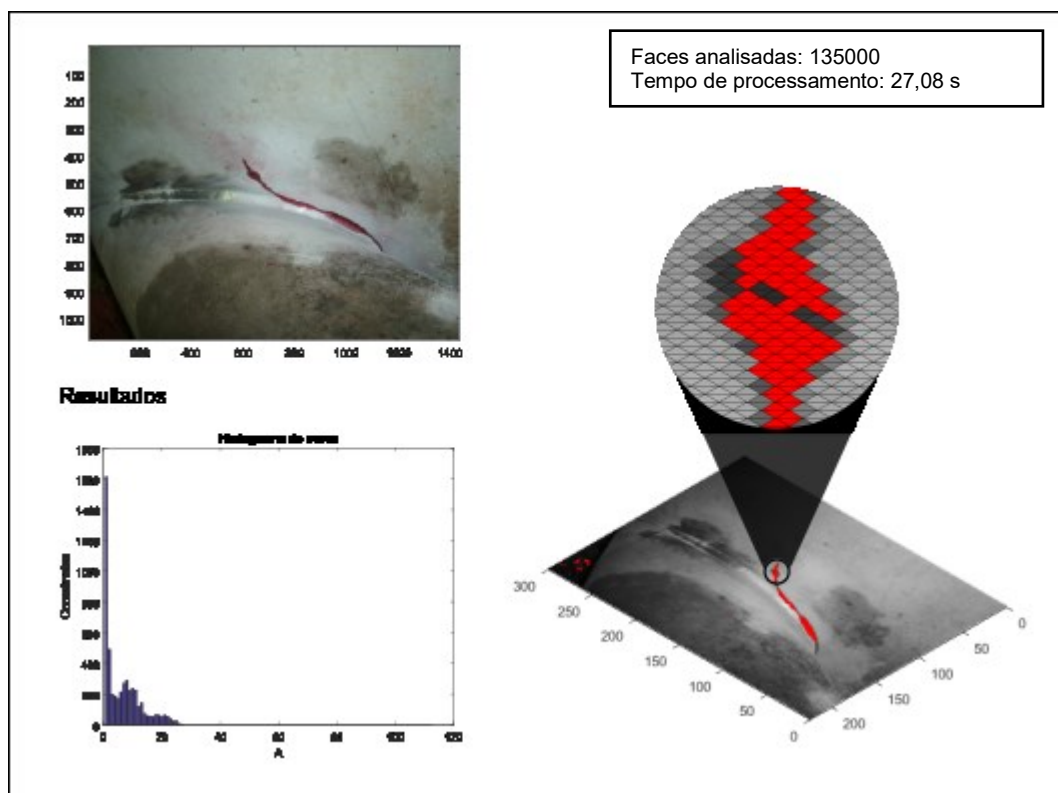
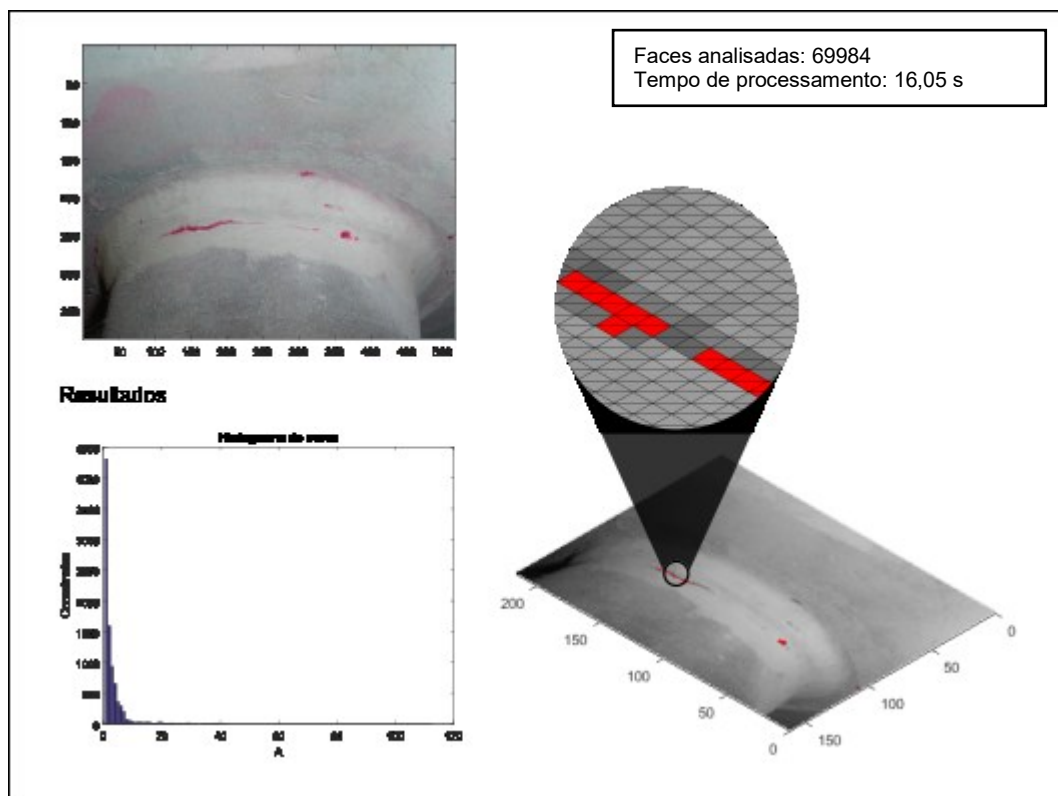


Figura 22: teste do algoritmo (Fonte: <http://www.novadatatestinglabs.com/liquid-dye-penetrant-testing.html>)

A seguir, são apresentados os resultados dos testes com o algoritmo de processamento de malhas obtidas a partir de fotos.





5 A interface

Por questões de conveniência, o programa para análise de superfícies foi feito utilizando-se a linguagem MATLAB, através da ferramenta *App Designer*, disponível na plataforma. O aplicativo, embora simples, permite um vislumbre das pretensões do projeto, podendo ser manuseado facilmente, dada a sua simplicidade.

O programa é composto por uma única janela (Figura 23).



Figura 23: interface criada

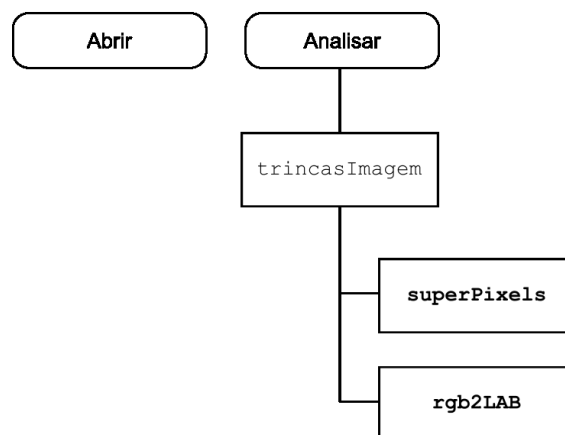
O usuário poderá importar para análise tanto fotos quanto malhas. Para fotos, a única forma de análise é aquela baseada em cores, já para malhas, há duas opções, sendo uma por distância e ângulos e outra por cores. Há ainda a opção de simular malhas coloridas a partir de imagens, como descrito anteriormente.

5.1 Fluxograma de funcionamento do programa

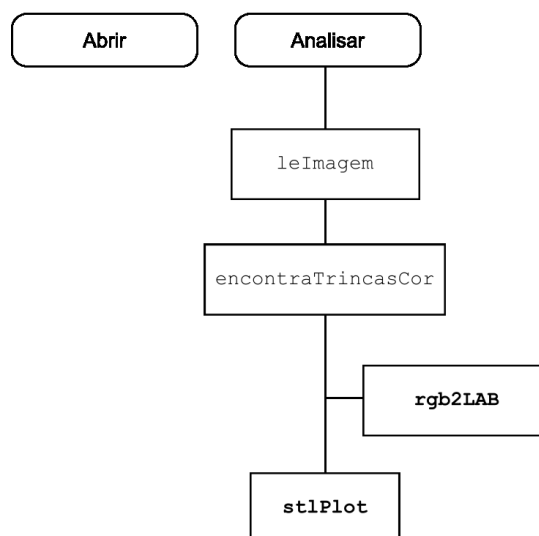
Nesta seção, é apresentado o fluxograma representativo do funcionamento do algoritmo. Para facilitar a compreensão, os esquemas foram feitos, separadamente, para as três partes do programa: FOTO, MALHA e SIMULAR MALHA. Em cada uma delas, foram selecionadas as funções mais relevantes desenvolvidas.

- **FOTO:** recebe, como entrada, imagens coloridas. O resultado é exposto na forma de imagens com as trincas destacadas, em diferentes níveis de intensidade da componente A.
- **MALHA:** recebe, como entrada, malhas no formato STL e realiza a análise com base nas distâncias e ângulos entre as faces do modelo. Para a análise com base em cores, a entrada se dá na forma de três documentos txt, contendo, cada um, os índices das faces, as coordenadas dos vértices e as cores de cada face.
- **SIMULAR MALHA:** recebe, como entrada, imagens coloridas, converte-as em malhas e realiza a análise com base em cores.

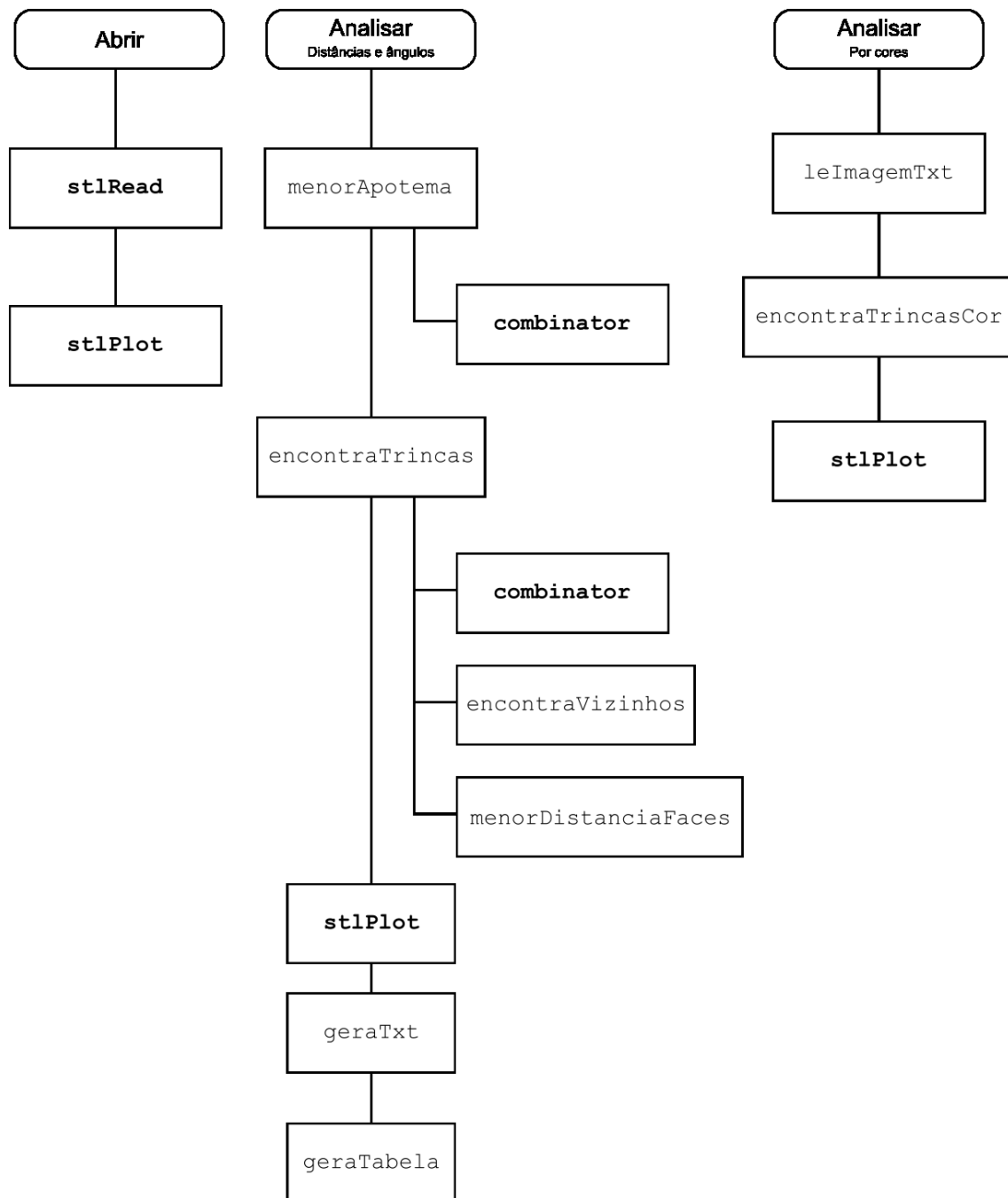
FOTO



SIMULAR MALHA



MALHA



Nos fluxogramas, as funções em **negrito** foram obtidas de terceiros, a saber: `stlRead`¹, `stlPlot`¹, `rgb2LAB`² e `combinator`³. Para fins de análise, também foi utilizada a função `progressbar`⁴.

¹ Copyright (c) 2017, Pau Micó; Copyright (c) 2015, Sven Holcombe; Copyright (c) 2011, Eric Johnson; Copyright (c) 2013, Adam H. Aitkenhead; Copyright (c) 2011, Francis Esmonde-White

² Copyright (c) 2009, Mark A. Ruzon

³ Copyright (c) 2009, Matt Fig

⁴ Copyright (c) 2005, Steve Hoelzer

6 Conclusão

Podemos dizer que os objetivos do trabalho foram concluídos. Os resultados para os dois algoritmos foram satisfatórios, na medida em que permitiram a detecção de trincas a partir de malhas de superfície.

Apesar do primeiro algoritmo, baseado nas distâncias e ângulos de faces de modelos tridimensionais, ser ainda essencialmente teórico, em função de limitações técnicas dos equipamentos disponíveis para o escaneamento 3D, o segundo, baseado nas cores de faces de modelos tridimensionais, possui aplicação real na indústria, na medida em que fornece resultados cada vez mais confiáveis pelo método do líquido penetrante, especialmente, quando tal método for aplicado em condições adversas, relacionadas com: iluminação, ergonomia, acessibilidade, temperatura, umidade, cansaço físico e fatores psicológicos desfavoráveis.

7 Perspectivas para o futuro

O algoritmo foi, completamente, desenvolvido utilizando-se a linguagem de programação do MATLAB. A migração para uma linguagem de programação com menos limitações, certamente, permitiria resultados mais rápidos e com mais liberdade para o desenvolvimento de ferramentas úteis.

Em se tratando de malhas de elementos finitos, seria válida a aplicação de análise de tensões para otimização do código, restringindo o algoritmo às regiões críticas somente.

Adicionalmente, poder-se-ia estabelecer uma parceria com algum fabricante de scanner 3D para que se pudessem fazer testes com modelos reais, o que tornaria os resultados da análise ainda mais confiáveis.

8 Referências bibliográficas

Entendendo o espaço de cor LAB. (11 de Junho de 2018). Fonte:

<http://sensing.konicaminolta.com.br/>

<http://sensing.konicaminolta.com.br/2013/11/entendendo-o-espaco-de-cor-lab/>

Pinho de Castro, J. T., & Meggiolaro, M. A. (2009). *Fadiga - Técnicas e Práticas de Dimensionamento Estrutural sob Cargas Reais de Serviço Volume II*. Scotts Valley: CreateSpace/Amazon.

Three algorithms for converting color to grayscale. (11 de Junho de 2018).

Fonte:

[www.johndcook.com:](http://www.johndcook.com/)

<https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>

Useful Color Equations. (11 de Junho de 2018). Fonte:

[http://www.brucelindbloom.com:](http://www.brucelindbloom.com/)

<http://www.brucelindbloom.com/index.html?Equations.html>