

MENSURAÇÃO DE PEGADAS DE MAMÍFEROS EM LEVANTAMENTOS FAUNÍSTICOS POR MEIO DE SMARTPHONE

Roberson Junior Fernandes Alves¹

Paulo Henrique Ferreira²

RESUMO

Este artigo teve por objetivo apresentar o estudo e o desenvolvimento de uma aplicação para auxiliar a mensuração de pegadas de animais em levantamentos faunísticos; foi desenvolvido a fim de promover uma forma mais eficaz, prática e fácil para a mensuração das pegadas dos animais. Tendo em vista a forma de análise das pegadas em levantamentos faunísticos, há uma grande necessidade de obter velocidade na identificação destas, sendo que para cada pegada identificada utilizam-se métodos bastantes antigos, e o excesso de bagagem e peso se torna um fator a se levar em conta, sendo que os locais das amostragens geralmente são de difícil acesso. O *app* foi projetado para a plataforma Android utilizando a biblioteca OpenCV para o tratamento digital das imagens obtidas. Com base nos experimentos realizados, os resultados mostram que ainda existem melhorias e testes a serem feitos, contudo o sistema se mostra promissor.

Palavras-chave: Levantamentos faunísticos. OpenCV. Pegadas. Frintter.

1 INTRODUÇÃO

Atualmente a expansão comercial e industrial é de muitas formas prejudicial ao meio ambiente. Para cada empreendimento industrial, como uma usina hidroelétrica ou uma mineradora, que necessitam de lugares maiores para serem empreendidas, os órgãos responsáveis pelo meio ambiente devem fiscalizar os impactos ambientais que serão gerados ao ser incluído tal empreendimento.

Em um caso específico de fiscalização está o levantamento faunístico do local, por meio do qual é realizada uma série de coletas de dados sobre os animais que residem naquele local; tais dados são levantados para tentar amenizar a diminuição extrema de espécies de animais, alguns até já ameaçados de extinção.

Levantamentos faunísticos consistem em coletar sinais ou amostras de animais que passaram por algum local específico da coleta. Estes estão divididos em dois grupos, amostragem direta, em que o animal está presente, como quando é capturada uma fotografia deste, ou armadilhas fotográficas. E a amostragem indireta que, por sua vez, é realizada a partir de vestígios dos animais, como pegadas, fezes e pelos.

Para casos de amostragem indireta de pegadas, são retiradas imagens fotográficas utilizando objetos como referência para medição. Em alguns casos é feito um molde de gesso para ser levado a laboratório a fim de se analisarem e medirem as marcas. Esse procedimento leva muito tempo e resulta em muita carga para o pesquisador, o que inviabiliza a coleta de maior quantidade de dados.

Com isso, foi percebida a necessidade de informatizar a parte de coleta de amostras indiretas por pegadas, desenvolvendo um aplicativo móvel (*app*) que realize a mensuração das pegadas no momento em que é retirada uma fotografia destas.

Para serem feitos os processos de tratamento de imagens, como localização de marcador e a utilização da câmera, foi utilizada a biblioteca OpenCV, que é designada para esse fim. Em todo o processamento foram utilizadas as funções disponibilizadas pela biblioteca, com exceção do cálculo-base para as medições.

¹ Graduando em Sistemas de Informação pela Universidade do Oeste de Santa Catarina de São Miguel do Oeste; roberson.alves@unoesc.edu.br

² Mestre em Computação Aplicada pela Universidade Estadual de Ponta Grossa; Professor no Curso de Ciência da Computação da Universidade do Oeste de Santa Catarina de São Miguel do Oeste; p.h.ferreirah@gmail.com

Pensando na disponibilidade deste *app*, ele foi desenvolvido para plataforma Android, por ter acesso mais fácil a aplicações e ser *open source*.

Para que se tenha um melhor entendimento da problemática que este artigo visa apresentar, as seções 2 e 3 deste artigo trarão informações importantes para o leitor a fim de esclarecer a ideia de levantamentos faunísticos e processamento digital de imagens, respectivamente. Na seção 4 são mostrados as ferramentas e os métodos utilizados para o desenvolvimento do *app*, bem como a importância de seu uso para o projeto. Já a seção 5 será responsável por mostrar o funcionamento do *app*. Na seção 6 serão apresentados os resultados do projeto. Por fim, na seção 7 são apresentadas as conclusões parciais obtidas nessa etapa do desenvolvimento.

2 LEVANTAMENTOS FAUNÍSTICOS

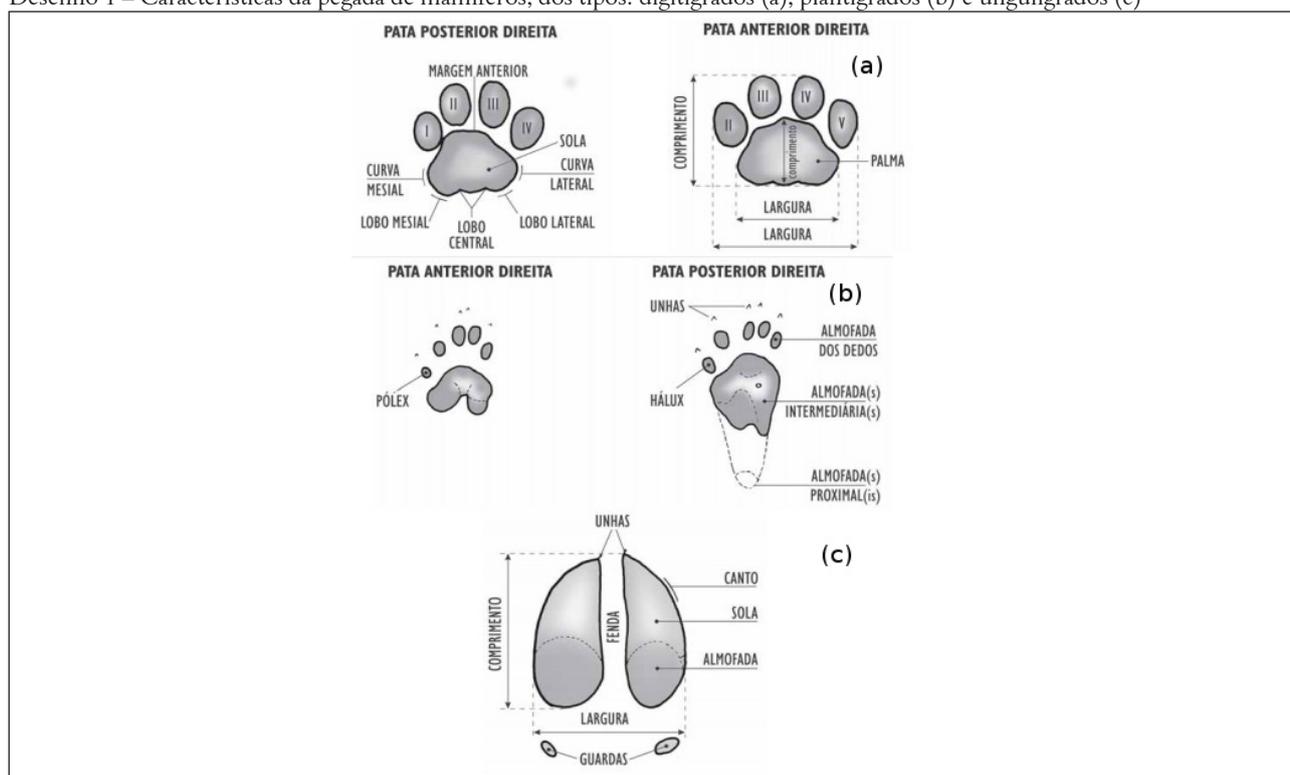
Levantamentos faunísticos são realizados para se analisar o impacto gerado pela implantação de um empreendimento ou um desastre ambiental. Todo empreendimento que é feito em uma região de mata nativa gera um impacto nos grupos de animais silvestres que habitam no local; considerando animais que estão ameaçados de extinção, é de extrema importância levantar tais dados para tomar uma ação preventiva (NASCIMENTO, 2011; GOVERNO DO ESTADO DE SANTA CATARINA, 2010; MIRANDA, 2008; SILVEIRA, 2010).

Para serem feitas análises é preciso fazer um levantamento da forma mais precisa possível dos animais que vivem no meio em questão; para isso, opta-se por várias técnicas de identificação. Para avaliar grupos de médios e grandes portes de animais, devem ser empregados alguns métodos de identificação, entre os quais armadilhas fotográficas, acervo fotográfico pessoal e registros de sinais, como pegadas e fezes (NASCIMENTO, 2011; GOVERNO DO ESTADO DE SANTA CATARINA, 2010; MIRANDA, 2008; SILVEIRA, 2010).

Os levantamentos faunísticos de pegadas também são importantes em trabalhos ecológicos para se avaliar a dinâmica das populações de animais em uma área. Esse tipo de avaliação é diferente do já citado, pois se trata de avaliação de impacto ambiental (BARBIERI, 1995).

O Desenho 1 ilustra algumas partes de pegadas que são usadas para serem feitas as devidas medidas a fim de caracterizá-la. Tais medidas devem ser realizadas com uma precisão considerável, pois com elas são definidas as características dos animais, como sexo e idade.

Desenho 1 – Características da pegada de mamíferos, dos tipos: digitígrados (a), plantígrados (b) e unglígrados (c)



Fonte: Moro-Rios et al. (2008, p. 12-13).

Visando ao melhor desempenho no levantamento desses dados optou-se por utilizar técnicas de processamento digital de imagens, que auxiliam de forma mais prática e eficiente em tais situações.

3 PROCESSAMENTO DIGITAL DE IMAGENS

O PDI é a área da computação que simula a visão humana. Também “Podemos dizer que processamento de imagens é um processo onde a entrada do sistema é uma imagem e a saída é um conjunto de valores numéricos, que podem ou não compor uma outra imagem.” (MARENGONI; STRINGHINI, 2009).

A OpenCV é um conjunto de ferramentas de programação para desenvolvimento de aplicações com Visão. Ela engloba também outro conceito importante, especialmente no meio acadêmico, o software livre. A biblioteca é completamente *open source*, e é distribuída gratuitamente, aberta a colaborações de qualquer indivíduo ou empresa voluntários. (DELAÍ; COELHO, 2018, grifo do autor).

A biblioteca OpenCV é comumente utilizada em *apps* móveis que possuem como objetivo principal o tratamento e/ou a edição de imagens, por atender às necessidades dos *apps* móveis, não utilizando processamento desnecessário e atendendo aos requisitos desejados, sem que haja perda de desempenho.

4 MATERIAIS E MÉTODOS

A aplicação foi desenvolvida utilizando-se a IDE Android Studio, especificamente para o desenvolvimento para a plataforma Android, em que foram configurados o SDK e o NDK, a fim de ter disponíveis no ambiente de desenvolvimento todas as funcionalidades necessárias para desenvolver um aplicativo para Android.

No processo de visão computacional foi utilizada a biblioteca OpenCV para a identificação de um marcador, o qual foi utilizado para garantir a mensuração mais precisa. A versão utilizada do OpenCV foi a 3.1.0 para Android, que está disponível atualmente. Após compilada, a biblioteca foi integrada ao projeto por meio das configurações do ambiente de desenvolvimento.

Toda a parte de reconhecimento do marcador é realizada pelo OpenCV, desde a localização do marcador na imagem até a criação da régua auxiliar e os pontos de mensuração.

O reconhecimento do marcador é realizado pelo algoritmo *Oriented FAST and Rotated BRIEF (ORB)*, que tem como especialidade detectar características que o permitam reconhecer uma imagem (LOWE, 1999). Ele começa seu processamento pela execução do algoritmo *Features from Accelerated Segment Test (FAST)* para encontrar pontos de destaque na imagem e, em seguida, para computar esses pontos de destaque, ele utiliza o descritor *Binary Robust Independent Elementary Features (BRIEF)*. O algoritmo surgiu como alternativa aos algoritmos como o *Speed-Up Robust Features (SURF)* e o *Scale-Invariant Feature Transform (SIFT)*. O ORB é um algoritmo aberto e tem apresentado bons resultados no reconhecimento de imagens (SANTANA, 2015).

Uma vez reconhecido o marcador, é realizado o cálculo da relação de tamanho da pegada com relação ao marcador. O cálculo realizado para formar a régua auxiliar é basicamente $M=N/L$, sendo que N é o lado do marcador e é conhecido (em testes foi utilizado um quadrado com lados de 15x15 cm), e L é a quantidade de pixels do ponto inicial e final (x, y) que formam o lado identificado do marcador. Então, M é o quociente da divisão de N por L (FRENSEL; DELGADO, 2011).

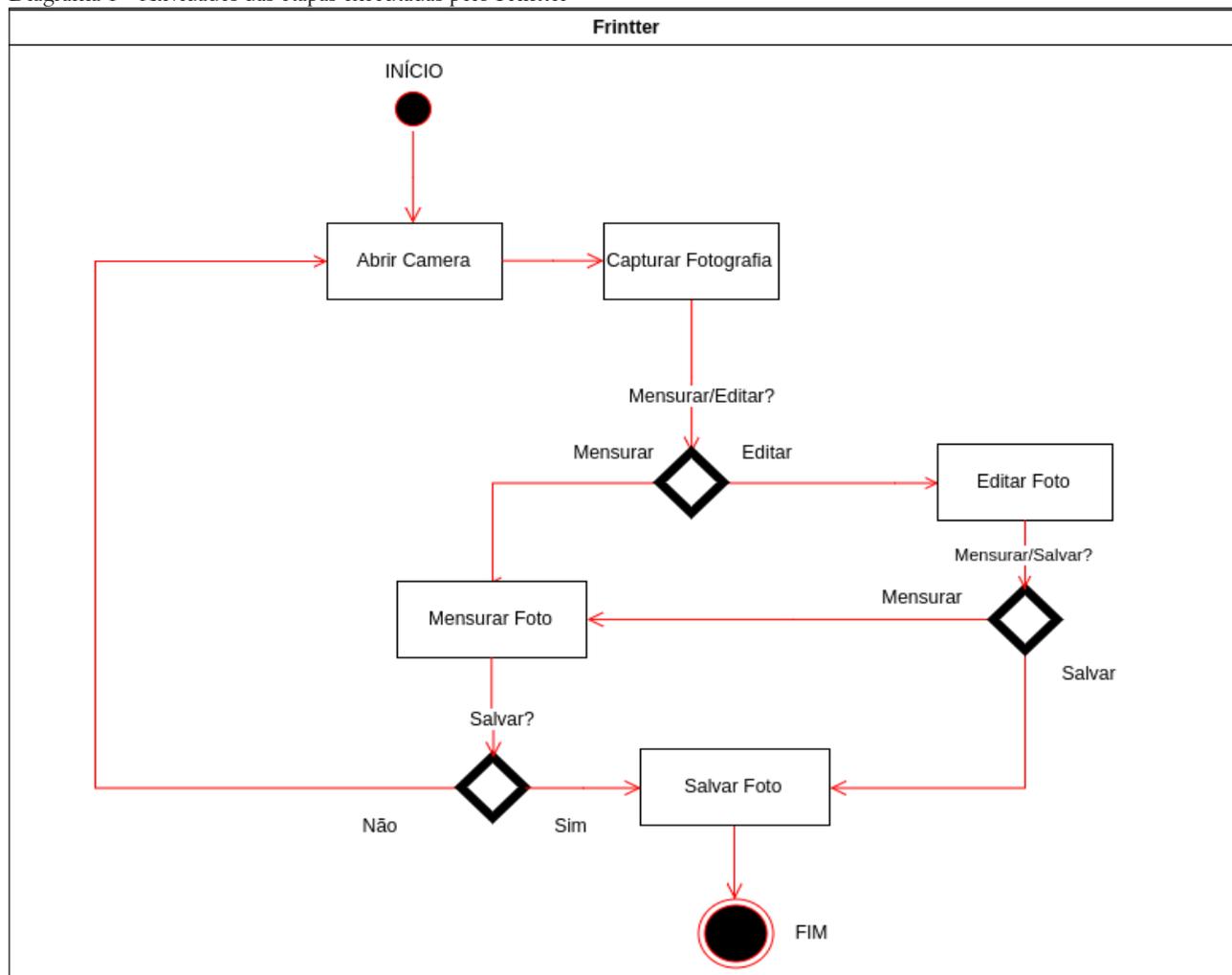
Assim, tendo conhecimento da dimensão de cada pixel na imagem é possível obter uma medida praticamente exata ao mundo real. Os pontos de mensuração são realizados a partir desse cálculo inicial.

As imagens retiradas são armazenadas em formato padrão *JPEG (Joint Photographics Experts Group)*, facilitando a visualização em outros dispositivos. A imagem é salva inicialmente original, com o marcador, a mostra e a régua auxiliar, ainda é possível editá-la com *app* de edição disponíveis nos *smartphones*, e ambas as imagens são mantidas.

Os testes foram realizados com dois *smartphones*, um Samsung Galaxy S6 e um LG Optimus L7 P716, nos quais foi instalada a biblioteca OpenCV e realizados testes de desempenho do *app*. O desempenho e produtividade dos *smartphones* na execução do aplicativo, assim como os resultados obtidos, são apresentados na seção de resultados parciais.

Para simplificar a utilização do aplicativo foi modelado um diagrama de atividades, no qual são mostradas as etapas executadas pelo *app* (Diagrama 1).

Diagrama 1 – Atividades das etapas executadas pelo Frintter



Fonte: os autores.

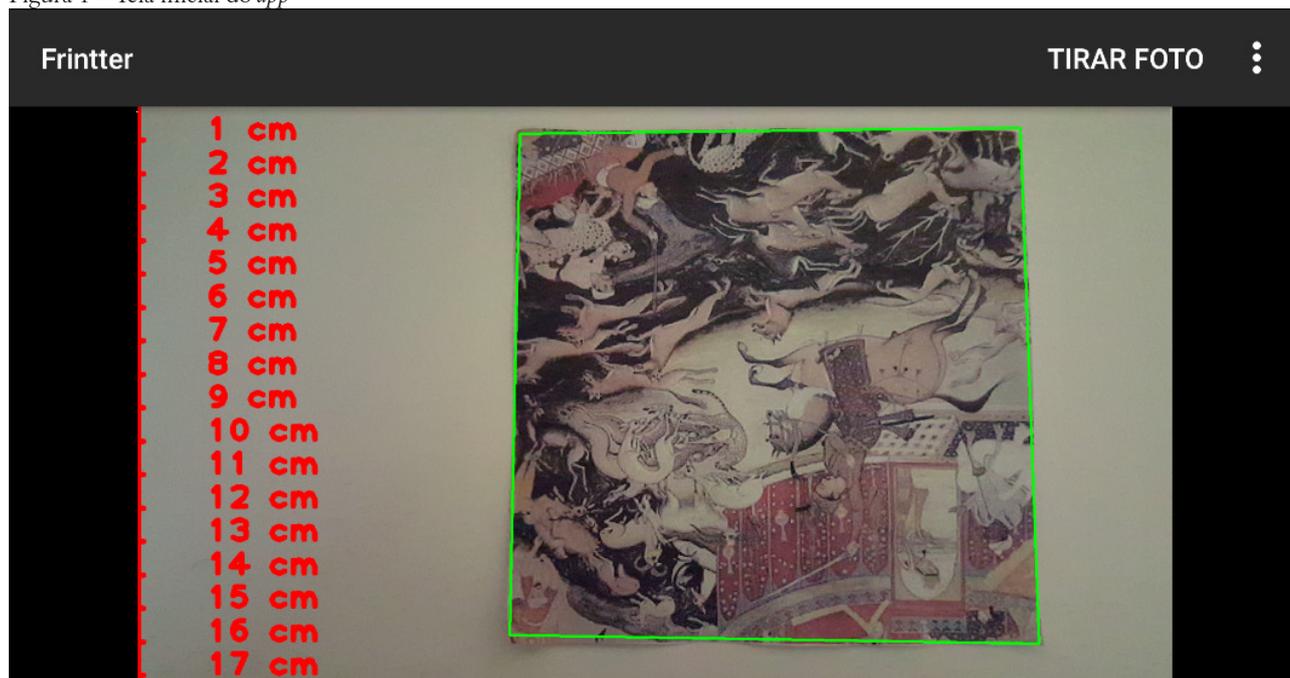
Com o diagrama de atividades é possível analisar o fluxo de atividades em um único processo. O diagrama mostra como uma atividade depende da outra. Como, por exemplo, só é possível tirar uma fotografia se a câmera estiver aberta/ligada. Na seção 5 será mostrada com mais detalhes cada atividade executada.

5 FRINTTER

Nesta seção são apresentadas as funcionalidades do *app*, assim como sua importância para que as informações sejam trazidas da forma mais correta possível.

O nome do *app* foi definido como Frintter, uma combinação das palavras *footprint* (pegada) e *ruler* (régua), ou seja, régua de pegadas. O *app* tem como principal função o auxílio na mensuração de pegadas de animais por meio de PDI. O processo de mensuração é exibido ao usuário de uma forma bem simples: é exibida uma imagem do marcador no canto esquerdo da tela; quando é feito o reconhecimento desse marcador essa imagem não fica mais visível ao usuário; também é exibida a régua auxiliar ao lado esquerdo, a fim de informar que já é possível ser tirada uma fotografia. Como é mostrado na Figura 1.

Figura 1 – Tela inicial do app



Fonte: os autores.

Para que esse processamento inicial seja feito corretamente é necessário seguir alguns passos importantes. Inicialmente deve-se possuir um marcador impresso – o tamanho é pré-definido em 15 cm –, a biblioteca OpenCV é responsável por localizar esse marcador. Existe uma função nessa biblioteca chamada *onCameraFrame*, que captura cada quadro da câmera e analisa se o marcador está nesse quadro; para cada quadro é feita uma busca e impressa na tela uma marcação ao redor desse marcador indicando a sua posição na imagem, como pode ser visto na Listagem de código 1 (Figura 2). Dentro do método *apply* foi desenvolvida uma rotina para desenhar na tela a régua auxiliar utilizando o cálculo que foi descrito em materiais e métodos.

Figura 2 – Listagem de código 1: Código da função responsável por aplicar as marcações e a régua auxiliar

```

@Override
public Mat onCameraFrame(final CvCameraViewFrame inputFrame) {
    final Mat rgba = inputFrame.rgba();
    if (mImageDetectionFilters != null) {
        mImageDetectionFilters[mImageDetectionFilterIndex].apply(
            rgba, rgba);
    }
    ...
}

Imgproc.line(dst, start, new Point(start.x, start.y + dY), new Scalar(255, 0, 0), 4);
start = new Point(start.x, start.y + dY);
Imgproc.line(dst, new Point(0, start.y), new Point(10, start.y), new Scalar(255, 0, 0));
Imgproc.putText(dst, "    " + (i + 1) + " cm", start, 1, 0.5, new Scalar(255, 0, 0));

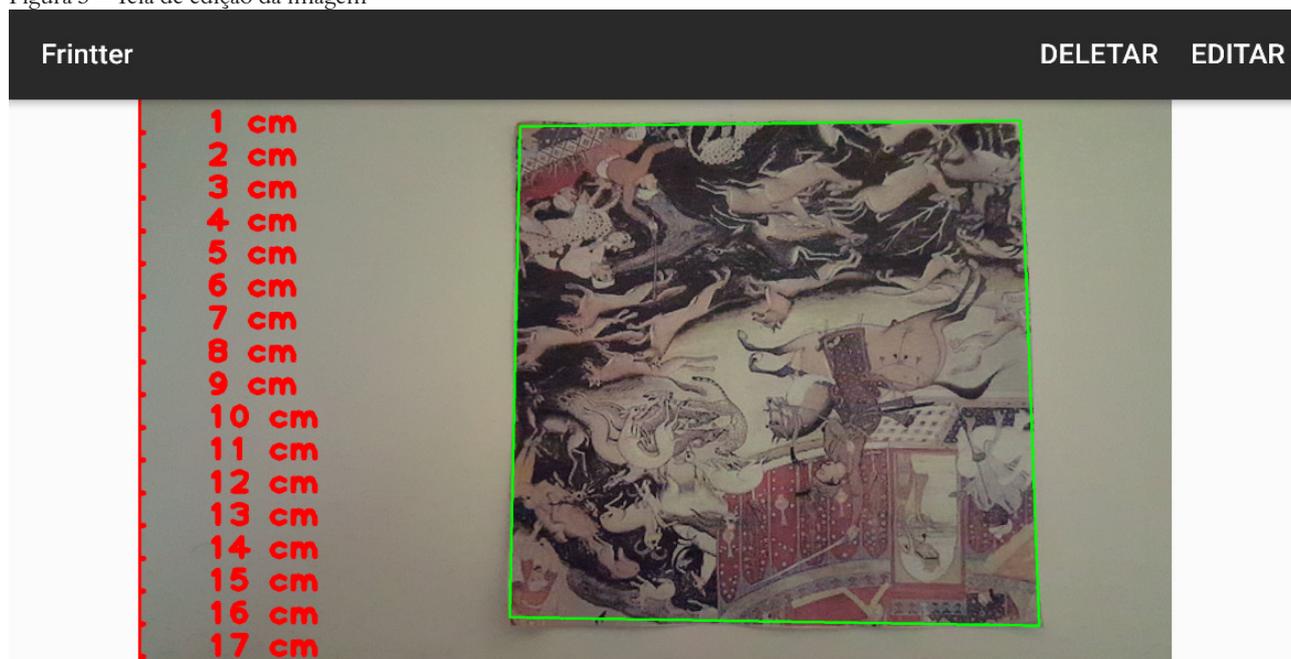
```

Fonte: os autores.

Ao ser capturada a foto, após devidamente detectado o marcador, o método *takephoto* é disparado, o qual captura o *frame* da câmera e converte para uma imagem no formato JPEG. Esse processo inicial deve ser realizado para se obter a medida mais precisa possível para o próximo passo, que é a mensuração.

Após ser tirada a fotografia, ela passa a ser exibida na próxima tela de edição, como é mostrado na Figura 3. Nesse momento a imagem já está armazenada nos arquivos do sistema. Também é possível editar a imagem, excluí-la ou realizar a mensuração. Ao excluir, a imagem não será possível visualizá-la posteriormente.

Figura 3 – Tela de edição da imagem



Fonte: os autores.

Para realizar a mensuração é necessário apenas o clique na tela para identificar o ponto inicial, arrastá-lo para a posição desejada e soltar. Quando é feito esse procedimento o método *onTouchEvent* é disparado, e nele são salvas as coordenadas x e y iniciais, que são representadas pelo evento *ACTION_DOWN* desse método, ou seja, quando for tocado na tela. Ao arrastar e soltar, os eventos *ACTION_MOVE* e *ACTION_UP* também são disparados, o evento *ACTION_MOVE* não é utilizado, mas foi mantido para se ter um entendimento desse método. Por fim, no evento *ACTION_UP* são armazenadas as coordenadas x e y finais que irão formar uma linha reta representando a medida. Na Figura 4 é exibido o procedimento realizado de uma mensuração.

Figura 4 – Tela de edição com a mensuração realizada



Fonte: os autores.

Por fim, é possível editar a imagem e retirar o marcador da imagem, deixando-se apenas as mensurações realizadas.

6 RESULTADOS E DISCUSSÃO

Testes realizados nos *smartphones* Galaxy S6 e LG Optimus L7 apresentaram considerável diferença de desempenho com relação à câmera e ao processamento. O Galaxy, por ter uma configuração de hardware mais robusta e uma câmera de alta qualidade, não possui *delay* ao executar os processos de leitura do marcador, já no LG L7 é bem perceptível o *delay* ao executar os mesmos processos. Notou-se pouca diferença no resultado final, pois o cálculo realizado é proporcional à resolução da câmera.

Os resultados obtidos nos testes também provaram ser bastante variados, visto que o método de localização do marcador se mostra pouco eficiente para marcadores com poucas características. Já para marcadores que possuem uma grande variedade de detalhes esse processo se torna mais rápido e confiável. A distância do objeto também é um fator que influencia no resultado final, pois a uma distância superior a 50 cm, mesmo uma imagem de 15x15 cm fica com os detalhes menos perceptíveis.

Do mesmo modo, ficou evidente a influência de fatores como a iluminação da cena. Um ambiente com pouca iluminação também tende a ter menos pontos perceptíveis, dificultando a localização do marcador. Outro fator levantado que interfere diretamente no resultado é o ângulo de diferença na posição da câmera no momento inicial, em que é feita a localização do marcador. Ainda, outro fator encontrado é que a perda do foco do objeto, considerando-se que uma variação no ângulo da câmera distorce a localização do marcador, dificultará o cálculo-base.

Contudo, o desenvolvimento do sistema se mostrou promissor, afinal, em situações em que as imagens são capturadas em condições favoráveis, os resultados obtidos mostraram-se satisfatórios, conforme foi exibido na Figura 4, visto que as medidas são realizadas com uma precisão considerável. Com base nas medidas realizadas pelo Frintter e nas medidas reais, foi observada uma taxa de erro de aproximadamente 10% acima da medida real.

7 CONCLUSÃO

O desenvolvimento do *app* Frintter tem por objetivo proporcionar uma alternativa para garantir o processo de mensuração de pegadas de animais e também diminuir os custos gerados com os métodos usados atualmente. O processo de desenvolvimento foi realizado com o uso de ferramentas de distribuição gratuita, sendo a maioria de código-fonte aberto. Com isso, o sistema foi projetado para plataforma Android, o que possibilita seu maior acesso e disseminação.

Os resultados obtidos com os testes realizados demonstram que o sistema é funcional e promissor. Com ele foi possível realizar medidas precisas, salvar as imagens e visualizá-las em outros dispositivos. Entretanto, existem algumas limitações, as quais se propõe eliminar em trabalhos futuros. Uma delas é melhorar o processo e, se possível, atingir uma precisão real das pegadas.

De modo geral, o *app* pode ser melhorado, porém o atual estado permite realizar as medidas de forma precisa, garantindo sua eficiência. Por se tratar de manipulação de imagens em um ambiente com 3D trazendo-as para um ambiente 2D, as dificuldades aumentam. Contudo, tais dificuldades podem ser contornadas com as ferramentas certas e o conhecimento para manipulá-las.

Frintter: mobile approach for footprint measurement of mammals in faunistic surveys

Abstract

This paper aims to present the study and development of an application to aid in the measurement of animal footprints in faunistic surveys. It has developed an aim to promote a more effective, practical and easy way for a measurement of animal footprints. Given a way of analyzing the footprints in faunistic surveys, there is a great need to obtain Speed in the identity of the same things, and for each identified footprint, old methods are used, excess baggage and weight becomes a factor taken into account, where the sampling sites are in places where it is difficult to access. The application is designed for an Android platform using an OpenCV library for the digital processing of the obtained images. Based on the experiments performed, the results show that there are still improvements and tests to be done, however, the system shows promise.

Keywords: Faunistic Surveys. OpenCV. Footprints. Frintter.

REFERÊNCIAS

BARBIERI, J. C. Avaliação de impacto ambiental na legislação brasileira. **RAE – Revista de Administração de Empresas**, v. 35, n. 2, 1995.

DELAI, R. L.; COELHO, A. D. **Visão computacional com a OpenCV – material apostilado e veículo seguidor autônomo**. Disponível em: <<http://docplayer.com.br/14034135-Visao-computacional-com-a-opencv-material-apostilado-e-veiculo-seguidor-autonomo-professora-da-escola-de-engenharia-maua-eem-ceun-imt.html>>. Acesso em: 08 nov. 2018.

FRENSEL, K.; DELGADO, J. **Geometria Analítica**. 2011. 269 p. Monografia (Especialização em Matemática)– Universidade Federal do Maranhão, São Luís, 2011. Disponível em: <<http://www.mat.ufmg.br/~portosil/geometria-analitica-ufma.pdf>>. Acesso em: 08 nov. 2016.

GOVERNO DO ESTADO DE SANTA CATARINA. **Lista das espécies da fauna ameaçada de extinção em santa catarina**: relatório técnico final. 2010. Disponível em: <http://www.fatma.sc.gov.br/upload/Fauna/relat9500rio_t9500cnico_final_lista_esp9500cies_amea9500adas.pdf>. Acesso em: 08 nov. 2016.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 1999, Kerkyra. **Anais...** 1999. p. 1550-1157.

MARENGONI, M.; STRINGHINI, D. Introdução à Visão Computacional usando OpenCV. **Rita**, Porto Alegre, v. 7, n. 1, p. 126-160, mar. 2009. Disponível em: <http://seer.ufg.br/rita/article/viewFile/rita_v16_n1_p125/7289>. Acesso em: 08 nov. 2016.

MIRANDA, J. R. et al. **Levantamento Faunístico e Avaliação da Biodiversidade Agrossistemas da Bacia do Rio Pardo**. 2008. Disponível em: <http://ainfo.cnptia.embrapa.br/digital/bitstream/CNPM/2176/1/doc66_fauna-bio_RioPardo.pdf>. Acesso em: 08 nov. 2016.

MORO-RIOS, R. F. et al. **Manual de Rastros da Fauna Paranaense**. Curitiba: Instituto Ambiental do Paraná, 2008. Disponível em: <<http://www.cesumar.br/dcl/cienciasbiologicas2/arquivos/rastros.PDF>>. Acesso em: 07 mar. 2016.

NASCIMENTO, J. L. do. Instituto Chico Mendes de Conservação da Biodiversidade. **Atlas da Fauna Brasileira Ameaçada de Extinção em Unidades de Conservação Federais**. 2011. Disponível em: <<http://www.icmbio.gov.br/portal/images/stories/documentos/Atlas-ICMBio-web.pdf>>. Acesso em: 08 nov. 2016.

SANTANA, B. A. S. et al. Desempenho de Algoritmos Detectores De Keypoints para um Sistema de Navegação Visual de Robôs Baseados em Smartphones. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 12., Natal, 2015. **Anais...** Natal, 25-28 out. 2015.

SILVEIRA, L. F. et al. **Para que servem os inventários de fauna?** 2010. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142010000100015>. Acesso em: 08 nov. 2016.