

Detecção de Trincas e Buracos em Estradas Asfaltadas

Mateus Junior Cassaniga

Mestrado em Computação Aplicada – Universidade do Vale do Itajaí (UNIVALI)
Caixa Postal 360 – CEP 88302-202 – Itajaí – SC – Brasil

mateus.cassaniga@gmail.com

Abstract. *The road condition is an essential factor for traffic safety. Currently, in many cases, the detection of potholes and cracks in asphalt pavements is manually performed, resulting in high costs and delays. This paper demonstrates the use of image processing techniques for detection of potholes and cracks in images of asphalt pavements. We performed experiments with 96 images, obtaining a precision of 78.5% and recall of 82.5%.*

Resumo. *A condição das estradas é um fator essencial para a segurança no trânsito. Atualmente, em muitos casos, a detecção de buracos e trincas em estradas asfaltadas é feita de forma manual, gerando custos elevados e demora. Este artigo demonstra a utilização de técnicas de processamento de imagem para a detecção de buracos e trincas em imagens de ruas asfaltadas. Foram realizados testes com 96 imagens, alcançando uma precisão de 78,5% e recall de 82,5%.*

1. Introdução

Trinca é o nome dado a problemas que surgem em pavimentos flexíveis que são estruturalmente inadequados para as cargas de tráfego a que estão sujeitos. Estes problemas permitem a infiltração da água, ampliando a área enfraquecida e dando origem a um buraco [Brasil 2005]. Os buracos ou panelas em estradas asfaltadas são problemas de ruptura estruturais localizados, que iniciam em locais que estão mais enfraquecidos que seu entorno. Os buracos afetam diretamente a segurança do tráfego e caso não sejam reparados, rapidamente se expandem comprometendo ainda mais a segurança no local. São causados principalmente por excesso de carga por eixo dos veículos, problemas de projeto e construção e ação da água devido às infiltrações [Brasil 2005].

Atualmente, em muitos casos, a detecção de buracos e trincas é feita manualmente. A detecção manual destes problemas acaba sendo demorada e tendo um custo elevado. A fim de melhorar esta etapa de detecção, diversos pesquisadores e empresas buscam por soluções para resolver este tipo de problema.

Recentemente diversos métodos para detecção de buracos e trincas foram propostos. Os métodos de detecção de buracos e trincas estão divididos em três abordagens diferentes: (1) métodos que utilizam técnicas de detecção de buracos e trincas por vibração; (2) métodos que utilizam o varrimento de lasers para reconstrução 3D; e (3) métodos baseados em visão [Kim e Ryu 2014]. Os métodos baseados em vibrações se destinam apenas a levantamentos preliminares já que carecem de precisão e confiabilidade. Os métodos baseados em reconstrução 3D sofrem basicamente de custos elevados de equipamentos [Koch e Brilakis 2011]. Por fim, os métodos baseados em visão utilizam câmeras e imagens 2D para detectarem os buracos e trincas presentes nas

estradas asfaltadas. Esta técnica proporciona um baixo custo, já que câmeras comuns podem ser utilizadas na captura das imagens.

Neste artigo, é apresentada uma solução para a detecção de buracos e trincas utilizando técnicas de processamento de imagens. As imagens utilizadas foram capturadas do Google® Street View. O método inicia com a segmentação da imagem, realiza operações morfológicas e analisa as partes resultantes. O protótipo desenvolvido foi implementado em Python 3.5.2, utilizando a biblioteca OpenCV 3.1.0. Foram realizados testes com 96 imagens de superfícies contendo asfalto com buracos, trincas e asfalto sem defeitos. Ao final, as técnicas demonstraram-se serem promissoras na resolução do problema.

2. Trabalhos Relacionados

Existem três abordagens distintas para detecção de buracos e trincas em estradas asfaltadas: métodos baseados em vibração, métodos baseados em varrimento laser (reconstrução 3D) e métodos baseados em visão. Neste artigo, o foco principal do estudo são métodos baseados em visão. Estes métodos podem ser subdivididos em duas abordagens: as que utilizam imagens em 2D e as que utilizam abordagens baseadas em vídeo.

2.1. Métodos baseados em visão utilizando imagens 2D

Koch e Brilakis (2011) apresentaram uma nova abordagem para a detecção automatizada de buracos em estradas asfaltadas. O método desenvolvido por eles inicia realizando a segmentação da imagem para dividi-la em regiões defeituosas e não defeituosas. Com base nas propriedades geométricas de uma região com defeito, é feita uma aproximação da forma potencial do buraco. Por fim, é realizada a extração da textura dentro da região com defeito e comparada com a textura das regiões ao redor. Para testar o método proposto, os autores implementam o protótipo no MATLAB, realizando o treinamento manual do algoritmo e testando com 120 imagens de pavimentos, representando diversos tipos de buracos, trincas, remendos e condições de iluminação (sombras). A abordagem proposta pelos autores obteve uma *accuracy* de 85,9%, *precision* de 81,6% e *recall* de 86,1%.

Buza et al. (2014) propuseram um método não supervisionado baseado em análise de imagens e agrupamento espectral que não requer filtros adicionais e fase de treinamento. O método proposto é dividido em três fases: detecção das imagens com defeitos, análise destas imagens através de um algoritmo de agrupamento espectral e por fim, identificação e extração dos buracos. O método foi implementado no MATLAB e os testes foram realizados com 50 imagens. Os autores obtiveram uma *accuracy* de 81% para a estimativa da área de superfície de buracos.

2.2. Métodos baseados em visão utilizando sequências de vídeo

Em seu trabalho anterior, Koch e Brilakis (2011) apresentaram um método para detecção automatizada de buracos em imagens de superfícies asfaltadas baseado em segmentação de imagem, aproximação da forma do buraco e comparação de texturas. Baseando-se no método anterior, Koch et. al. (2013) aprimoraram o método para detecção de buracos realizando alterações no método de detecção de texturas. O método também realiza a marcação e acompanhamento dos buracos detectados em sequências de vídeo, permitindo

a contagem dos buracos. O método proposto foi desenvolvido no MATLAB e testado com 39 vídeos, contendo mais de 10.000 frames, provendo uma grande variedade de buracos, trincas e remendos. Os autores obtiveram uma *precision* de 75% e *recall* de 84%. Comparado com seu trabalho anterior, os autores conseguiram um aumento de 53% na performance de comparação de texturas e uma redução de 57% no tempo computacional.

Huidrom et al. (2013) apresentaram um método para detecção automatizada de buracos, trincas e remendos em sequências de vídeo de rodovias indianas. No método proposto, os defeitos são detectados e quantificados usando várias técnicas de processamento de imagem suportadas por uma heurística de lógica de decisão. O método proposto é dividido em dois passos: inicialmente é realizada a separação dos frames em defeituosos e não defeituosos, utilizando um algoritmo chamado de *Distress Frames Selection* (DFS). Em seguida os frames que possuem defeitos são analisados e classificados através de um algoritmo chamado *Critical Distress Detection, Measurement and Classification* (CDDMC). O algoritmo CDDMC utiliza informações das propriedades visuais (calculando o desvio padrão, circularidade e largura) destes defeitos para a classificação. O método proposto classifica os defeitos em quatro categorias distintas: imagens com buracos, imagens com trincas, imagens com remendos e imagens sem defeitos. Os testes foram realizados em 1275 frames de vídeos selecionados aleatoriamente após a aplicação do algoritmo DFS em vários vídeo cliques. Os autores obtiveram uma *accuracy* de 97% com 95% de *precision* e 81% de *recall* na detecção de frames com buracos, 94% de *accuracy* com 93% de *precision* e 98% de *recall* na detecção de frames com trincas e 90% de *accuracy* com 8,5% de *precision* e 19% de *recall* na detecção de frames com remendos.

3. Metodologia

O objetivo deste artigo é a experimentação de técnicas de processamento de imagem para a detecção de buracos e trincas em imagens de estradas asfaltadas. O artigo teve como base as técnicas propostas por Huidrom et al. (2013), aplicadas em um *dataset* criado especificamente para o protótipo.

A abordagem utilizada na detecção de buracos e trincas é baseada na ideia de detecção total ou parcial de imperfeições que representam tais defeitos. Para a identificação de buracos e trincas, considerou-se que as principais características destes defeitos são:

- a) Buracos e trincas possuem sombras e são mais escuras que o resto da imagem da estrada;
- b) A textura interna de buracos e trincas possui uma variação e contraste maior que o restante da imagem da estrada.

Tendo como base estas duas características, o método proposto foi dividido em três etapas: (1) segmentação da imagem, onde são separados os possíveis defeitos; (2) remoção de objetos pequenos que não serão considerados; e (3) validação de defeitos encontrados, onde serão determinados se os itens serão ou não considerados defeitos.

3.1. Segmentação da imagem

Na etapa de segmentação da imagem, as informações das cores (valores RGB) não foram considerados essenciais para a detecção de buracos e trincas. Sendo assim, a primeira etapa da segmentação de imagem é a transformação da imagem colorida para escala de

cinza. Em seguida, um filtro de mediana de tamanho 5×5 é aplicado sobre a imagem para a remoção de pequenos ruídos que podem comprometer a etapa de segmentação.

Esta etapa do processamento tem como objetivo encontrar e separar os pontos mais escuros da imagem. A tarefa pode ser realizada utilizando técnicas de limiarização. Para realizar a separação destes pontos foi utilizado uma técnica de limiarização adaptativa. Este tipo de limiarização é utilizado por melhor se adaptar a imagens onde a iluminação ou refletância é muito variável [Gonzalez e Wood 2010]. A fórmula do algoritmo utilizado pode ser visualizada na Equação (1):

$$f(x) = \begin{cases} 1, & I(x, y) > T(x, y) - C \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

Onde $f(x)$ é a imagem destino, $I(x, y)$ é o bloco que está sendo limiarizado e $T(x, y) - C$ é a média dos valores dos pixels do bloco menos uma constante C . O bloco consiste nas várias subdivisões da imagem. A Equação (2) descreve o cálculo para definir o tamanho do bloco.

$$tamanho = \frac{\max(w, h)}{D} \quad (2)$$

Onde $\max(w, h)$ é o maior valor entre a largura e a altura da imagem e D uma constante. A Figura 1 apresenta o resultado a limiarização adaptativa em uma imagem.

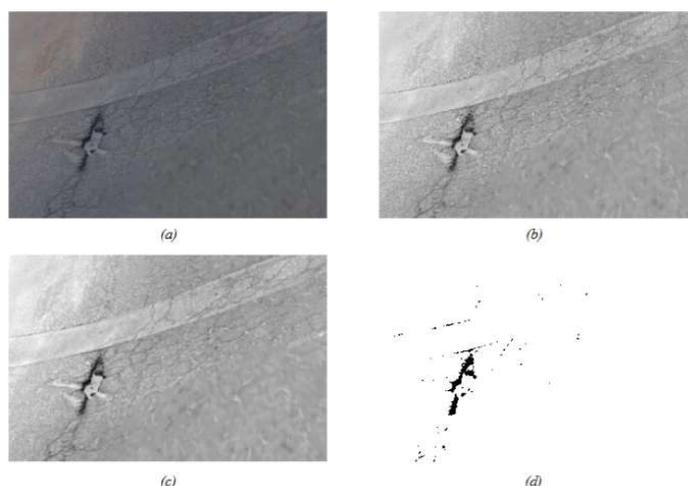


Figura 1. Sequência de passos para a limiarização

Na imagem acima, (a) representa a imagem original, (b) representa a imagem transformada para tons de cinza, (c) demonstra a imagem após a aplicação do filtro de mediana 5×5 e (d) demonstra a limiarização adaptativa de (c).

3.2. Remoção de objetos desnecessários

Após a etapa de segmentação, é necessário realizar uma limpeza na imagem a fim de remover objetos segmentados e que são desnecessários. Para isso são realizadas operações morfológicas de erosão e dilatação com *kernel* circular de tamanhos 7×7 e 5×5 . Esta etapa do processamento tem como objetivo remover pontos isolados da segmentação e conexão de pontos próximos. Em seguida as cores da imagem são invertidas para que um algoritmo de contorno possa ser aplicado. O algoritmo de contorno é utilizado para calcular a área de cada um dos objetos segmentados. Os objetos que

possuem uma área inferior ao valor da *area*, calculada pela Equação (3), são desconsiderados.

$$area = (w * h) * P \quad (3)$$

Onde $(w * h)$ representa a área total da imagem e P uma constante que representa um percentual deste tamanho (fixado em 0.0005). Com isso, objetos com tamanho menor que 0,05% da área total da imagem são descartados. A Figura 2 apresenta o resultado desta etapa.

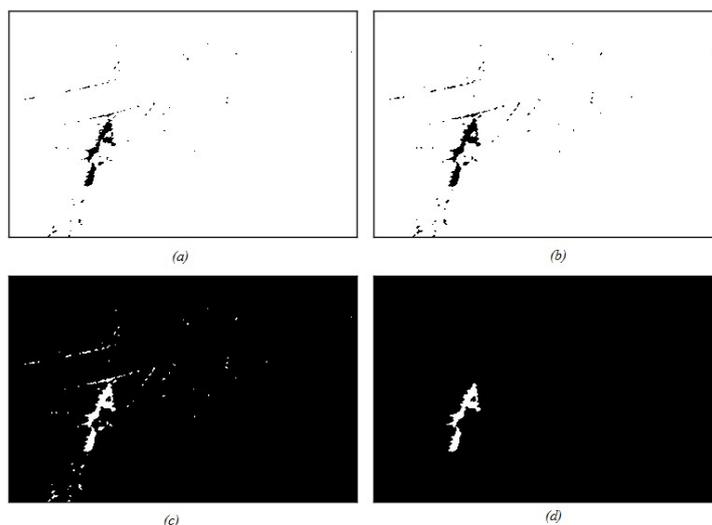


Figura 2. Remoção de objetos desnecessários

Na imagem acima, (a) representa a imagem segmentada, (b) após as operações morfológicas de erosão e dilatação, (c) a inversão das cores e (d) a imagem sem os objetos menores que 0,05%.

3.3. Validação e classificação dos defeitos

A última etapa verifica os objetos resultantes das etapas anteriores e os define como buracos, trincas ou se devem ser desconsiderados.

Inicialmente é realizada uma dilatação com um *kernel* circular de 3×3 para conectar e melhorar a forma dos objetos. Após isto são calculados os contornos de cada um dos objetos resultantes. Então para cada objeto são calculadas sua área, desvio padrão e média. Um objeto é considerado um buraco ou trinca apenas se:

- Sua área for superior ao resultado da Equação (3);
- Seu desvio padrão for maior que o desvio padrão da imagem; e
- A média dos seus pixels for menor que a média dos pixels da imagem.

As regras acima definem as características de um buraco: eles devem possuir ao menos 0,05% da área da imagem, possuir uma variação de contraste maior que os demais locais e serem mais escuros que o restante da imagem. A Figura 3 apresenta o resultado desta operação.



Figura 3. Identificação do defeito

Na figura, (a) representa a imagem original e (b) representa o buraco ou trinca detectado.

4. Testes e Resultados

O protótipo do algoritmo foi implementado utilizando a linguagem de programa Python 3.5.2 em conjunto com a biblioteca OpenCV 3.1.0.

4.1. Criação do *dataset*

Um *dataset* foi exclusivamente criado para a realização dos testes. Várias imagens de resolução 13312×6656 foram extraídas manualmente do Google® Street View. A partir destas, foram recortadas 96 partes distintas contendo nenhum, um ou vários buracos e/ou trincas. A Figura 4 exemplifica como foram extraídas as partes das imagens.

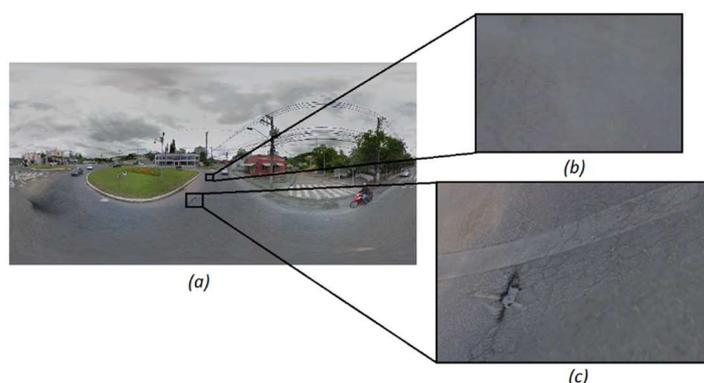


Figura 4. Extração das partes

Na Figura 4, (a) demonstra uma imagem original de resolução 13312×6656 e (b) e (c) as duas partes extraídas. (b) é um exemplo de uma imagem sem asfalto danificado e (c) apresenta a imagem com asfalto danificado. As informações de todas as partes extraídas estão dispostas na Tabela 1.

Tabela 1. Resumo do *dataset* montado

Uso	Imagens com asfalto danificado	Imagens sem asfalto danificado	Total
Ajuste de parâmetros	15	22	37
Testes	29	30	59
Total	44	52	96

Na tabela acima, são descritas 44 partes de imagens contendo buracos e/ou trincas. Nestas 44 imagens foram realizados procedimentos manuais de identificação destes defeitos (buracos e/ou trincas) apresentados. Uma imagem pode conter vários defeitos distribuídos nela. A Tabela 2 apresenta a quantidade total de defeitos selecionados.

Tabela 2. Quantidade de defeitos selecionados

Dataset	Buracos/trincas manualmente identificados
Ajuste de parâmetros	19
Testes	40
Total	59

Dentro das 44 imagens de asfaltos danificados, foram manualmente selecionadas 59 áreas identificadas como buracos ou trincas.

4.2. Escolha do conjunto de parâmetros

A escolha do melhor conjunto de parâmetros foi definida com base na *precision* e *recall* dos resultados obtidos com o determinado conjunto. *Precision* indica quantos buracos ou trincas identificadas pelo método estão realmente corretos e o *recall* indica quantos buracos ou trincas do total foram identificados. *Precision* e *recall* foram calculados conforme Equação (4):

$$Precision (\%) = VP / (VP + FP) * 100 \quad (4)$$

$$Recall (\%) = VP / (VP + FN) * 100$$

Onde *VP* é a quantidade de Verdadeiros Positivos, *FP* a quantidade de Falsos Positivos e *FN*, os Falsos Negativos.

Para a definição dos melhores parâmetros para a etapa de segmentação, foram realizados diversos testes alterando os valores dos parâmetros do método de limiarização adaptativa. Durante os testes, notou-se que o parâmetro *D* utilizado para definir o tamanho do bloco não tem um impacto significativo nos resultados, conforme demonstrado na Figura 5:

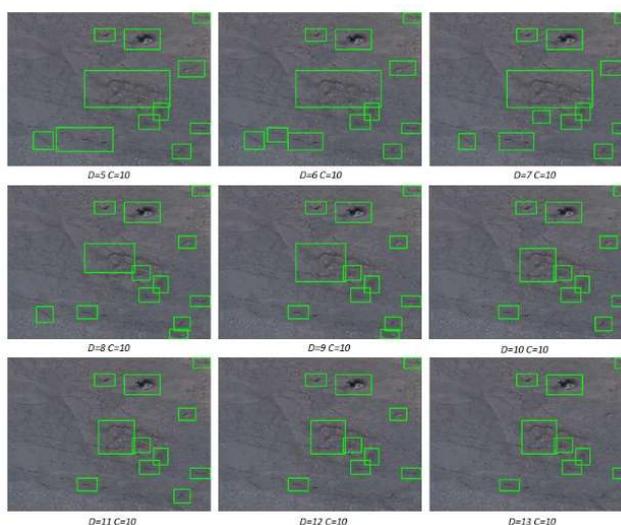


Figura 5. Variação do parâmetro *D* do cálculo do tamanho do bloco

Na Figura 5 são apresentados 9 resultados alterando apenas o valor do parâmetro D da fórmula ($D = 5 \sim 13$). Nota-se que a variação nos resultados se dá, principalmente, no tamanho das regiões encontradas, porém não se tem diminuição expressiva nos falsos positivos.

O parâmetro C , responsável por definir o valor de corte no algoritmo de limiarização adaptativa demonstrou-se o principal parâmetro do método utilizado, sendo que este é responsável por diminuir expressivamente os falsos positivos. A Figura 6 apresenta os resultados:

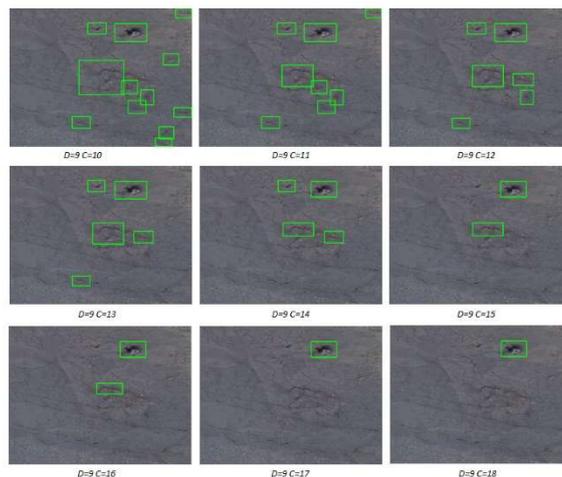


Figura 6. Variação do parâmetro C utilizado como valor de corte

Na imagem acima, ao incrementar o valor do parâmetro C , tem-se uma diminuição progressiva na quantidade de possíveis defeitos encontrados.

Com base nos testes, foram realizados manualmente os cálculos de *precision* e *recall*. Para a determinação de valores VP , foram considerados todos os buracos ou trincas detectadas, mesmo que parcialmente. Com base nisto, o melhor conjunto de parâmetros encontrado ($C = 18$ e $D = 9$) obteve uma *precision* de 78,5% e um *recall* de 82,5%, conforme Tabela 3.

Tabela 3. Performance do método utilizado

Performance	
Total Verdadeiro Positivo (VP)	33
Total Falso Positivo (FP)	9
Total Falso Negativo (FN)	7
<i>Precision</i>	78,5%
<i>Recall</i>	82,5%

Ao analisar os resultados, constatou-se que o algoritmo é bastante sensível a iluminação. Em imagens onde estão presentes sombras fortes, em muitos casos, o método proposto interpreta como buracos ou trincas. Notou-se também, que em buracos onde o interior é uniforme (como no caso de poças d'água), o algoritmo acaba descartando-o pelo fato de possuir um desvio padrão menor que o exterior. Em muitos casos, a identificação do defeito é feita parcialmente, ou de maneira fragmentada (quando o

algoritmo interpreta um defeito como vários). A Figura 7 exemplifica os problemas encontrados com o protótipo implementado.

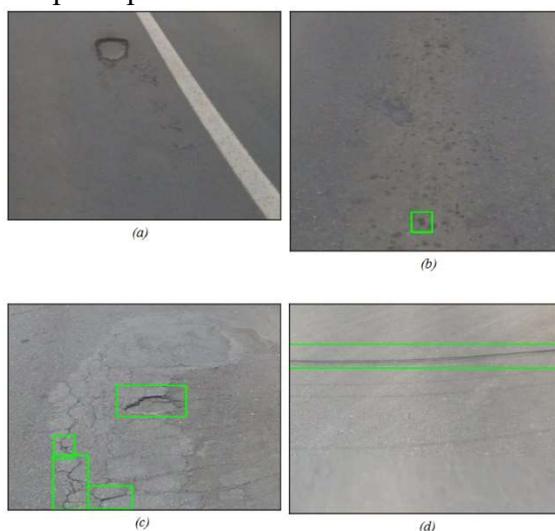


Figura 7. Problemas encontrados com o método utilizado

Na Figura 7, (a) demonstra um buraco não detectado, (b) e (d) sombras ou sujeira classificados como defeitos e (c) a fragmentação da identificação.

5. Conclusão e Trabalhos Futuros

A condição das estradas é um fator fundamental para a segurança no trânsito. A identificação de defeitos em rodovias feita manualmente é custosa e demorada. Existem diversas abordagens para a detecção destes defeitos em rodovias, como métodos que utilizam vibração e métodos que utilizam lasers, porém estes métodos são limitados ou possuem um custo elevado de equipamento. A utilização de métodos baseados em visão permite a identificação de defeitos a um custo aceitável com uma boa precisão.

Este artigo demonstrou a utilização de técnicas de processamento de imagens utilizando segmentação e transformações morfológicas para a identificação de trincas e buracos em estradas asfaltadas. A base desta experimentação foram as técnicas propostas por Huidrom et al. (2013), em seu artigo. O método utilizado demonstrou ter uma taxa aceitável de *precision* e *recall*, porém constatou-se que o mesmo é bastante sensível a iluminação.

A utilização de imagens extraídas do Google® Street View permitiu a realização dos testes e validação dos resultados. Um banco de dados maior e com imagens com melhor resolução permitiriam a melhor análise e desenvolvimento do protótipo. As imagens utilizadas não foram suficientes para encontrar o melhor resultado para o algoritmo proposto. As técnicas utilizadas, como análise do desvio padrão seriam melhor empregadas caso a resolução das imagens do asfalto fossem maiores. A utilização de uma câmera apontada para o asfalto a uma distância pré-definida permitiria a melhor identificação dos defeitos abordados.

A utilização de algoritmos para comparação de texturas poderia ser empregado para a melhor classificação dos defeitos, melhorando a identificação de falsos positivos. Também notou-se que a utilização do método proposto, em muitos casos, permite apenas a identificação parcial de buracos e trincas.

No futuro, pretende-se melhorar a detecção de buracos e trincas com (a) utilização de um *dataset* maior e com uma variedade maior de defeitos (incluindo remendos), contendo imagens de resolução maior e a uma distância fixa da estrada, (b) utilização de outras técnicas que considerem a análise de textura na detecção dos defeitos e (c) a classificação dos defeitos encontrados.

Referências

- Brasil. Departamento Nacional de Infra-Estrutura de Transportes. Diretoria de Planejamento e Pesquisa. Coordenação Geral de Estudos e Pesquisa. Instituto de Pesquisas Rodoviárias. Manual de conservação rodoviária. 2. ed. – Rio de Janeiro, 2005. 564p.
- Buza, E., Omanovic, S., Huseinovic, A. (2014). Pothole Detection with Image Processing and Spectral Clustering. *Recent Advances in Computer Science and Networking*, pages 2–7.
- Gonzalez, R. C. Woods, R. E. Processamento Digital de Imagens. 3. ed. São Paulo: Pearson Prentice Hall, 2010.
- Huidrom, L., Das, L. K., Sud, S. K. (2013). Method for automated assessment of potholes, cracks and patches from road surface video clips. *Procedia - Social and Behavioral Sciences*, v. 104, pages 312–321. Elsevier B.V.
- Kim, T., Ryu, S. (2014). Review and Analysis of Pothole Detection Methods. *Journal of Emerging Trends in Computing and Information Sciences*, v. 5, pages 603–608.
- Koch, C., Brilakis, I. (2011). Pothole detection in asphalt pavement images. *Advanced Engineering Informatics*, v. 25, pages 507–515. Elsevier Ltd.
- Koch, C., Jog, G. M., Brilakis, I. (2013). Automated Pothole Distress Assessment Using Asphalt Pavement Video Data. *Journal of Computing in Civil Engineering*, v. 27, Issue 4, pages 370–378.